Solving high-dimensional PDEs, approximation of path space measures and importance sampling of diffusions

Von der Fakultät 1 – MINT – Mathematik, Informatik, Physik, Elektro- und Informationstechnik der Brandenburgischen Technischen Universität Cottbus-Senftenberg genehmigte Dissertation zur Erlangung des akademischen Grades eines Dr. rer. nat. vorgelegt von

Lorenz Richter

geboren am 10.01.1988 in Köln

Vorsitzender: Prof. Dr. Gennadiy Averkov
Gutachter: Prof. Dr. Carsten Hartmann
Gutachter: Prof. Dr. Gabriel Stoltz
Gutachter: Prof. Dr. Eric Vanden-Eijnden

Tag der mündlichen Prüfung: 8.12.2021

Contents

Die Mathematiker sind eine Art Franzosen: redet man zu ihnen, so übersetzen sie es in ihre Sprache, und dann ist es alsobald ganz etwas anderes. Johann Wolfgang von Goethe

Abstract

Motivated by computing functionals of high-dimensional, potentially metastable diffusion processes, this thesis studies robustness issues appearing in the numerical approximation of expectation values and their gradients. A major challenge being high variances of corresponding estimators, we investigate importance sampling of stochastic processes for improving statistical properties and provide novel nonasymptotic bounds on the relative error of corresponding estimators depending on deviations from optimality. Numerical strategies that aim to come close to those optimal sampling strategies can be encompassed in the framework of path space measures, and minimizing suitable divergences between those measures suggests a variational formulation that can be addressed in the spirit of machine learning. A key observation is that while several natural choices of divergences have the same unique minimizer, their finite sample properties differ vastly. We provide the novel *log-variance divergence*, which turns out to have favorable robustness properties that we investigate theoretically and apply in the context of path space measures as well as in the context of densities, for instance offering promising applications in Bayesian variational inference.

Aiming for optimal importance sampling of diffusions is (more or less) equivalent to solving Hamilton-Jacobi-Bellman PDEs and it turns out that our numerical methods can be equally applied for the approximation of rather general high-dimensional semi-linear PDEs. Motivated by stochastic representations of elliptic and parabolic boundary value problems we refine variational methods based on backward SDEs and provide the novel *diffusion loss*, which can be related to other state-of-the-art attempts, while offering certain numerical advantages.

Zusammenfassung

Motiviert durch das Berechnen von Funktionalen hochdimensionaler stochastischer Prozesse, untersucht diese Dissertation Robustheitseigenschaften, die in der numerischen Approximation von Erwartungswerten sowie deren Gradienten auftreten. Insbesondere für numerische Algorithmen, die auf geschätzten Werten basieren, ist eine gewisse Stabilität bezüglich möglicherweise auftretender zufälliger Fluktuationen unabdingbar. Da für Erwartungswerte in der Regel keine geschlossenen Formeln existieren, greift man hier auf die Monte-Carlo-Methode zurück, welche jedoch die Herausforderung von hohen Varianzen, d.h. hohen statistischen Fehlern, mit sich bringt. Insbesondere bei der Behandlung so genannter seltener Ereignisse kann der relative Fehler so hoch sein, dass die geschätzten Zahlen gänzlich unbrauchbar sind. Es ist Ziel dieser Arbeit, diese (Nicht-)Robustheitseigenschaften von statistischen Schätzern besser zu verstehen und Algorithmen zu entwickeln, welche insbesondere in hohen Dimensionen effizient anwendbar sind. Dabei fokussieren wir unsere Analyse auf das Importance Sampling als eine populäre Methode der Varianzreduktion.
Wir entwickeln neuartige nicht-asymptotische Schranken für den relativen Fehler von Importance-Sampling-Schätzern in Abhängigkeit von Abweichungen von der optimalen Lösung – zunächst in einem abstrakten Sinne. Schließlich betrachten wir insbesondere hochdimensionale, möglicherweise metastabile stochastische Prozesse – dies korrespondiert zu Importance Sampling im so genannten Pfadraum, wofür wir zusätzliche Formeln für den relativen Fehler der Schätzer entwickeln. Strategien für ein möglichst optimales Importance Sampling von Diffusionsprozessen korrespondieren zu stochastischen Optimalsteuerungsproblemen. Inspiriert von diesem Zusammenhang entwickeln wir einen einheitlichen Rahmen aus der Perspektive von Pfadmaßen, welcher neue numerische Verfahren motiviert. Insbesondere legt er die Minimierung geeigneter Divergenzen zwischen diesen Maßen in Form einer Variationsformulierung nahe, welche im Sinne des Machine Learnings gelöst werden kann. Hierbei ist es eine zentrale Erkenntnis, dass verschiedene Divergenzen zwar denselben eindeutigen Minimierer haben, sich aber hinsichtlich ihrer statistischen Eigenschaften stark unterscheiden können, etwa in Abhängigkeit der Dimension oder bezüglich möglicher Fluktuationen in der Nähe der optimalen Lösung. Im Zuge dessen schlagen wir die neuartige *Log-Varianz-Divergenz* vor, welche vorteilhafte Robustheitseigenschaften mit sich bringt, die wir durch theoretische Analyse und zahlreiche numerische Beispiele belegen können. Als eine zusätzliche Anwendung dieser Divergenz für Dichten betrachten wir die Bayes'sche variationelle Inferenz. Dies führt zu einem verbesserten Gradientenschätzer, welcher sowohl theoretisch als auch in numerischen Experimenten seinem naiven Gegenpart deutlich überlegen ist.
Das Streben nach optimalem Importance Sampling von Diffusionsprozessen ist (mehr oder weniger) äquivalent zum Lösen von Hamilton-Jacobi-Bellman-PDEs, und es stellt sich heraus, dass unsere entwickelten numerischen Methoden gleichermaßen für die Approximation bestimmter allgemeinerer hochdimensionaler semilinearer PDEs angewendet werden können. Motiviert durch stochastische Darstellungen von elliptischen und parabolischen Randwertproblemen entwickeln wir darauf aufbauend bestimmte variationelle Methoden weiter, die auf rückwärts-SDEs basieren, und schlagen den neuartigen *diffusion loss* vor, welcher gegenüber den alternativen Methoden einige numerische Vorteile mit sich bringt.

Acknowledgements

This thesis would not have been possible without the help of many great people, for which I am deeply thankful. First of all, I want to thank my supervisor, Carsten Hartmann, for his endless support and trust. Without him I would not have completed this endeavor with so much joy, but maybe even more importantly, without him I would most probably not even have started it. I am grateful for his continuous guidance, for providing me the freedom I needed and for believing in me. It was a great pleasure to benefit from his wisdom and his inspirations, offering a working environment in which I felt home.
The first time I met Nikolas Nüsken was when I accidentally woke him up in a shared room at a conference at CIRM in Marseille in the middle of the night. Ever since then, he has become a great teacher and a friend. I am extremely grateful that he decided to work with me and took the time to be there whenever needed. His genius contributed significantly to many of the ideas developed in this thesis.
I would not have made it this far in my mathematical journey had it not been for two very inspiring people: My high school teacher Eberhard Petschel taught me curiosity and showed me what really counts in mathematical thinking, and in my first university semesters, it were Klaus Ecker's very enthusiastic and clear lectures that convinced me to follow my path.
My last years have been a pleasure partly due to great working conditions. I am thankful to Ralf Wunderlich for welcoming me very warmly in Cottbus and making it all possible in the beginning, as well as to Christof Schütte for having me in the Biocomputing group. The support of Annette Fischer, Dorothé Auth and Nina Fabiančič could not have been better and I thank them for their assistance. It was a pleasure to be part of the stimulating interdisciplinary research atmosphere in the CRC 1114 and I thank everyone being involved. I am especially grateful for having very nice colleagues at FU Berlin and BTU Cottbus-Senftenberg – thanks, Wei Zhang, Lara Neureither, Omar Kebiri, Markus Strehlau, Jannes Quer and Enric Ribera Borrell for spending your time with me!
Without stimulating discussions science can get empty. I want to thank Gabriel Stoltz and Grégoire Ferré for having me in Paris and Marseille – these were great times. It has been fun and fruitful to work with Simon Becker and Martin Redmann. I also thank Jeremy Heng for inspiring conversations and Arnulf Jentzen for being interested in my work. Certainly, the collaborations with Ayman Boustati, Francisco J. R. Ruiz, Ömer Deniz Akyildiz and Leon Sallandt have been a stimulating pleasure. Finally, I am very grateful that Eric Vanden-Eijnden and Gabriel Stoltz take their time to review my thesis.
I want to thank Philipp Jackmuth for his trust and patience. He and the whole dida team have been extremely supporting and I am happy that I can rely on all of them. Also I want to thank Michael Flamm, for his great support and interest.
Finally, I should say that without the love and unconditional support of my family, I would not be who I am now. For this I thank my brother Julius, and my parents, to whom I owe everything.

Published articles and preprints

Some publications and preprints have been finished during the work on this thesis, all of which would not have been possible without fantastic collaborators. The following papers are related to the main theme of the thesis – some of them will be referred to at later stages.

- C. Hartmann, L. Richter, C. Schütte, and W. Zhang. Variational characterization of free energy: Theory and algorithms. *Entropy*, 19(11):626, 2017
- C. Hartmann, O. Kebiri, L. Neureither, and L. Richter. Variational approach to rare event simulation using least-squares regression. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(6):063107, 2019
- N. Nüsken and L. Richter. Solving high-dimensional Hamilton–Jacobi–Bellman PDEs using neural networks: perspectives from the theory of controlled diffusions and measures on path space. *Partial Differential Equations and Applications*, 2(4):1–48, 2021
- L. Richter, A. Boustati, N. Nüsken, F. Ruiz, and O. D. Akyildiz. VarGrad: A low-variance gradient estimator for variational inference. *Advances in Neural Information Processing Systems*, 33, 2020
- C. Hartmann and L. Richter. Nonasymptotic bounds for suboptimal importance sampling. *arXiv preprint arXiv:2102.09606*, 2021
- L. Richter, L. Sallandt, and N. Nüsken. Solving high-dimensional parabolic PDEs using the tensor train format. *International Conference on Machine Learning*, 2021
- N. Nüsken and L. Richter. Interpolating between BSDEs and PINNs: deep learning for elliptic and parabolic boundary value problems. *arXiv preprint arXiv:2112.03749*, 2021

The following two articles deal with model reduction of deterministic and stochastic processes and are therefore not directly related to the focus of this thesis.

- S. Becker, C. Hartmann, M. Redmann, and L. Richter. Error bounds for model reduction of feedback-controlled linear stochastic dynamics on Hilbert spaces. *Stochastic Processes and their Applications*, 2022
- S. Becker and L. Richter. Model order reduction for (stochastic-) delay equations with error bounds. *arXiv preprint arXiv:2008.12288*, 2020

Chapter 1

Introduction

Any meaningful method in quantitative science must be robust. In particular, any numerical algorithm relying on values that need to be estimated has to be stable with respect to fluctuations to some extend in order to yield correct results and account for reliable insights. A key quantity for the analysis of any random system is the expectation value of some observable,

$$\mathbb{E}\left[f(X)\right], \tag{1.1}$$

where X is a random variable and f some prescribed functional. Since usually no closed-form formulas for the computation of (1.1) are available one has to rely on statistical estimations. In fact, the numerical approximation of expectations by the so-called Monte Carlo method is ubiquitous in various disciplines such as quantitative finance [106, 107], machine learning [34], computational statistics [101] or statistical physics [275], to name just a few. Depending on the situation at hand, this estimation problem can be more or less difficult, but it turns out that a major problem are potentially large statistical errors of naive sampling strategies, noting that especially high-dimensional settings pose formidable challenges [287, 291]. It is therefore a common goal to build estimators that exhibit a small variance, as compared to the quantity of interest, and thus a small relative error. A typical situation, in which variance reduction is indispensable, is the simulation of rare events with its characteristic exponential divergence of the relative error in the parameter that controls the rarity of the quantity of interest [43].

There exist multiple strategies that strive for making estimation problems more robust [7]. In this thesis we shall focus on importance sampling as a standard tool for variance reduction. The idea is to sample from an alternative probability measure and reweight the resulting random variables with the likelihood ratio in order to still produce an unbiased estimator for the quantity of interest. Naturally, the question arises which probability distribution to choose. In theory, under appropriate assumptions, there exists an optimal proposal measure that yields a zero-variance estimator and therefore removes all the stochasticity from the problem. However, this measure depends on the quantity of interest itself and is therefore practically useless. Coming up with feasible proposals, on the other hand, is a science in itself, and various numerical experiments demonstrate that it is indeed a crucial one, as making bad choices can even increase the relative error of importance sampling estimators significantly, therefore counteracting the original intention and leading to substantial robustness issues [32, 108, 206, 272]. Loosely speaking, importance sampling gets increasingly difficult and sensitive to small deviations from an optimal proposal distribution if the quantity of interest is mainly supported on small regions which have little overlap with the regions of the proposal; such a phenomenon is more likely to appear in high dimensions. Moreover, concentration of measure that may lead to degeneracies of likelihood ratios when the probability of certain events becomes exponentially small is more likely to occur in high dimensional settings [23, 237].

In this thesis we are particularly interested in random variables related to high-dimensional stochastic processes. Being for instance relevant in engineering [298], physics [189, 208] or data assimilation [187], this setting brings additional computational challenges, such as the numerical discretization of the process, metastable behaviors leading to rare event phenomena or large trajectory lengths – all of which can affect the relative error of corresponding estimators in non-trivial ways and make robust estimation a challenging endeavor. Importance sampling has largely been studied in the context of sampling from probability densities in $\mathbb{R}^d$ and has become popular for sampling diffusion processes only in recent years, with applications ranging from molecular dynamics [131] to mathematical finance [106, 107] and climate modelling [240]. What the aforementioned fields have in common, is that the quantities of interest are often related to rare events or large deviations from a mean or an equilibrium state, and, often, the dynamics exhibits metastability, i.e. it features rare transitions between semi-stable equilibria. To simulate these systems, variance reduction techniques like importance sampling are

essential. Unfortunately, however, robust importance sampling of diffusions seems even more challenging than in the density case and potential non-robustness issues have been observed in particular if the state space dimension or trajectory length is large [80].

Motivated to better understand these non-robustness issues of importance sampling (and non-robustness issues in sampling in general), while still aiming for feasible numerical strategies in high dimensions, we thus state the following two guiding research questions:

1. How can we quantify the estimator's statistical performance when it relies on suboptimal importance sampling proposal measures?
2. How can we analyze and improve robust numerical strategies that aim to identify optimal importance sampling strategies for diffusions, even if the dimensionality of the state space is large?

For stochastic processes our guiding problem can be understood as importance sampling in path space, where drawing trajectories from alternative path space measures is equivalent to adding a feedback control to the original stochastic dynamics. In fact, it turns out that aiming for the optimal zero-variance control (i.e. the optimal importance sampling path measure) can be understood as a stochastic optimal control problem and it seems reasonable to take into account methods and theoretical connections that have been developed in this branch of mathematics over the last decades. A classical result in control theory is the connection of an optimal control function to the solution of a nonlinear partial differential equation (PDE) of Hamilton-Jacobi type. In fact, one can show an explicit correspondence between the optimal importance sampling control and the solution to a specific Hamilton-Jacobi-Bellman (HJB) PDE, or similarly, by an appropriate transformation, to a linear Feynman-Kac PDE. It is therefore evident that numerically aiming for the optimal importance sampling measure is deeply connected to approximating solutions of PDEs.

The numerical treatment of PDEs is a prominent field in applied mathematics and methods such as finite differences or finite elements have been studied extensively [239]. However, their practical use is often limited by the fact that solving those equations becomes notoriously difficult in high-dimensional settings. The so-called "curse of dimensionality" refers to the phenomenon that the computational effort scales exponentially in the dimension, rendering classical grid based methods infeasible [19]. In recent years, however, there have been fruitful developments in combining Monte Carlo based algorithms with neural networks in order to tackle high-dimensional problems in a way that seemingly does not suffer from this curse, resting primarily on stochastic representations of the PDEs under consideration [86, 87, 144, 242]. Many of the suggested algorithms perform remarkably well in practice and some theoretical results proving beneficial approximation properties of neural networks in the PDE setting are now available [119, 157]. Still, a complete picture remains elusive, and the optimization aspect in particular continues to pose challenging and mostly open problems, both in terms of efficient implementations and theoretical understanding. Most of those machine learning based attempts can be understood as variational methods, where suitable loss functionals that admit a global minimum representing the solution to the problem at hand are minimized by stochastic gradient descent. The loss is typically given in terms of expectation values and consequently needs to be estimated based on a sample. For a principled understanding of variational attempts it is therefore central to analyze the properties of loss functions and corresponding Monte Carlo estimators and identify guidelines that promise good and robust performance. In this respect, important desiderata are the absence of local minima as well as the availability of low-variance gradient estimators, noting that smaller variances of gradient estimators usually yield better and faster convergence of the corresponding algorithms [37].

Coming back to our guiding problem of identifying optimal importance sampling controls, the particular structure of the connected HJB PDE implies that a variety of loss functions can be constructed and analyzed in terms of divergences between probability measures on the path space associated to controlled stochastic processes, thereby providing a unifying framework for variational formulations. In analogy to the existence of an optimal control function there exists a target path space measure that one can aim to approximate via minimizing divergences in a family of proposal measures. Several natural choices for divergences are available, all having the same unique minimizer, however it will turn out that they differ vastly in their finite sample properties, leading to different losses and different robustness properties of corresponding algorithms. Furthermore, some of the divergences can be readily applied for the approximation of more general semi-linear PDEs for which the nonlinearity only depends on the solution through its gradient, leading to robustness improvements that seem to be significant especially in challenging PDE problems for instance exhibiting metastable features.

1.1 Outline of the thesis

This thesis is structured as follows. Starting with the goal of identifying optimal importance sampling strategies, Section 1.3 formally introduces five connected problems that relate to different fields of mathematics, such

as stochastic optimal control, PDEs and backward SDEs (BSDEs). In Section 1.4 we will show that, given appropriate conditions, all these problems are in fact (more or less) equivalent and thereby build an appropriate starting point for our analysis. There exist multiple numerical strategies for solving either of the five problems and Section 1.5 shall provide a comprehensive literature overview, also referring to work related to robustness issues of importance sampling. We will end the introductory part by discussing potential generalizations of the stated problems.

Chapter 2 shall provide proper theoretical foundations for most of the aspects this thesis is based on. In Section 2.1 we introduce to stochastic optimal control theory and derive the Hamilton-Jacobi-Belmann PDE. In Section 2.2 we review stochastic representations of PDEs, focusing on linear PDEs in Section 2.2.1 and nonlinear PDEs via the introduction of BSDEs in Section 2.2.2. We will discuss numerical discretizations of both forward and backward processes in Section 2.2.3. In Section 2.3.1 we will formally introduce importance sampling, in particular characterizing zero-variance strategies and focusing on sampling of diffusions in Section 2.3.2, before in Section 2.3.3 we will draw connections to the theory of large deviations. In Section 2.3.4 we will review the Donsker-Varadhan variational relation, again outlining connections to our original sampling intention. Finally, in Section 2.4 we will define neural networks, refer to some recent results on their approximation capabilities in particular in high-dimensional settings and briefly discuss stochastic optimization for targeting the approximation task.

Chapter 3 will study importance sampling in more depth, aiming to provide nonasymptotic bounds on statistical errors for suboptimal choices of proposal measures. In Section 3.1 we define importance sampling in an abstract setting and recall the notions of divergences between proposal and target measures, while refining a bound on the relative error and highlighting robustness issues in high dimensions. In Section 3.2 we will move to importance sampling of stochastic processes. We will translate the bounds from the previous section to this setting and derive an exact formula for the relative error with which we can then state novel bounds that allow for interpretations with respect to robustness in higher dimensions and long time horizons. When focusing on PDE methods in Section 3.2.1 we can essentially re-derive bounds from the previous section. In Section 3.2.2 we comment on how our bounds can help to understand potential issues in the small noise regime, and finally, in Section 3.3 we present some numerical examples with which we will illustrate the previously discussed issues.

Chapter 4 will take the perspective of path space measures for the design of robust iterative algorithms to solve either of the problems introduced in Chapter 1. As a unifying viewpoint, in Section 4.1 we will define viable loss functions via divergences on path space and discuss their connections to the algorithmic approaches encountered in Section 1.5. In particular, we will introduce the novel *log-variance divergence* and elucidate its relationship with forward-backward SDEs. In the two upcoming sections we will analyze properties of the suggested losses, where in Section 4.2 we obtain equivalence relations that hold in an infinite batch size limit and in Section 4.3 we investigate the variances associated to the losses' estimator versions. In the latter case, we will consider stability close to the solution as well as in high dimensional settings. In Section 4.4 we will provide numerical examples that illustrate our findings, while Appendix B.6 will bring further demonstrations by explicit calculations for linear Ornstein-Uhlenbeck dynamics.

In Chapter 5 we will apply the log-variance divergence to Bayesian variational inference, resulting in a low-variance gradient estimator. We will first provide background and context in Section 5.1, before in Section 5.2 we will derive *VarGrad*, our low-variance gradient estimator, and show its connection to the log-variance loss. Theoretical analysis on VarGrad will be presented in Section 5.3, highlighting its interpretation as a control variate version of a corresponding naive estimator. We can show that VarGrad is close to the optimal control variate scaling under some mild assumptions, in particular demonstrating lower variance than the naive estimator. In Section 5.4 we provide numerical experiments which demonstrate these findings, leading to computational advantages and faster convergence of corresponding algorithms.

Chapter 6 will be devoted to numerical strategies for solving more general semi-linear PDEs. In Section 6.1 we will start with a variational approximation method for linear PDEs which is based on L^2 projections. In Section 6.2 we will show how parabolic PDEs on unbounded domains can be approached by BSDEs via backward iterations – we will provide multiple numerical experiments showing in particular how this framework can be used to exploit the tensor train format for efficient approximations in high dimensions. In Section 6.3 we will review and extend residual and BSDE based methods for solving nonlinear PDEs from the perspective of variational formulations of elliptic and parabolic boundary value problems. We will introduce the novel *diffusion loss* as a combination of the previous two methods, which will turn out to bring some numerical advantages. In Section 6.3.4 we will show how our framework can be used to solve special prominent PDE problems and in Section 6.3.5 how it can be extended to linear and nonlinear elliptic eigenvalue problems. Section 6.3.6 will discuss algorithmic design including some further modifications of the losses. Finally, in Section 6.3.7 we will provide various numerical experiments comparing the different losses.

We will conclude this thesis in Chapter 7, providing also multiple possible directions of future research. In particular, we will demonstrate connections to the Schrödinger problem that motivate novel algorithmic approaches and we will provide a proof of concept for learning zero-variance importance sampling densities via the concept of normalizing flows.

1.2 Contributions and main results

- We systematically review the connections and equivalences between sampling of diffusions, stochastic optimal control and certain HJB PDEs (Theorem 1.2).
- We provide nonasymptotic bounds on the relative error of suboptimal importance sampling, explaining fragility that has often been observed in numerical simulations before. Some of those bounds are formulated in an abstract measurable space and can be readily applied to densities (Proposition 3.9). For path space measures, we deduce some additional bounds that, in particular, highlight the sampling challenges due to high dimensionality or long trajectories (Proposition 3.15).
- We develop a principled framework for solving specific HJB PDEs based on divergences between path space measures, encompassing various existing methods. The perspective of constructing loss functions via those divergences offers a systematic approach to algorithmic design and analysis. We show that modifications of recently proposed approaches based on forward-backward SDEs [86, 122] can be placed within this framework (Chapter 4).
- We introduce the novel *log-variance divergence*, encapsulating a family of forward-backward SDE systems (Definition 4.4). The aforementioned adjustments needed to establish the path space perspective often lead to faster convergence and more accurate approximation of the solution, as we demonstrate by means of numerical experiments (Chapter 4).
- We show that certain instances of algorithms based on the KL divergence (or control objective) and the log-variance divergence (or forward-backward SDEs) are equivalent when the sample size is large (Proposition 4.19).
- We investigate the properties of sample based gradient estimators associated to the losses and divergences under consideration. In particular, we define two notions of stability: robustness of a divergence under tensorization (related to stability in high-dimensional settings) and robustness at the optimal control solution (related to stability of the final approximation). From the losses and divergences considered in this thesis, we show that only the log-variance divergence satisfies both desiderata and illustrate our findings by means of extensive numerical experiments (Propositions 4.25 and 4.29).
- We apply the log-variance divergence to densities in the context of Bayesian variational inference, leading to a low-variance gradient estimator which we call *VarGrad*. We show theoretically that this estimator is somehow close to an optimal control variate scaling and conclude that under mild assumptions it has lower variance than the corresponding naive estimator (Propositions 5.6 and 5.9).
- We review and extend strategies for solving semi-linear elliptic and parabolic boundary value problems from the perspective of variational formulations, incorporating and generalizing residual and BSDE based methods. We introduce the novel *diffusion loss*, which combines ideas from both methods, and show that it can in fact be interpreted as an interpolation between the two (Propositions 6.24 and 6.25). We illustrate potential advantages of this approach in numerical experiments.
- We show that the variational attempts for solving elliptic boundary value problems can be extended to approximating principal eigenpairs to linear and nonlinear eigenvalue problems (Proposition 6.26).
- We propose to use the tensor train format in backward iteration schemes for solving high-dimensional parabolic PDEs and demonstrate potential numerical advantages compared to neural network based attempts in various numerical experiments (Section 6.2.3).
- We connect the Schrödinger problem to optimal importance sampling and suggest an algorithm for sampling from specified target densities based on control theoretic ideas and the log-variance divergence (Chapter 7 and Appendix B.10).
- We provide a proof of concept on how normalizing flows can be used to learn optimal importance sampling densities via minimizing a suitable log-variance loss (Chapter 7 and Appendix B.11).

1.3 Five perspectives on (more or less) the same problem

This section introduces the main ambition of the thesis more formally. Starting with the principal goal to develop robust methods for the computation of expectation values related to diffusion processes via importance sampling, we will show how this endeavor is inherently connected to four other problems, each being rooted in different branches of mathematics.

Throughout, we will assume a fixed filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0}, \Lambda)$ satisfying the 'usual conditions' [171, Section 21.4] and consider stochastic differential equations (SDEs) of the form

$$\mathrm{d}X_s = b(X_s, s)\,\mathrm{d}s + \sigma(X_s, s)\,\mathrm{d}W_s, \qquad X_t = x_{\mathrm{init}}, \tag{1.2}$$

on the time interval $s \in [t, T]$, $0 \leq t < T < \infty$. Here, $b : \mathbb{R}^d \times [t, T] \to \mathbb{R}^d$ denotes the drift coefficient, $\sigma : \mathbb{R}^d \times [t, T] \to \mathbb{R}^{d\times d}$ the diffusion coefficient, $(W_s)_{t\leq s\leq T}$ denotes standard d-dimensional Brownian motion[1], and $x_{\mathrm{init}} \in \mathbb{R}^d$ is the (deterministic) initial condition. For now, we will work under the following conditions specifying the regularity of b and σ.

Assumption 1 (Coefficients of the SDE (1.2))**.** *The coefficients b and σ are continuously differentiable, σ has bounded first-order spatial derivatives, and $(\sigma\sigma^\top)(x, s)$ is positive definite for all $(x, s) \in \mathbb{R}^d \times [t, T]$. Furthermore, there exist constants $C, c_1, c_2 > 0$ such that*

$$|b(x, s)| \leq C\,(1 + |x|)\,, \qquad \textit{(linear growth)} \tag{1.3a}$$

$$c_1|\xi|^2 \leq \xi \cdot (\sigma\sigma^\top)(x, s)\xi \leq c_2|\xi|^2, \qquad \textit{(ellipticity)} \tag{1.3b}$$

for all $(x, s) \in \mathbb{R}^d \times [t, T]$ and $\xi \in \mathbb{R}^d$.

Let us further introduce a modified version of (1.2),

$$\mathrm{d}X^u_s = (b(X^u_s, s) + \sigma(X^u_s, s)u(X^u_s, s))\,\mathrm{d}s + \sigma(X^u_s, s)\,\mathrm{d}W_s, \qquad X^u_t = x_{\mathrm{init}}, \tag{1.4}$$

where we think of $u : \mathbb{R}^d \times [t, T] \to \mathbb{R}^d$ as a control term steering the dynamics. We will assume that $u \in \mathcal{U}$, the set of *admissible controls*. For definiteness, we will set

$$\mathcal{U} = \left\{u \in C^1(\mathbb{R}^d \times [0, T], \mathbb{R}^d) : \quad u \text{ grows at most linearly in } x, \text{ in the sense of (1.3a)}\right\}, \tag{1.5}$$

but note that the smoothness and boundedness assumptions can be relaxed in various scenarios. Under Assumption 1 and with $\mathcal{U}$ as defined in (1.5), the SDEs (1.2) and (1.4) admit unique strong solutions (see Theorem B.1 in the appendix).

We will usually fix the initial time to be $t = 0$, i.e. consider the SDEs (1.2) and (1.4) on the interval $[0, T]$. Later we will discuss how T can be replaced with a random stopping time. For a fixed initial condition $x_{\mathrm{init}} \in \mathbb{R}^d$, let us introduce the path space

$$\mathcal{C} = C_{x_{\mathrm{init}}}([0, T], \mathbb{R}^d) = \left\{X : [0, T] \to \mathbb{R}^d \mid X \text{ continuous},\ X_0 = x_{\mathrm{init}}\right\}, \tag{1.6}$$

equipped with the supremum norm and the corresponding Borel-σ-algebra, and denote the set of probability measures on $\mathcal{C}$ by $\mathcal{P}(\mathcal{C})$. The SDEs (1.2) and (1.4) induce probability measures on $\mathcal{C}$ defined to be the laws associated to the corresponding strong solutions; those measures will be denoted by $\mathbb{P}$ and $\mathbb{P}^u$, respectively[2] [285, 286].

Sampling problems

In various applications one is interested in the computation of expectation values of path functionals related to diffusions. They can provide an 'average' behavior of certain observables connected to the dynamics and thereby represent an important characteristic of the underlying stochastic process. We consider expectations that are of the form

$$\mathcal{Z} = \mathbb{E}\left[\exp(-\mathcal{W}(X))\right], \tag{1.7}$$

[1] Most of the upcoming analysis can be undertaken when considering an m-dimensional Brownian motion, where then $\sigma : \mathbb{R}^d \times [0, T] \to \mathbb{R}^{d\times m}$. However, for notational convenience and potential issues that may arise in the importance sampling or optimal control problems we shall focus on the case $m = d$.

[2] Of course, we have that $\mathbb{P}^0$ coincides with the path measure associated to the uncontrolled dynamics, i.e. $\mathbb{P}^0 = \mathbb{P}$.

where the *work functional* $\mathcal{W} : \mathcal{C} \to \mathbb{R}$ is defined[3] as

$$\mathcal{W}(X) = \int_0^T f(X_s, s)\,\mathrm{d}s + g(X_T), \tag{1.8}$$

for suitable functions $f \in C^1(\mathbb{R}^d \times [0,T], [0,\infty))$ and $g \in C^1(\mathbb{R}^d, \mathbb{R})$. The exponential form in (1.7) constraints the expectation to be positive, which will turn out to be helpful for dualities that we shall discuss later on (cf. Section 2.3). In statistical physics the expectation value (1.7) appears in a quantity called *free energy* [130], which, associated to the dynamics (1.2) and the work functional (1.8), can be defined as

$$-\log \mathbb{E}\left[\exp(-\mathcal{W}(X))\right] = -\log \mathcal{Z}. \tag{1.9}$$

In almost all interesting scenarios the free energy cannot be computed analytically and one usually relies on the Monte Carlo method to yield appropriate approximations [275]. However, quite often, the variance associated to the random variable $\exp(-\mathcal{W}(X))$ is so large as to render direct estimation of the expectation $\mathbb{E}\left[\exp(-\mathcal{W}(X))\right]$ computationally infeasible[4]. A natural approach is then to use the identity

$$\mathbb{E}\left[\exp(-\mathcal{W}(X))\right] = \mathbb{E}\left[\exp(-\mathcal{W}(X^u))\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^u}(X^u)\right], \qquad u \in \mathcal{U}, \tag{1.10}$$

where we recall that X and X^u refer to the strong solutions of (1.2) and (1.4), respectively, and $\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^u}$ denotes the Radon-Nikodym derivative, explicitly given by Girsanov's theorem[5] (Theorem B.3),

$$\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^u} = \exp\left(-\int_0^T u(X_s^u, s)\cdot \mathrm{d}W_s - \frac{1}{2}\int_0^T |u(X_s^u, s)|^2\,\mathrm{d}s\right), \tag{1.11}$$

see the proof of Theorem 1.2. As will be explained in more detail in Section 2.3.2, techniques leveraging (1.10) may be thought of as instances of importance sampling on path space, where the idea is to sample from another measure and weight back with a corresponding likelihood ratio in order to still get an unbiased estimator – we will further elaborate on statistical consequences of this attempt in Chapter 3.

Given that (1.10) holds for all $u \in \mathcal{U}$, it is clearly desirable to choose the control such as to guarantee favorable statistical properties. This is formulated as our first problem.

Problem 1.1 (Variance minimization)**.** *Find $u^* \in \mathcal{U}$ such that*

$$\mathrm{Var}\left(\exp(-\mathcal{W}(X^{u^*}))\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^{u^*}}(X^{u^*})\right) = \inf_{u\in\mathcal{U}} \mathrm{Var}\left(\exp(-\mathcal{W}(X^u))\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^u}(X^u)\right). \tag{1.12}$$

Under suitable conditions, it turns out that there exists $u^* \in \mathcal{U}$ such the variance expression (1.12) is in fact zero (see Theorem 1.2, (1d), Theorem 2.33 and Proposition B.6), therefore providing a perfect sampling scheme. At the same time, it is known that choices $u \neq u^*$ can potentially increase the variance of an importance sampling estimator significantly [32, 108, 206, 272]. Let us already anticipate that it seems to be not obvious how other choices of u influence the variance exactly and that finding a robust u seems to be non-trivial. Chapter 3 will provide some quantitative analysis on this aspect.

Conditioning and rare events

Sampling gets particularly challenging when considering so-called rare events, which are characterized by the fact that they only occur with a very small probability, usually obeying some exponential decay (see also Section 2.3.3). In diffusions this phenomenon often appears if the dynamics exhibits some metastable behavior (cf. Example 1.1). Any efficient sampling strategy should have the goal of making the event of interest commonly observable. Ideally, we want to condition the dynamics on this event such that it can be observed easily, making a statistically sound estimation of corresponding observables possible. This goal of conditioning a dynamics can

[3] As already hinted at, later we will as well replace the deterministic time horizon T with some random stopping time.

[4] In fact, the variance is particularly large in metastable scenarios such as those to be sketched in Example 1.1.

[5] By a slight abuse of notation, (1.11) is to be interpreted as a random variable on Ω provided by the measurable map $\omega \mapsto X^u$ induced by (1.4). In other words, the left-hand side should be read as $\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^u}(X^u(\omega))$.

be formalized in terms of weighted measures on path space. Consider again the work functional $\mathcal{W}$, as defined in (1.8). It induces a *reweighted*[6] path measure $\mathbb{Q}$ on $\mathcal{C}$ via

$$\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}} = \frac{e^{-\mathcal{W}}}{\mathcal{Z}}, \tag{1.13}$$

assuming f and g are such that $\mathcal{Z}$ is finite (we shall tacitly make this assumption from now on). We may ask whether $\mathbb{Q}$ can be obtained as the path measure related to a controlled SDE of the form (1.4). This leads to our second problem.

Problem 1.2 (Conditioning)**.** *Find $u^* \in \mathcal{U}$ such that the path measure $\mathbb{P}^{u^*}$ associated to* (1.4) *coincides with* $\mathbb{Q}$.

Referring to the above as a conditioning problem is justified by the fact that (1.13) may be viewed as an instance of Bayes' formula relating conditional probabilities, where $\mathbb{P}$ can be interpreted as a prior measure and $e^{-\mathcal{W}}$ as a likelihood function, yielding the posterior measure $\mathbb{Q}$ [246]. This connection can be formalized using Doob's h-transform [75, 76] and applied to diffusion bridges and quasistationary distributions, for instance (see Appendix B.10 and [55]).

Let us now present a guiding example which encompasses a stereotypical rare event scenario in a metastable diffusion, demonstrating the potential sampling challenges.

Example 1.1 (Rare event simulation in metastable diffusion)**.** *Let us consider SDEs of the form* (1.2)*, where the drift is a gradient, i.e. $b = -\nabla\Psi$, and the potential Ψ is of multimodal type, yielding the overdamped Langevin equation*

$$\mathrm{d}X_s = -\nabla\Psi(X_s)\,\mathrm{d}s + \sigma(X_s, s)\,\mathrm{d}W_s. \tag{1.14}$$

As an example we shall discuss the one-dimensional case $d = 1$ and assume that $\Psi \in C^\infty(\mathbb{R})$ is given by

$$\Psi(x) = \kappa(x^2 - 1)^2, \tag{1.15}$$

with $\kappa > 0$. Furthermore, let us fix the initial conditions $x_{\mathrm{init}} = -1$ and $t = 0$, and assume a constant diffusion coefficient of size unity, $\sigma = 1$. Observe that Ψ exhibits two local minima at $x = \pm 1$, separated by a barrier at $x = 0$, the height of which is modulated by the parameter κ. When κ is sufficiently large, the dynamics induced by (1.2) *exhibits a metastable behavior: transitions between the two basins happen very rarely as the transition time grows exponentially in the height of the barrier [26, 178] (see also Example 3.13).*

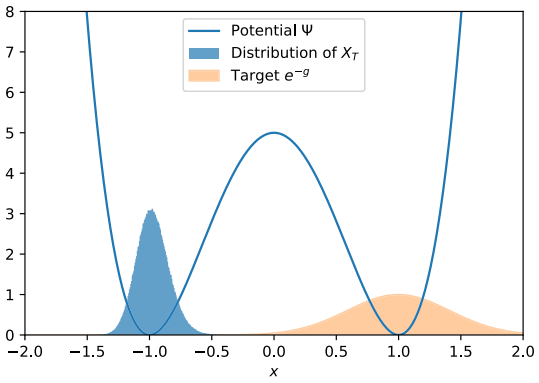


Figure 1.1: Illustration of rare events in a metastable double well potential.

Applications such as molecular dynamics are often concerned with statistics and derived quantities from these rare events, as those are typically directly linked to biological functioning [84, 264, 265]. At the same time, computational approaches face a difficult sampling problem as transitions are hard to obtain by direct simulation from (1.2)*. Choosing $f = 0$ and g such that e^{-g} is concentrated around $x = 1$ (consider, for instance, $g(x) = \nu(x-1)^2$ with $\nu > 0$ sufficiently large), we see that $\mathbb{Q}$ as defined in* (1.13) *predominantly charges paths initialized in $x = -1$ at $t = 0$ and enter a neighbourhood of $x = 1$ at final time T. Since in practice the distribution of X_T (corresponding to the path measure $\mathbb{P}$ in this particular case) often barely overlaps with the function e^{-g}, as*

[6] The reweighting formula (1.13) can sometimes be found under the name Feynman-Kac formula [64], not to be confused with the Feynman-Kac formula from Theorem 2.14, and it builds the foundation of so-called resampling schemes that aim at heuristic strategies for improving statistical sampling properties [77].

illustrated in Figure 1.1, the identification of $\mathbb{Q}$ might be challenging. Problem 1.2 can therefore be understood as the task of finding a control u that allows efficient simulation of transition paths. Similar issues arise in the context of stochastic filtering, where the objective is to sample paths that are compatible with available data [246].

Example 1.1 can be understood as a guiding problem, to which we will come back repeatedly throughout this thesis. Next, we will move towards a different perspective.

Optimal control of diffusions

Usually the problem of finding *good* controls in SDEs such as (1.4) is addressed in the theory of stochastic optimal control. Here, a certain objective that one wants to optimize has to be specified. We can for instance consider the cost functional

$$J(u; x_{\text{init}}, t) = \mathbb{E}\left[\int_t^T \left(f(X_s^u, s) + \frac{1}{2}|u(X_s^u, s)|^2\right) \mathrm{d}s + g(X_T^u)\middle| X_t^u = x_{\text{init}}\right], \tag{1.16}$$

where $f \in C^1(\mathbb{R}^d \times [t,T], \mathbb{R})$ specifies a part of the running costs, $g \in C^1(\mathbb{R}^d, \mathbb{R})$ specifies the terminal costs, and $(X_s^u)_{t\le s\le T}$ denotes the unique strong solution to the controlled SDE (1.4) with initial condition $X_t^u = x_{\text{init}}$. Throughout we assume that f and g are such that the expectation in (1.16) is finite, for all $(x_{\text{init}}, t) \in \mathbb{R}^d \times [0,T]$. Our objective is to find a control $u \in \mathcal{U}$ that minimizes (1.16):

Problem 1.3 (Optimal control)**.** *For $(x_{\text{init}}, t) \in \mathbb{R}^d \times [0,T]$, find $u^* \in \mathcal{U}$ such that*

$$J(u^*; x_{\text{init}}, t) = \inf_{u\in\mathcal{U}} J(u; x_{\text{init}}, t). \tag{1.17}$$

The specific form of the cost functional (1.16) is chosen with care and it will turn out that it is directly linked to our importance sampling expression (1.10).

There exist many strategies to actually solve the optimal control Problem 1.3 and we will come back to some of them later. For now, let us specify a quantity that will lead to yet another perspective. We define the *value function* as the 'optimal cost-to-go' [98, Section I.4], namely

$$V(x,t) = \inf_{u\in\mathcal{U}} J(u; x, t), \tag{1.18}$$

noting that it is a function of the initial time and the initial condition. A classic result in control theory (to be detailed in Section 2.1) shows that this function can be characterized as the solution to a specific parabolic partial differential equation, which motivates our fourth perspective.

High-dimensional PDEs

It is well-known that under suitable conditions, V satisfies a Hamilton-Jacobi-Bellman PDE involving the infinitesimal generator [230, Section 2.3] associated to the uncontrolled SDE (1.2),

$$L = \frac{1}{2}\sum_{i,j=1}^d (\sigma\sigma^\top)_{ij}(x,t)\partial_{x_i}\partial_{x_j} + \sum_{i=1}^d b_i(x,t)\partial_{x_i}. \tag{1.19}$$

The optimal control solving (1.17) can then be recovered from $u^* = -\sigma^\top \nabla V$ (we will specify details later, e.g. in Theorem 1.2). Let us state this reformulation of Problem 1.3 as follows:

Problem 1.4 (Hamilton-Jacobi-Bellman PDE)**.** *Find a solution V to the PDE* [7]

$$(\partial_t + L)V(x,t) - \frac{1}{2}|\sigma^\top \nabla V(x,t)|^2 + f(x,t) = 0, \qquad (x,t) \in \mathbb{R}^d \times [0,T), \tag{1.20a}$$

$$V(x,T) = g(x), \qquad x \in \mathbb{R}^d, \tag{1.20b}$$

where f and g are as in (1.16)*.*

Throughout, we will focus on solutions to (1.20) that admit bounded and continuous derivatives of up to first order in time and second order in space (see, however, Remark 1.3). This set will be denoted by $C_b^{2,1}(\mathbb{R}^d \times [0,T], \mathbb{R})$. Note that we have now connected our probabilistic Problems 1.1-1.3 to a solely deterministic one in Problem 1.4. Throughout this thesis, we will repeatedly encounter these two sides of the same matter and the connection shall be further discussed in Section 2.2. Finally, our last problem brings yet another stochastic perspective.

[7] Here and throughout this thesis, we slightly shorten notation by writing $\sigma^\top \nabla V(x,t)$ instead of $\sigma^\top(x,t)\nabla V(x,t)$.

Forward-backward SDEs

Solutions to elliptic and parabolic PDEs admit probabilistic representations by means of the celebrated Feynman-Kac formulae and their nonlinear extensions. To wit, we consider the following coupled system of forward-backward SDEs (in the following FBSDEs for short):

Problem 1.5 (Forward-backward SDEs)**.** *For $(x_{\text{init}}, t) \in \mathbb{R}^d \times [0,T]$, find progressively measurable stochastic processes $Y : \Omega \times [t,T] \to \mathbb{R}$ and $Z : \Omega \times [t,T] \to \mathbb{R}^d$ such that*

$$\mathrm{d}X_s = b(X_s, s)\,\mathrm{d}s + \sigma(X_s, s)\,\mathrm{d}W_s, \qquad X_t = x_{\text{init}}, \tag{1.21a}$$

$$\mathrm{d}Y_s = -f(X_s, s)\,\mathrm{d}s + \frac{1}{2}|Z_s|^2\,\mathrm{d}s + Z_s \cdot \mathrm{d}W_s, \qquad Y_T = g(X_T), \tag{1.21b}$$

almost surely.

Y is called a backward process due to its specified terminal condition and its requirement to be progressively measurable indicates that it shall not be confused with a time-reversed process. Under suitable conditions, Itô's formula implies that it is connected to the value function V as defined in (1.18) via $Y_s = V(X_s, s)$. Similarly, Z is connected to the optimal control u^* through $Z_s = -u^*(X_s, s) = \sigma^\top \nabla V(X_s, s)$. We refer to [228, 229], Theorem 1.2 and in particular to Section 2.2.2 for further details and will later see that this perspective will open the door to Monte Carlo strategies for quite a wide range of problems, leading to computationally feasible methods especially in higher dimensions.

1.4 Connections and equivalences

We have so far stated five perspectives, some of which coming from seemingly different areas of mathematics. Let us now elaborate on their relations to one another. The following theorem shows that all of the above problems are intimately connected.

Theorem 1.2 (Connections and equivalences)**.** *The following holds:*

1. *Let $V \in C_b^{2,1}(\mathbb{R}^d \times [0,T], \mathbb{R})$ be a solution to Problem 1.4, i.e. solve the HJB PDE* (1.20)*. Set*

$$u^* = -\sigma^\top \nabla V. \tag{1.22}$$

   Then

   (a) *the control u^* provides a solution to Problem 1.3, i.e. u^* minimizes the objective* (1.16)*,*

   (b) *the pair*

$$Y_s = V(X_s, s), \qquad Z_s = \sigma^\top \nabla V(X_s, s) \tag{1.23}$$

   solves the FBSDE (1.21)*, i.e. Problem 1.5,*

   (c) *the measure $\mathbb{P}^{u^*}$ associated to the controlled SDE* (1.4) *coincides with $\mathbb{Q}$, i.e. u^* solves Problem 1.2,*

   (d) *the control u^* provides the minimum-variance estimator in* (1.12)*, i.e. u^* solves Problem 1.1. Moreover, the variance is zero, i.e. the random variable*

$$\exp(-\mathcal{W}(X^{u^*}))\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^{u^*}}(X^{u^*}) \tag{1.24}$$

   is almost surely constant.

   Furthermore, we have that

$$J(u^*; x_{\text{init}}, 0) = V(x_{\text{init}}, 0) = Y_0 = -\log \mathcal{Z}. \tag{1.25}$$

2. *Conversely, let $u^* \in \mathcal{U}$ solve Problem 1.2, i.e. assume that $\mathbb{P}^{u^*}$ coincides with $\mathbb{Q}$. Then statement* (1d) *holds. Furthermore, setting*

$$Y_0 = -\log \mathcal{Z}, \qquad Z_s = -u^*(X_s, s), \tag{1.26}$$

   solves the backward SDE (1.21b) *from Problem 1.5, i.e.* (1.26) *together with the first equation in* (1.21b) *determines a process $(Y_s)_{0 \le s \le T}$ that satisfies the final condition $Y_T = g(X_T)$, almost surely.*

We will later extend the connections between the optimal control formulation (Problem 1.3) and FBSDEs (Problem 1.5) in Proposition 4.19, see also Remark 4.20.

Remark 1.3 (Regularity, uniqueness, and further connections). Going beyond classical solvability of the HJB PDE (1.20) and introducing the notion of *viscosity solutions* [98, 228], the strong regularity and boundedness assumptions on V in the first statement could be relaxed and the connections exposed in Theorem 1.2 could be extended [234, 304]. As a case in point, we note that in the current setting, neither a solution to Problem 1.3 nor to Problem 1.5 necessarily provides a classical solution to the PDE (1.20), as optimal controls are known to be non-differentiable, in general.
However, assuming classical well-posedness of the HJB PDE (1.20), Theorem 1.2 implies that the solution can be found by addressing one of the Problems 1.1, 1.2, 1.3 or 1.5 and using the formulas (1.22) and (1.23), as long as those problems admit *unique* solutions, in an appropriate sense. For the latter issue, we refer the reader to [174] and [281, Chapter 11] in the context of forward-backward SDEs and to [33] in the context of measures on path space. We note that in particular the forward SDE (1.21a) can be thought of as providing a random grid for the solution of the HJB PDE (1.20), obtained through the backward SDE (1.21b) (cf. also Section 6.3).

Remark 1.4 (Random initial conditions). The equivalence between Problems 1.4 and 1.5 shows that u^* does not depend on x_{init}. Consequently, the initial condition in (1.21a) can be random rather than deterministic. In Section 4.4.3 we demonstrate potential benefit of this extension for FBSDE-based algorithms.

Remark 1.5 (Variational formulas and duality). The identities (1.25) connect key quantities pertaining to the problem formulations 1.2, 1.3, 1.4 and 1.5. The fact that $J(u^*; x_{\text{init}}, 0) = -\log \mathcal{Z}$ can moreover be understood in terms of the Donsker-Varadhan formula [40], furnishing an explicit expression for the value function,

$$V(x,t) = -\log \mathbb{E}\left[\exp\left(-\int_t^T f(X_s,s)\,\mathrm{d}s - g(X_T)\right)\middle| X_t = x\right], \tag{1.27}$$

as discussed in [60, 61, 130] and Section 2.3.4.

Proof of Theorem 1.2. The statement (1a) is a classical result in stochastic optimal control theory, often referred to as a *verification theorem*, and can for instance be found in [98, Theorem IV.4.4] or [234, Theorem 3.5.2]. We will also state it in full generality in Theorem 2.4. The implication (1b) is a direct consequence of Itô's formula, cf. [234, Proposition 6.3.2], [46, Proposition 2.14] or Theorem 2.25. Before proceeding to (1c), we note that the first equality in (1.25) now follows from (1.18) (for background, see [98, Section IV.2]), while the second equality is a direct consequence of (1b). Using (1.21) and (1b), the third equality follows from

$$\mathcal{Z} = \mathbb{E}\left[\exp(-\mathcal{W}(X)\right] = \exp(-Y_0)\,\mathbb{E}\left[\exp\left(\int_0^T u^*(X_s,s)\cdot \mathrm{d}W_s - \frac{1}{2}\int_0^T |u^*(X_s,s)|^2 \mathrm{d}s\right)\right] = \exp(-Y_0), \tag{1.28}$$

relying on the facts that Y_0 is deterministic (again using (1b)), and that the term inside the second expectation is a martingale (as u^* is assumed to be bounded). Turning to (1c), let us define an equivalent measure $\widetilde{\Lambda}$ on $(\Omega, \mathcal{F})$ via

$$\frac{\mathrm{d}\widetilde{\Lambda}}{\mathrm{d}\Lambda} = \exp\left(\int_0^T u^*(X_s,s)\cdot \mathrm{d}W_s - \frac{1}{2}\int_0^T |u^*(X_s,s)|^2\,\mathrm{d}s\right). \tag{1.29}$$

Since u^* is assumed to be bounded, Novikov's condition is satisfied, and hence Girsanov's theorem asserts that the process $(\widetilde{W}_t)_{0\le t\le T}$ defined by

$$\widetilde{W}_t = W_t - \int_0^t u^*(X_s,s)\,\mathrm{d}s \tag{1.30}$$

is a Brownian motion with respect to $\widetilde{\Lambda}$. Consequently, we have that

$$\frac{\mathrm{d}\mathbb{P}^{u^*}}{\mathrm{d}\mathbb{P}}(X(\omega)) = \frac{\mathrm{d}\widetilde{\Lambda}}{\mathrm{d}\Lambda}(\omega) = \exp\left(Y_0 - \mathcal{W}(X(\omega))\right) = \frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}(X(\omega)), \qquad \omega\in\Omega, \tag{1.31}$$

using (1.21) and (1.25) in the last step. We note that similar arguments can be found in [165], [47, Section 3.3.1].
For the proof of (1d) we refer to Theorem 2.33. The proof of the second statement is very similar to the argument presented for (1c), resting primarily on (1.29) and (1.31), and is therefore omitted. □

1.5 Related work and generalizations

The numerical treatment of Problems 1.1-1.5 has been addressed multiple times, relying an various different approaches. In this section we shall provide a literature overview and discuss potential generalizations.

Problem 1.1 motivates minimizing the variance of estimators via importance sampling, which is a classic variance reduction method in Monte Carlo simulation and introductions can be found in many textbooks, such as in [224, Section 9] or [106, 198], however, mostly the finite-dimensional case in $\mathbb{R}^d$ is treated. The non-robustness of importance sampling in high dimensions is well known and has often been observed in numerical experiments [32, 108, 206, 272]. Recently, the authors of [51] have proved that the sample size required for importance sampling to be accurate scales exponentially in the KL divergence between the proposal and the target measure, when accuracy is understood in the sense of the L^1 error, rather than the commonly used relative error. (Clearly, an unbounded L^1 error implies that the relative error will be unbounded.) Similar results can be found in [3], in which the authors analyze a self-normalized importance sampling estimator, in connection with inverse problems and filtering. Necessary conditions that any importance sampling proposal distribution has to satisfy have been derived in [260], using the more general f-divergences and adopting an information-theoretic perspective.

An important class of techniques for building proposal distributions is known by the name *sequential importance sampling*, where we recommend [77] for a comprehensive review. Closely related are methods based on interacting particle systems and nonlinear (mean-field) Feynman-Kac semigoups, in which the variance is controlled by adaptively annihilating and generating particles to approximate good proposal distributions [65]. Adaptive importance sampling for rare events simulation has been pioneered in [81, 82], based on exponential change of measure techniques and the theory of large deviations and going back to the seminal work [270]. For diffusion processes, large deviation principles can be used to approximate the optimal change of measure in the small noise regime, where the resulting change of measure turns out to be be asymptotically optimal [274, 290]. Similarly, [80] studies potential problems that appear when an optimal importance sampling proposal is not available, in particular when the time horizon of the problem is large. A non-asymptotic variant of the aforementioned approaches for finite noise diffusions is based on the stochastic control formulation of the optimal change of measure [127, 130]. Furthermore we should note that there have been many attempts to find good (low-dimensional) proposals by taking advantage of specific structures of the problem at hand, using simplified models that approximate a complicated multiscale system [79, 128, 133, 273]. Recently, the scaling properties of certain approximations to control-based importance sampling estimators with the system dimension have been analyzed in [217], suggesting that the empirical loss function that is used to numerically approximate the optimal proposal distribution is essential.

For a recent numerical attempt to approach variance minimization based on neural networks we refer the reader to [213, Section 5.2], for a theoretical analysis of convergence rates we refer to [4], and for a general overview regarding adaptive importance sampling techniques we refer to [44]. The relationship between optimal control and importance sampling (see Theorem 1.2) has been exploited by various authors to construct efficient samplers [161, 279], in particular also with a view towards the sampling based estimation of hitting times, in which case optimal controls are governed by elliptic rather than parabolic PDEs [126, 127, 131, 132]. Similar sampling problems have been addressed in the context of sequential Monte Carlo [66, 136] and generative models [283, 284]. The latter works examine the potential of the controlled SDE (1.4) as a sampling device targeting a suitable distribution of the final state X_T^u.

Conditioned diffusions (Problem 1.2) have been considered in a large deviation context [81] as well as in a variational setting [127, 130] motivated by free energy computations, building on earlier work in [40, 61], see also [9, 55, 60, 95]. The simulation of diffusion bridges has been studied in [207] and conditioning via Doob's h-transform has been employed in a sequential Monte Carlo context [136]. The formulation in Problem 1.2 identifies the target measure $\mathbb{Q}$, motivating approaches that seek to minimize certain divergences on path space. This perspective will be developed in detail in Section 4.1.1, building bridges to Problems 1.1, 1.3, 1.4 and 1.5. Prior work following this direction includes [33, 113, 131, 160, 245], in particular relying on a connection between the KL-divergence (or relative entropy) on path space and the cost functional (1.16), see also Proposition 4.7. A similar line of reasoning leads to the *cross-entropy method* [130, 161, 255, 308], see Proposition 4.9 and equation (4.32) in Section 4.1.3.

The numerical treatment of optimal control problems has been an active area of research for many decades and multiple perspectives on solving Problem 1.3 have been developed. The monographs [30] and [182] provide good overviews to *policy iteration* and *Q-learning*, strategies that have been further investigated in the machine learning literature and that are generally subsumed under the term *reinforcement learning* [238]. We also recommend [159] as an introduction to the specific setting considered in this thesis. To cope with the key issue

of high dimensionality, the authors of [222] suggest solving a certain type of control problem in the framework of hierarchical tensor products. Another strategy of dealing with the curse of dimensionality is to first apply a model reduction technique and only then solve for the reduced model. Here, recent results on *balanced truncation* for controlled linear S(P)DEs have for instance been suggested in [17], and approaches for systems with a slow-fast scale separation via the *homogenization* method can be found in [128].

Solutions to Problem 1.4, i.e. to HJB PDEs of the type (1.20), can be approximated through finite difference or finite volume methods [2, 218, 233]. However, these approaches are usually not applicable in high-dimensional settings. In contrast, the recently introduced *Multilevel Picard* method [145] based on a combination of the Feynman-Kac and Bismut-Elworthy-Li formulas has been proven to beat the curse of dimensionality in a variety of settings, see [14, 146, 148, 149, 150].

The FBSDE formulation (Problem 1.5) has opened the door for Monte Carlo based methods that have been developed since the early 90s. We mention in particular *least-squares Monte Carlo*, where $(Z_s)_{0\leq s\leq T}$ is approximated iteratively backwards in time by solving a regression problem in each time step, along the lines of the dynamic programming principle [234, Chapter 3]. A good introduction can be found in [109]; for extensive analysis on numerical errors we refer the reader to [110, 306]. Recently, this approach has also been connected with deep learning, replacing Galerkin approximations by neural networks [144] (see also Section 6.2).

Another method leveraging the FBSDE perspective has been put forward in [86, 122] and further developed in [12, 13]. Here, the main idea is to enforce the terminal condition $Y_T = g(X_T)$ in (1.21b) by iteratively minimizing the loss function

$$\mathcal{L}(u, y_0) = \mathbb{E}\left[(Y_T(y_0, u) - g(X_T))^2\right], \tag{1.32}$$

using a stochastic gradient descent scheme (see also Definition 6.17). The notation $Y_T(y_0, u)$ indicates that the process in (1.21b) is to be simulated with given initial condition y_0 and control u (these representing a priori guesses or current approximations, typically relying on neural networks), hence viewing (1.21b) as a forward process. Consequently, the approach thus described can be classified as a *shooting method* for boundary value problems. We note that this idea allows treating rather general parabolic and elliptic PDEs [118, 147], as well as – with some modifications – optimal stopping problems [15, 16]. Using neural network approximations in conjunction with FBSDE-based Monte-Carlo techniques holds the promise of alleviating the curse of dimensionality; understanding this phenomenon and proving rigorous mathematical statements has been been the focus of intense current research [29, 118, 119, 147, 157]. Let us also mention that similar algorithms have been suggested in [241, 242], in particular proposing to modify the loss function (1.32) in order to encode the backward dynamics (1.21b), and extensive investigation of optimal network design and choice of tunable parameters has been carried out [50]. Furthermore, we refer to [48, 49] for convergence results in the broader context of mean field control. In [127, Section III.B] it has been proposed to modify the forward dynamics (1.21a) (and, to compensate, also the backward dynamics (1.21b)) by an additional control term. This idea is central for some main results of this thesis, see Section 4.1.2. Similar ideas for other types of PDEs have been proposed as well, see for instance [89, 242]. We refer to Section 6.3 for variational approaches on more general parabolic and elliptic PDEs.

Generalizations

The problem formulations 1.3, 1.4 and 1.5 admit generalizations that keep parts of the connections expressed in Theorem 1.2 intact. From the PDE-perspective (Problem 1.4), it is possible to consider more general nonlinearities,

$$(\partial_t + L)V(x,t) + h(x,t,V(x,t),\sigma^\top \nabla V(x,t)) = 0, \qquad (x,t) \in \mathbb{R}^d \times [0,T), \tag{1.33a}$$

$$V(x,T) = g(x), \qquad x \in \mathbb{R}^d, \tag{1.33b}$$

with h being a function satisfying appropriate regularity and boundedness assumptions. As in Theorem 1.2 (1b), the nonlinear parabolic PDE (1.33) is related to a generalization of the forward-backward system (1.21),

$$\mathrm{d}X_s = b(X_s, s)\,\mathrm{d}s + \sigma(X_s, s)\,\mathrm{d}W_s, \qquad X_t = x_{\text{init}}, \tag{1.34a}$$

$$\mathrm{d}Y_s = -h(X_s, s, Y_s, Z_s)\,\mathrm{d}s + Z_s \cdot \mathrm{d}W_s, \qquad Y_T = g(X_T), \tag{1.34b}$$

where the connection is still given by (1.23), see [234, Section 6.3] and Section 2.2.2. From the perspective of optimal control (Problem 1.3), it is possible to extend the discussion to SDEs of the form

$$\mathrm{d}X_s^u = \widetilde{b}(X_s^u, s, u_s)\,\mathrm{d}s + \widetilde{\sigma}(X_s^u, s, u_s)\,\mathrm{d}W_s, \tag{1.35}$$

replacing (1.4), and to running costs $\widetilde{f}(X_s^u, u_s, s)$ instead of $f(X_s^u, s) + \frac{1}{2}|u(X_s^u, s)|^2$ in (1.16), assuming that u_s has values in $U \subset \mathbb{R}^m$, for some $m \in \mathbb{N}$. This setting gives rise to more general HJB PDEs,

$$\partial_t V(x,t) + H(x,t,\nabla V(x,t), \nabla^2 V(x,t)) = 0, \tag{1.36}$$

where $\nabla^2 V$ denotes the Hessian of V, and the *Hamiltonian* H is given by

$$H(x,t,p,A) = \inf_{u \in U} \left\{ \widetilde{b}(x,t,u) \cdot p + \tfrac{1}{2} \operatorname{Tr}(\widetilde{\sigma}\widetilde{\sigma}^\top A)(x,t,u) + \widetilde{f}(x,t,u) \right\}, \tag{1.37}$$

see [98, 234] and Section 2.1. In certain scenarios [307, Section 4.5.2], it is then possible to relate (1.36) to (1.34), noting however that typically h will be given in terms of a minimization problem as in (1.37). The relationship to Problems 1.1 and 1.2 as well as the identity (1.22) rest on the particular structure[8] inherent in (1.4) and (1.16), enabling the use of Girsanov's theorem (see the Proof of Theorem 1.2). The methods that will be developed in this thesis can straightforwardly be extended to equations of the form (1.33) in the case when h depends on V only through ∇V, owing to the invariance of the PDE under shifts of the form $V \mapsto V + \text{const.}$, see Remark 4.14. In order to address optimal control problems involving additional minimization tasks posed by Hamiltonians such as (1.37) it might be feasible to include appropriate penalty terms in the loss functional. We leave this direction for future work.

[8] Note that this structure connects the PDEs (1.36) and (1.20) in view of $H(x,t,\nabla V, \nabla^2 V) = LV + f + \min_{u \in U} \left\{ \sigma u \cdot \nabla V + \frac{1}{2}|u|^2 \right\}$ and $\min_{u \in U} \left\{ \sigma u \cdot \nabla V + \frac{1}{2}|u|^2 \right\} = -\frac{1}{2}|\sigma^\top \nabla V|^2$.

Chapter 2

Theoretical foundations

This thesis combines topics from different areas of mathematics. Some of the connections between them are obvious, some of them rather non-standard. This chapter shall foster some of the relations, while providing a solid theoretical foundation on the concepts that are used throughout the thesis. We will focus on key concepts and central theorems and note that this collection is by no means exhaustive. For most topics, we follow the strategy of first providing some rather formal and non-technical introduction with the purpose to gain helpful intuition, while in the theorems we will be precise up to an extend that does not complicate our intentions too much. For further technical subtleties we will try to provide corresponding references for the interested reader.

This chapter is organized as follows. In Section 2.1 we will provide an introduction to stochastic optimal control theory, focusing on the dynamic programming principle and the Hamilton-Jacobi-Bellman PDE as an equation that brings a deterministic viewpoint into a stochastic problem. In this spirit, Section 2.2 deals with the stochastic representation of (deterministic) PDEs, where Section 2.2.1 treats the special case of linear PDEs and states the celebrated Feynman-Kac theorem, while Section 2.2.2 introduces FBSDEs as a means to represent nonlinear PDEs. Here we recall some existence and uniqueness results, before in Section 2.2.3 we deal with the numerical discretization of SDEs and BSDEs. Section 2.3.1 then brings a proper introduction to importance sampling, first in an abstract setting, then focusing on importance sampling of diffusions in Section 2.3.2. Section 2.3.3 introduces the theory of large deviations and Section 2.3.4 recalls the Donsker-Varadhan variational formula, both having strong ties to importance sampling. Finally, in Section 2.4 we will define neural networks and briefly discuss stochastic optimization for targeting the approximation task.

2.1 Stochastic optimal control

We start with an introduction to the theory of stochastic optimal control, which deals with identifying optimal strategies in somewhat noisy environments, in our case time-continuous diffusion processes. Optimal control theory goes back to the 1950s with the dynamic programming principle as the key concept for characterizing optimality [19]. Even though simple and very intuitive in its formulation, it turns out that rigorous proofs are very technical and can be approached with many different methods [24, 180]. A main result is the Hamilton-Jacobi-Belmann PDE being the determining equation for optimality, thereby providing optimal control strategies. However solutions to optimal control problems often do not posses enough regularity in order to formally fulfill this equation, such that a complete theory of optimal control needs to introduce an appropriate concept of weak solutions, leading to so-called viscosity solutions that have been extensively studied starting in the 1980s (cf. Remark 2.7) [98, 197]. Even though being an interesting mathematical subject on its own, we shall not focus on this aspect. As stated before, the intention of this short introduction is to provide a good basic understanding and to state theorems that will be needed at later stages. For extensive introductions and further details we refer to the monographs [97, 98, 234, 289], on which this introduction is heavily based on.

In a general setting, stochastic optimal control problems consider controlled diffusions[9]

$$\mathrm{d}X_s^u = \widetilde{b}(X_s^u, s, u_s)\,\mathrm{d}s + \widetilde{\sigma}(X_s^u, s, u_s)\,\mathrm{d}W_s, \qquad X_t^u = x_{\mathrm{init}}, \tag{2.1}$$

for $s \in [t, T]$ on a filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_s)_{s\geq 0}, \Lambda)$, where $\widetilde{b}$ and $\widetilde{\sigma}$ are suitable functions and W is a d-dimensional Brownian motion. The control u_s shall be an element of some control set[10] $\mathcal{U}$ (as for instance defined

[9] We put tildes on top of certain quantities in order to avoid confusion with other quantities defined beforehand, as for instance in the controlled diffusion (1.4) that we have defined in the introduction.

[10] Note the difference of $\mathcal{U}$ being the set of progressively measurable control functions, and U being the set of the values u_s is allowed to take, i.e. $\mathcal{U} \ni u_s : \Omega \times [0, T] \to U$.

in (1.5)) taking values in $U \subset \mathbb{R}^d$ and being progressively measurable w.r.t. $(\mathcal{F}_s)_{s\geq 0}$, $\widetilde{b} : \mathbb{R}^d \times [0,T] \times U \to \mathbb{R}^d$ and $\widetilde{\sigma} : \mathbb{R}^d \times [0,T] \times U \to \mathbb{R}^{d\times d}$ shall be measurable and satisfy a uniform Lipschitz condition, such that for all $x, y \in \mathbb{R}^d, s \in [0,T], u \in U$ and a constant $\widetilde{L} > 0$ it holds

$$|\widetilde{b}(x,s,u) - \widetilde{b}(y,s,u)| + |\widetilde{\sigma}(x,s,u) - \widetilde{\sigma}(y,s,u)| \leq \widetilde{L}\,|x-y|. \tag{2.2}$$

Of course we assume the equation (2.1) to have a unique solution, which can for instance be guaranteed by considering controls u_s for which

$$\mathbb{E}\left[\int_0^T \left(|\widetilde{b}(\mathbf{0},s,u_s)|^2 + |\widetilde{\sigma}(\mathbf{0},s,u_s)|^2\right) \mathrm{d}s\right] < \infty. \tag{2.3}$$

A control that fulfills all the above properties is said to be *admissible* and we denote by $\mathcal{U}$ the set of admissible controls.

We now specify the control objective. To this end, let $\widetilde{f} : \mathbb{R}^d \times [0,T] \times U \to \mathbb{R}, g : \mathbb{R}^d \to \mathbb{R}$ be two measurable functions. We suppose that g is lower bounded and that it satisfies a quadratic growth condition, i.e.

$$|g(x)| \leq C(1 + |x|^2) \tag{2.4}$$

for all $x \in \mathbb{R}^d$ and come constant $C > 0$. We further demand

$$\mathbb{E}\left[\int_t^T |\widetilde{f}(X_s^u, s, u_s)| \mathrm{d}s \middle| X_t^u = x\right] < \infty \tag{2.5}$$

for all $(x,t) \in \mathbb{R}^d \times [0,T]$ and all $u_s \in \mathcal{U}$. We can then define the cost functional to be

$$J(u;x,t) = \mathbb{E}\left[\int_t^T \widetilde{f}(X_s^u, s, u_s)\mathrm{d}s + g(X_T^u) \middle| X_t^u = x\right], \tag{2.6}$$

where $\widetilde{f}$ specifies running and g represents terminal costs. The objective in optimal control is to minimize this quantity over all admissible control functions $u_s \in \mathcal{U}$, and we therefore introduce the so-called *value function*

$$V(x,t) = \inf_{u\in\mathcal{U}} J(u;x,t) \tag{2.7}$$

as the optimal costs conditioned on being in position x at time t.

Remark 2.1 (Time horizon). The above definitions operate on a deterministic time horizon. For two alternative variants of the optimal control problem we can either consider random time horizons, where T is replaced by a random stopping time τ, or infinite time horizons, where T can be infinitely large. In this work we shall discuss the two first options only and omit the infinite time horizon case[11]. In the sequel we will focus on the case of a deterministic T, having in mind that, given suitable assumptions, most statements readily transfer to the random time case. We will further elaborate on this aspect in Remark 2.9.

Often a control process is of the form[12] $u_s = u(X_s^u, s)$ for some measurable function $u : \mathbb{R}^d \times [0,T] \to U$, where we recall $U \subset \mathbb{R}^d$ to be the set of admissible control values. We call such a control *Markov control* since the process X_t^u as defined in (2.1) is a Markov process. In the sequel, we shall tacitly assume that the value function as defined in (2.7) is measurable in its arguments. This is actually not trivial a priori, and we refer to measurable section theorems for sufficient conditions (see [67, Chapter III, Appendix]).

Let us come back to our goal of identifying an optimal control u^* that minimizes the cost functional (2.6). The following section will lead us towards strategies to accomplish this goal.

[11] Note for instance that in the infinite time horizon case, the running control costs have to be decreasing in time and that the coefficients $\widetilde{b}, \widetilde{\sigma}$ cannot explicitly depend on time anymore in order to have a well-defined control problem.

[12] Note the slight abuse of notation here, as with $u_s \in \mathcal{U}$ we refer to a random path, whereas $u : \mathbb{R}^d \times [0,T] \to U$ is a deterministic function, to which randomness is added via the process X_s^u.

2.1.1 The dynamic programming principle

The so-called dynamic programming principle goes back to Richard Bellman [19]. Loosely speaking, it tries to approach the optimal control problem by subdividing it into multiple smaller problems, of which each is then solved separately. The following statement provides the key observation that we can relate the value function (2.7) at time t to the value function at any later time $t + \Delta t$ for $\Delta t \in [0, T - t]$.

Theorem 2.2 (Dynamic programming principle)**.** *Let $(x, t) \in \mathbb{R}^d \times [0, T]$, let $\widetilde{f}, u_s$ be as defined above and let V be the value function as defined in (2.7), it then holds for any $\Delta t \geq 0$ with $t + \Delta t \leq T$ that*

$$V(x,t) = \inf_{u \in \mathcal{U}} \mathbb{E}\left[\int_t^{t+\Delta t} \widetilde{f}(X_s^u, s, u_s)\mathrm{d}s + V(X_{t+\Delta t}^u, t + \Delta t) \middle| X_t^u = x \right]. \tag{2.8}$$

Proof. Let us gain some intuition with a formal derivation for the case of u_s being a Markov control. For the cost functional the relation

$$J(u; x, t) = \mathbb{E}\left[\int_t^{t+\Delta t} \widetilde{f}(X_s^u, s, u_s)\mathrm{d}s + J(u; X_{t+\Delta t}^u, t + \Delta t) \middle| X_t^u = x \right] \tag{2.9}$$

then follows immediately from the definition of J as in (2.6), the Markov property as well as the tower property of the conditional expectation. Let

$$u_s' := \begin{cases} u_s & s \in [0, t + \Delta t) \\ u_s^* & s \in [t + \Delta t, T] \end{cases}, \tag{2.10}$$

with u_s^* being the optimal control, which brings

$$V(x,t) \leq J(u'; x, t) = \mathbb{E}\left[\int_t^{t+\Delta t} \widetilde{f}(X_s^{u'}, s, u_s')\mathrm{d}s + V(X_{t+\Delta t}^{u'}, t + \Delta t) \middle| X_t^{u'} = x \right] \tag{2.11}$$

by the definition of the value function as in (2.7). Now choosing $u_s = u_s^*$ for $s \in [t, t + \Delta t)$ brings equality and therefore the desired relation. A more general derivation can be found in [98] and more technical details are provided in [24]. □

A consequence of the dynamic programming principle is that the optimization problem (2.7) can be split into two, or, by iterating the argument, into multiple optimization problems. Given a time grid $0 = t_0 < t_1 < \cdots < t_N = T$ and noting that $V(x, T) = g(x)$ is known, we can for instance first compute the optimal control on the time interval $[t_{N-1}, t_N]$, yielding the quantity $V(x, t_{N-1})$ for some x. This can then be used when computing the optimal control in the preceding time interval $[t_{N-2}, t_{N-1}]$ and so on. We will discuss more details on algorithmic approaches in this spirit for instance in Section 6.2.

Remark 2.3. We can note the following, more probabilistic point of view on the dynamic programming principle [289]. Let

$$M_t^u = \int_0^t \widetilde{f}(X_s^u, s, u_s)\mathrm{d}s + V(X_t^u, t), \tag{2.12}$$

then the dynamic programming principle says: M_t^u is always a submartingale, while it is a martingale for $u = u^*$.

A natural question is to study what happens if we let $\Delta t \to 0$ in (2.8). What will come out is a key statement of optimal control theory: a partial differential equation for the value function (2.7).

2.1.2 Hamilton-Jacobi-Bellman PDE

One can think of the Hamilton-Jacobi-Bellman (HJB) equation as the infinitesimal version of the dynamic programming principle that we have stated in Theorem 2.2. Loosely speaking, it describes the local behavior of the value function when the time increment Δt is sent to 0. Importantly, let us for now assume that V is sufficiently differentiable.

Theorem 2.4 (Hamilton-Jacobi-Bellman PDE, verification theorem)**.** *Let $V \in C^{2,1}(\mathbb{R}^d \times [0,T], \mathbb{R})$ fulfill the PDE*

$$\partial_t V(x,t) + \inf_{u \in U} \left\{ \widetilde{f}(x,t,u) + \widetilde{b}(x,t,u) \cdot \nabla V(x,t) + \frac{1}{2}(\widetilde{\sigma}\widetilde{\sigma}^\top)(x,t,u) : \nabla^2 V(x,t) \right\} = 0, \quad (x,t) \in \mathbb{R}^d \times [0,T), \tag{2.13a}$$

$$V(x,T) = g(x), \qquad x \in \mathbb{R}^d, \tag{2.13b}$$

such that $|V(x,t)| \leq C(1+|x|^2)$ for some $C > 0, (x,t) \in \mathbb{R}^d \times [0,T]$, and suppose there exists a measurable function $\mathcal{U} \ni u^* : \mathbb{R}^d \times [0,T] \to U$ that attains the above infimum. Let further the corresponding controlled SDE X^{u^*} have a strong solution. Then V coincides with the value function as defined in (2.7) and u^* is an optimal Markovian control.

Proof. A proof can for instance be found in [234, Theorem 3.5.2]. In order to gain some intuition let us provide a formal derivation of the HJB equation. To this end, we can apply Itô's formula to the value function V to get

$$V(X^u_{t+\Delta t}, t+\Delta t) = V(X^u_t, t) + \int_t^{t+\Delta t} (\partial_s + L^u) V(X^u_s, s) \mathrm{d}s + \int_t^{t+\Delta t} \sigma^\top \nabla V(X^u_s, s) \cdot \mathrm{d}W_s, \tag{2.14}$$

where L^u is the infinitesimal generator of the controlled process defined by (2.1). Assuming that the Itô integral is a martingale we can take conditional expectation to get

$$V(X^u_t, t) = \mathbb{E}\left[- \int_t^{t+\Delta t} (\partial_s + L^u) V(X^u_s, s) \mathrm{d}s + V(X^u_{t+\Delta t}, t+\Delta t) \middle| X^u_t \right]. \tag{2.15}$$

Let us recall the dynamic programming principle (2.8), namely

$$V(X^{u^*}_t, t) = \mathbb{E}\left[\int_t^{t+\Delta t} \widetilde{f}(X^{u^*}_s, s, u^*{}_s) \mathrm{d}s + V(X^{u^*}_{t+\Delta t}, t+\Delta t) \middle| X^{u^*}_t \right], \tag{2.16}$$

which we can combine with (2.15) to get

$$\mathbb{E}\left[\int_t^{t+\Delta t} \left((\partial_s + L^{u^*}) V(X^{u^*}_s, s) + \widetilde{f}(X^{u^*}_s, s, u^*_s) \right) \mathrm{d}s \middle| X^{u^*}_t \right] = 0. \tag{2.17}$$

Now, dividing by Δt and letting $\Delta t \to 0$, we formally get

$$\partial_t V(x,t) + \inf_{u \in U} \{ L^u V(x,t) + \widetilde{f}(x,t,u) \} = 0, \tag{2.18}$$

which is the HJB equation as stated in (2.13a). The boundary condition (2.13b) follows from the definition of V as stated in (2.7).

□

Remark 2.5 (Semi-linear PDE). If the diffusion coefficient $\widetilde{\sigma}$ does not depend on the control then the PDE (2.13a) becomes semi-linear:

$$\left(\partial_t + \frac{1}{2}(\sigma\sigma^\top)(x,t) : \nabla^2 \right) V(x,t) + \inf_{u \in U} \left\{ \widetilde{f}(x,t,u) + \widetilde{b}(x,t,u) \cdot \nabla V(x,t) \right\} = 0, \tag{2.19a}$$

$$V(x,T) = g(x). \tag{2.19b}$$

Remark 2.6 (Pointwise optimization). Let us appreciate that the infimum in the HJB equation (2.13a) is merely over the set $U \subset \mathbb{R}^d$ and not over $\mathcal{U}$, i.e. entire paths as in (2.7), so the minimization reduces to a pointwise operation. For a sanity check, note that if u^* solves the optimal control problem then the HJB equation (2.13a) translates to

$$\partial_t V(x,t) + \widetilde{f}(x,t,u^*) + \widetilde{b}(x,t,u^*) \cdot \nabla V(x,t) + \frac{1}{2}(\widetilde{\sigma}\widetilde{\sigma}^\top)(x,t,u^*) : \nabla^2 V(x,t) = 0 \tag{2.20}$$

and the Feynman-Kac theorem (which will be stated in Theorem 2.14) brings that

$$V(x,t) = \mathbb{E}\left[\int_t^T \widetilde{f}(X^{u^*}_s, t, u^*) \mathrm{d}s + g(X^{u^*}_T) \middle| X^{u^*}_t = x \right], \tag{2.21}$$

as expected from (2.7).

Remark 2.7 (Viscosity solutions). Existence results for smooth solutions to parabolic PDEs of HJB type are provided in [97], [105] or [180]. Here, the main required condition is a uniform ellipticity condition. As already mentioned, often a solution to an optimal control problem is not in $C^{2,1}$ however and it is a priori not clear how to interpret the HJB equation (2.13a) in such a setting. For this, the notion of viscosity solutions as an adequate concept of weak solutions has been introduced and extensively studied in the last decades. Although being an interesting topic by itself, we will not elaborate any further and rather point to [98, 228] for details.

Remark 2.8 (Parabolic problem on bounded domain). Instead of considering the control problem on the entire $\mathbb{R}^d$, we can instead constrain it to a bounded set $\mathcal{D} \subset \mathbb{R}^d$. In this case we do not consider a fixed time horizon T, but let the dynamics evolve up to time $\tau \wedge T$, where $\tau = \inf\{t > 0 : X_t \notin \mathcal{D}\}$ is the random exit time from the domain. The control costs we consider are then given by

$$J(u; x, t) = \mathbb{E}\left[\int_t^{\tau\wedge T} \widetilde{f}(X_s^u, s, u_s)\mathrm{d}s + g(X_{\tau\wedge T}^u)\middle| X_t^u = x\right] \tag{2.22}$$

and problem (2.13) becomes a Dirichlet problem with boundary conditions $V(x, T) = g(x)$ for $x \in \mathcal{D}$ and $V(x, t) = g(x)$ for $(x, t) \in \partial\mathcal{D} \times [0, T]$. Compare also to Section 6.3 for general semi-linear PDEs on bounded domains.

Remark 2.9 (Autonomous version). We can as well consider an autonomous version of the stochastic dynamics (2.1)

$$\mathrm{d}X_s^u = \widetilde{b}(X_s^u, u_s)\,\mathrm{d}s + \widetilde{\sigma}(X_s^u, u_s)\,\mathrm{d}W_s, \qquad X_t^u = x_{\text{init}}, \tag{2.23}$$

on a bounded domain $\mathcal{D} \subset \mathbb{R}^d$, where now $\widetilde{b}$ and $\widetilde{\sigma}$ do not explicitly depend on time anymore. The control then takes the feedback form

$$u_s = u(X_s^u) \tag{2.24}$$

and we consider the cost functional

$$J(u; x) = \mathbb{E}\left[\int_0^{\tau} \widetilde{f}(X_s^u, u_s)\mathrm{d}s + g(X_\tau^u)\middle| X_0^u = x\right], \tag{2.25}$$

which also does not depend on time anymore (note in fact that due to the Markov property the starting time $t = 0$ is arbitrary). $\tau = \inf\{t > 0 : X_t \notin \mathcal{D}\}$ is again the first exit time from $\mathcal{D}$, for which we usually impose conditions guaranteeing $\tau < \infty$ almost surely. Of course (2.25) implies that also the corresponding value function

$$V(x) = \inf_{u\in\mathcal{U}} J(u; x) \tag{2.26}$$

does now not depend on time and a HJB equation analog to (2.13) is

$$\inf_{u\in U}\left\{\widetilde{f}(x, u) + \widetilde{b}(x, u)\cdot\nabla V(x) + \frac{1}{2}(\widetilde{\sigma}\widetilde{\sigma}^\top)(x, u) : \nabla^2 V(x)\right\} = 0, \qquad x \in \mathcal{D}, \tag{2.27a}$$

$$V(x) = g(x), \qquad x \in \partial\mathcal{D}, \tag{2.27b}$$

A proof of a corresponding verification theorem is along the lines of the proof of Theorem 2.4. Further details can for instance be found in [97, Theorem 4.2].

We will end this section on optimal control by noting that for a special (and prominent) choice of running costs $\widetilde{f}$ the minimization appearing in the HJB equation (2.13a) can be solved explicitly, therefore leading to a closed-form PDE. It turns out that this connection is of special relevance for our importance sampling endeavor (cf. Problem 1.4, Remark 1.5, Lemma 2.11, Theorem 2.33 and Remark 2.46).

Corollary 2.10 (HJB equation with quadratic running costs)**.** *If the diffusion coefficient $\widetilde{\sigma}$ does not depend on the control, the control enters additively in the drift, i.e. $\widetilde{b}(x, t, u) = b(x, t) + \sigma u(x, t)$, and the running costs take the form*

$$\widetilde{f}(x, s, u_s) = f(x, s) + \frac{1}{2}|u_s|^2, \tag{2.28}$$

then the HJB PDE (2.13) *can be stated in closed form*

$$(\partial_t + L)V(x, t) - \frac{1}{2}|\sigma^\top\nabla V(x, t)|^2 + f(x, t) = 0, \qquad (x, t) \in \mathbb{R}^d \times [0, T), \tag{2.29a}$$

$$V(x, T) = g(x) \qquad x \in \mathbb{R}^d\,. \tag{2.29b}$$

Proof. We formally compute

$$\inf_{u\in U}\left\{\widetilde{f}(x,t,u)+\widetilde{b}(x,t,u)\cdot\nabla V(x,t)\right\}=f(x,t)+b(x,t)\cdot\nabla V(x,t)+\inf_{u\in U}\left\{\frac{1}{2}|u(x,t)|^2+\sigma u(x,t)\cdot\nabla V(x,t)\right\}, \tag{2.30}$$

and realize that the infimum is attained when choosing $u^*=-\sigma^\top\nabla V$. Plugging this into the HJB PDE (2.19) and recalling the definition of the infinitesimal generator L as provided in (1.19), we readily get the above PDE (2.29). □

It turns out that we can transform the nonlinear HJB PDE from Corollary 2.10 to a linear Feynman-Kac PDE, that we will discuss in Theorem 2.14 in the next section, by using a logarithmic transformation known as Hopf-Cole transformation [93, Section 4.4.1].

Lemma 2.11 (Linearization of HJB equation)**.** *Let $V\in C^{2,1}(\mathbb{R}^d\times[0,T],\mathbb{R})$ solve the HJB PDE* (2.29)*, then $\psi=e^{-V}$ fulfills the linear Feynman-Kac PDE as stated in* (2.44) *(with $k=0$ and appropriately transformed boundary condition), i.e.*

$$(\partial_t+L-f(x,t))\psi(x,t)=0, \qquad (x,t)\in\mathbb{R}^d\times[0,T), \tag{2.31a}$$

$$\psi(x,T)=e^{-g(x)}, \qquad x\in\mathbb{R}^d. \tag{2.31b}$$

Subsequently, we get an explicit representation for the value function by

$$V(x,t)=-\log\mathbb{E}\left[\exp\left(-\int_t^T f(X_s,s)\mathrm{d}s-g(X_T)\right)\middle|X_t=x\right]. \tag{2.32}$$

Proof. See Appendix C.1. □

2.2 Stochastic representations of PDEs

The Hamilton-Jacobi-Bellman equation that we have discussed in the previous section is a PDE that connects stochastic processes to deterministic quantities (in this case the value function) and is therefore a prominent example of the interplay between these two sides of (more or less) the same problem. We have so far taken the route of first stating the stochastic problem and subsequently identifying its deterministic counterpart as a convenient tool for solving it. Likewise, it will turn out to be helpful to take the other direction, i.e. to rely on stochastic representations in order to develop computational methods for solving deterministic PDEs. Our control theory considerations from the previous section highlight one particular aspect of this fruitful connection, which we want to generalize to other types of PDEs in the upcoming sections. We will start with certain linear PDEs in Section 2.2.1, whose solutions can be explicitly expressed as conditional expectations via the Feynman-Kac theorem, for instance following [162, 219, 248]. In Section 2.2.2 we will then move to semi-linear PDEs, whose stochastic counterpart can be described by backward stochastic differential equations (BSDEs). These representations are rather implicit, but will still provide fruitful numerical algorithms for the approximation of PDEs, as will be described in more detail in Chapters 4 and 6 later. In Section 2.2.2 we will first try to give an intuitive introduction and state some general existence and uniqueness results. Since the treatment of HJB PDEs will be of central importance throughout this thesis, we will include an additional well-posedness result for the case where the nonlinear term of the PDE can have a quadratic dependency on the gradient of the solution (and therefore not be Lipschitz continuous). Finally, in Section 2.2.3 we will discuss some basic results for the numerical discretization of both forward and backward processes that will be needed for the implementation of the stochastic representations later on.

2.2.1 Stochastic representations of linear PDEs

Let us first introduce stochastic representations of linear PDEs. We recall that in its general form a linear PDE of k-th order can be written as

$$\sum_{|\alpha|\le k}a_\alpha(x,t)D^\alpha V(x,t)=a_0(x,t), \tag{2.33}$$

where $a_\alpha:\mathbb{R}^d\times[0,T]\to\mathbb{R}$ and $a_0:\mathbb{R}^d\times[0,T]\to\mathbb{R}$ are given functions and where we use the multi-index notation for partial derivatives[13] (see Appendix A). We note that the linear parabolic differential operator

$$\partial_t+L=\partial_t+\frac{1}{2}\sum_{i,j=1}^d(\sigma\sigma^\top)_{ij}(x,t)\partial_{x_i}\partial_{x_j}+\sum_{i=1}^d b_i(x,t)\partial_{x_i}, \tag{2.34}$$

[13] To be precise, we introduce the variable $z=(x,t)^\top\in\mathbb{R}^{d+1}$, for which the multi-index notation gets applied.

that we have already defined in (1.19), is a special case of the one in (2.33). Let us further recall that L is the infinitesimal generator of the stochastic process

$$\mathrm{d}X_s = b(X_s, s)\,\mathrm{d}s + \sigma(X_s, s)\,\mathrm{d}W_s, \qquad X_t = x_{\text{init}}, \tag{2.35}$$

that we have for instance considered in (1.2), with $b : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ and $\sigma : \mathbb{R}^d \times [0,T] \to \mathbb{R}^{d\times d}$ being functions that fulfill Assumption 1 in order to guarantee for strong solutions of the SDE.

The simplest linear PDE that can be connected to stochastic processes such as (2.35) is the Kolmogorov backward equation, which we state in the following theorem.

Theorem 2.12 (Kolmogorov backward equation)**.** *Let X_t be a strong solution of* (2.35)*, let*

$$\psi(x,t) = \mathbb{E}[g(X_T)|X_t = x], \tag{2.36}$$

with[14] *$g \in C_b(\mathbb{R}^d)$ and assume $\psi \in C^{2,1}(\mathbb{R}^d \times [0,T], \mathbb{R})$. Then ψ solves the parabolic terminal value problem*

$$(\partial_t + L)\psi(x,t) = 0, \qquad (x,t) \in \mathbb{R}^d \times [0,T), \tag{2.37a}$$

$$\psi(x,T) = g(x), \qquad x \in \mathbb{R}^d. \tag{2.37b}$$

Proof. See for instance [230, Theorem 2.1] for $d = 1$, where a multidimensional extension is straightforward, or [83, Chapter 9], [281, Proposition 2.6]. □

Remark 2.13. An original intention for studying the above connection was to use tools from the theory of PDEs in order to establish, under suitable conditions, the existence of a solution to the transition probability of a stochastic process and therefore the existence of continuous Markov processes, where we refer to [276, Chapters 2 and 3] for modern approaches relating to this methodology. As will turn out in Chapters 4 and 6, and as already mentioned in the introduction of this section, we will have a quite different goal.

The Kolmogorov backward equation from Theorem 2.12 can be generalized by the Feynman-Kac formula. Before we will state the proper theorem, let us first provide a formal derivation of this relation in order to gain some intuition (ignoring most technical details for a moment). To this end, assume there exists a solution $\psi \in C^{2,1}(\mathbb{R}^d \times [0,T], \mathbb{R})$ to the linear PDE

$$(\partial_t + L - f(x,t))\psi(x,t) = 0, \qquad (x,t) \in \mathbb{R}^d \times [0,T), \tag{2.38a}$$

$$\psi(x,T) = g(x), \qquad x \in \mathbb{R}^d, \tag{2.38b}$$

where $f \in C(\mathbb{R}^d \times [0,T], \mathbb{R})$ is a given function. A concept that we will repeatedly encounter throughout this thesis is to consider the solution ψ along a trajectory of the stochastic process (2.35), namely $Y_t := \psi(X_t, t)$. An application of Itô's formula (as specified in Theorem B.2) then brings

$$Y_t = g(X_T) - \int_t^T f(X_s, s) Y_s\,\mathrm{d}s - \int_t^T \sigma^\top \nabla\psi(X_s, s) \cdot \mathrm{d}W_s \tag{2.39}$$

and taking conditional expectations on both sides yields

$$\psi(x,t) = \mathbb{E}\left[Y_t | X_t = x\right] = \mathbb{E}\left[g(X_T) - \int_t^T f(X_s, s) Y_s \mathrm{d}s \middle| X_t = x \right]. \tag{2.40}$$

We can now identify the ODE

$$\mathrm{d}Y_s = -f(X_s, s) Y_s\,\mathrm{d}s, \qquad Y_T = g(X_T), \tag{2.41}$$

'inside' the expectation operator, which can be solved explicitly by an application of the variation of constants formula[15], yielding

$$\psi(x,t) = \mathbb{E}\left[e^{-\int_t^T f(X_s,s)\mathrm{d}s} g(X_T) \middle| X_t = x \right]. \tag{2.43}$$

[14] Note that this function g should not be confused with the g appearing in the work functional $\mathcal{W}$ defined in (1.8). We accept this slight abuse of notation in order to draw a connection to the later appearing more general semi-linear PDEs. In fact, as explained in Corollary 2.10, g in Theorem 2.12 is connected to g in (1.8) via the transformation $x \mapsto e^{-x}$.

[15] To make the argument more precise, the stochastic version of the variation of constants formula for the SDE (2.39) yields

$$Y_t = e^{-\int_t^T f(X_s,s)\mathrm{d}s} g(X_T) - \int_t^T e^{-\int_t^s f(X_r,r)\mathrm{d}r} (\sigma^\top \nabla\psi)(X_s, s) \cdot \mathrm{d}W_s, \tag{2.42}$$

where the martingale term vanishes when taking conditional expectations, therefore yielding expression (2.43), cf. also [307, Proposition 4.1.2].

We have ignored some technical details here, but in fact the common proof of the following theorem, which will make our considerations more precise and general, essentially takes the same arguments in reversed order.

Theorem 2.14 (Feynman-Kac)**.** *Let $f, k \in C(\mathbb{R}^d \times [0,T], \mathbb{R})$, let $g \in C^2(\mathbb{R}^d, \mathbb{R})$ be bounded from below, let $\psi \in C^{2,1}(\mathbb{R}^d \times [0,T], \mathbb{R})$ have bounded derivatives and let it solve the parabolic terminal value problem*

$$(\partial_t + L - f(x,t))\psi(x,t) + k(x,t) = 0, \qquad (x,t) \in \mathbb{R}^d \times [0,T), \tag{2.44a}$$

$$\psi(x,T) = g(x), \qquad x \in \mathbb{R}^d. \tag{2.44b}$$

Then

$$\psi(x,t) = \mathbb{E}\left[\int_t^T e^{-\int_t^r f(X_s,s)\mathrm{d}s} k(X_r,r)\mathrm{d}r + e^{-\int_t^T f(X_s,s)\mathrm{d}s} g(X_T) \middle| X_t = x\right], \tag{2.45}$$

where X_s is a strong solution to (1.2)*.*

Proof. The proof, whose main ingredient is Itô's Lemma, can for instance be found in [162, Theorem 5.7.6]. □

Remark 2.15. Given suitable regularity assumptions, the converse statement also holds: the stochastic representation of ψ as in (2.45) implies that it solves the PDE (2.44), see for instance [155]. When not having appropriate assumptions on the other hand, one should be careful saying anything about the existence or the regularity of the solution.

Remark 2.16 (Initial value problems). In both the Kolmogorov backward and the Feynman-Kac equation, time can be reversed and initial instead of terminal value problems can be formulated, for instance yielding

$$(\partial_t + L - f(x,t))\psi(x,t) + k(x,t) = 0, \qquad (x,t) \in \mathbb{R}^d \times (0,T], \tag{2.46a}$$

$$\psi(x,0) = g(x), \qquad x \in \mathbb{R}^d, \tag{2.46b}$$

instead of (2.44), now with the stochastic representation

$$\psi(x,t) = \mathbb{E}\left[\int_0^t e^{-\int_0^r f(X_s,s)\mathrm{d}s} k(X_r,r)\mathrm{d}r + e^{-\int_0^t f(X_s,s)\mathrm{d}s} g(X_t) \middle| X_0 = x\right]. \tag{2.47}$$

Remark 2.17 (Bounded domains). In analogy to Remarks 2.8 and 2.9 we can restrict ourselves to bounded domains $\mathcal{D} \subset \mathbb{R}^d$ and consider the parabolic PDE (2.44a) on $\mathcal{D}$, adding the additional boundary condition[16] $\psi(x,t) = g(x)$ for $x \in \partial\mathcal{D}$. The stochastic representation then becomes

$$\psi(x,t) = \mathbb{E}\left[\int_t^{\tau\wedge T} e^{-\int_t^r f(X_s,s)\mathrm{d}s} k(X_r,r)\mathrm{d}r + e^{-\int_t^{\tau\wedge T} f(X_s,s)\mathrm{d}s} g(X_{\tau\wedge T}) \middle| X_t = x\right], \tag{2.48}$$

with $\tau = \inf\{t > 0 : X_t \notin \mathcal{D}\}$, where we usually assume $\tau < \infty$ almost surely. Likewise, we can consider the elliptic boundary value problem

$$(L - f(x,t))\psi(x) + k(x) = 0, \qquad x \in \mathcal{D}, \tag{2.49a}$$

$$\psi(x) = g(x), \qquad x \in \partial\mathcal{D}, \tag{2.49b}$$

where now the solution ψ, the coefficients b and σ in the SDE (2.35) as well as f and k do not depend explicitly on time anymore, yielding the stochastic representation

$$\psi(x) = \mathbb{E}\left[\int_0^\tau e^{-\int_t^r f(X_s)\mathrm{d}s} k(X_r)\mathrm{d}r + e^{-\int_t^\tau f(X_s)\mathrm{d}s} g(X_\tau) \middle| X_0 = x\right], \tag{2.50}$$

again with $\tau = \inf\{t > 0 : X_t \notin \mathcal{D}\}$, see e.g. [162, Proposition 5.7.2].

Remark 2.18 (Semi-group property). Further intuition for the Feynman-Kac formula as stated in Theorem 2.14 can be gained by taking the semi-group perspective. To this end we first define the *transfer operator* $P_t : C(\mathbb{R}^d, \mathbb{R}) \to C(\mathbb{R}^d, \mathbb{R})$ by

$$(P_t g)(x) = \mathbb{E}[g(X_t) | X_0 = x], \tag{2.51}$$

[16] One can of course also consider a boundary condition that is different from the terminal condition (2.44b), see for instance [190, Proposition 6.1] or (6.64).

noting that it is the main ingredient in the definition of the infinitesimal generator L,

$$Lg = \lim_{t\to 0} \frac{P_t g - g}{t}, \tag{2.52}$$

which we have specified already in (1.19), assuming $g \in C^2(\mathbb{R}^d, \mathbb{R})$. By formally solving the Kolmogorov backward equation (2.36), which (considered as an initial value problem) we can write as

$$\partial_t (P_t g)(x) = L(P_t g)(x), \qquad (P_0 g)(x) = g(x), \tag{2.53}$$

we realize that $(P_t)_{t\geq 0}$ defines a semi-group and can be expressed as

$$(P_t g)(x) = (e^{Lt} g)(x). \tag{2.54}$$

Along those lines, now corresponding to the Feynman-Kac formula from Theorem 2.14, we can further define the operator

$$(P_t^f g)(x) = \mathbb{E}\left[g(X_t) e^{-\int_0^t f(X_s)\mathrm{d}s} \middle| X_0 = x \right], \tag{2.55}$$

where now for simplicity f does not explicitly depend on time, bringing the *Feynman-Kac semi-group*

$$(P_t^f g)(x) = (e^{(L-f)t} g)(x). \tag{2.56}$$

In fact, the Feynman-Kac Theorem 2.14 and its relation to the Kolmogorov Theorem 2.12 can be understood via this semi-group formula [228]. To this end, let us consider a time grid $t = t_0 < t_1 < \cdots < t_N = T$ and note that for a small step-size $\Delta t > 0$ the Trotter–Kato formula [121, Theorem 20.1] brings

$$\psi(x,t) = (e^{(L-f)t} g)(x) \approx P_{\Delta t} \circ e^{-\Delta t f(\cdot)} \circ P_{\Delta t} \circ e^{-\Delta t f(\cdot)} \circ \cdots \circ P_{\Delta t}(e^{-\Delta t f(\cdot)} g), \tag{2.57}$$

where the operator $P_{\Delta t} \circ e^{-\Delta t f(\cdot)}$ is applied N times with $P_{\Delta t}$ as defined in (2.51). Note that due to the Markov property we have for any $s \geq 0$ that $(P_{\Delta t} g)(x) = \mathbb{E}\left[g(X_{s+\Delta t}) | X_s = x \right]$. Hence we can write

$$\psi(x,t) \approx \mathbb{E}\left[e^{-\Delta t f(X_{t_1})} \mathbb{E}\left[e^{-\Delta t f(X_{t_2})} \cdots \mathbb{E}\left[e^{-\Delta t f(X_T)} g(X_T) \middle| \mathcal{F}_{t_{N-1}} \right] \cdots \middle| \mathcal{F}_{t_1} \right] \middle| X_t = x \right], \tag{2.58}$$

which due to the tower property

$$\mathbb{E}\left[e^{-\Delta t f(X_{t_i})} \mathbb{E}\left[\xi | \mathcal{F}_{t_i} \right] \middle| \mathcal{F}_{t_{i-1}} \right] = \mathbb{E}\left[e^{-\Delta t f(X_{t_i})} \xi \middle| \mathcal{F}_{t_{i-1}} \right] \tag{2.59}$$

becomes

$$\psi(x,t) \approx \mathbb{E}\left[e^{-\Delta t \sum_{i=1}^N f(X_{t_i})} g(X_T) \middle| X_t = x \right]. \tag{2.60}$$

Taking the limit $\Delta t \to 0$ we can formally recover the (shifted and time-reversed) Feynman-Kac formula (2.55), i.e. $\psi(x,t) = \mathbb{E}\left[e^{-\int_t^T f(X_s)\mathrm{d}s} g(X_T) \middle| X_t = x \right]$. We will come back to this semi-group viewpoint in the next section when considering its nonlinear extension. For further semi-group analysis we recommend [278, Chapter 1] and for a study on its numerical discretization properties we refer to [94].

2.2.2 Stochastic representations of nonlinear PDEs via FBSDEs

We have discussed a stochastic representation of a certain kind of linear PDE via the Feynman-Kac formula stated in Theorem 2.14. In this section, we want to move one step further and approach nonlinear PDEs. To be precise, we will consider PDEs of semi-linear type. In its general form a semi-linear PDE of k-th order can be defined as

$$\sum_{|\alpha|=k} a_\alpha(x,t) D^\alpha V(x,t) + a_0(x,t,V(x,t), DV(x,t), \ldots, D^{k-1}V(x,t)) = 0, \tag{2.61}$$

where $a_\alpha : \mathbb{R}^d \times [0,T] \to \mathbb{R}$ and $a_0 : \mathbb{R}^d \times [0,T] \times \mathbb{R} \times \mathbb{R}^d \times \cdots \times \mathbb{R}^{d^{k-1}} \to \mathbb{R}$ are given functions, recalling the multi-index notation from Appendix A. Exploiting again the correspondence between the differential operator (2.34) and the SDE (2.35), however this time allowing for nonlinear terms, will lead to so-called backward stochastic differential equations (BSDEs) as a stochastic counterpart to certain PDEs of semi-linear form. BSDEs have first been introduced in the 1970s [35] and a systematic study began in the 1990s [229], leading to many technically challenging subtleties. At the same time BSDEs turn out to bring quite versatile tools for computational approaches, which we will use heavily later on. In this section we will introduce basic concepts as well as well-posedness statements based on [109, 228, 234, 281, 307], to which we refer for further technical details. In some of those references statements are posed in a slightly more general version, cf. Remark 2.24.

Let us start by gaining some intuition along the lines of Remark 2.18, where we have discussed the Feynman-Kac semi-group (again following [228]). Instead of finding a stochastic representation of the linear PDE

$$(\partial_t + L)V(x,t) - f(x)V(x,t) = 0, \qquad V(x,T) = g(x), \tag{2.62}$$

we now aim at the nonlinear PDE

$$(\partial_t + L)V(x,t) + h(V(x,t)) = 0, \qquad V(x,T) = g(x), \tag{2.63}$$

where $h : \mathbb{R} \to \mathbb{R}$ is some nonlinear (usually Lipschitz continuous) function. Unfortunately, we cannot write down an explicit expression for the semi-group, as we did in (2.56) for the linear case, anymore. Instead, let us denote by $\Phi_t(V)$ the value at time t of the solution to the ODE

$$\frac{\mathrm{d}X_t}{\mathrm{d}t} = h(X_t), \qquad X_0 = V. \tag{2.64}$$

In analogy to (2.58), we can then formally apply the Trotter-Kato formula

$$\psi(x,t) \approx \mathbb{E}\left[\Phi_{\Delta t} \circ \mathbb{E}\left[\Phi_{\Delta t} \circ \cdots \mathbb{E}\left[\Phi_{\Delta t} \circ g(X_T)\middle|\mathcal{F}_{t_{n-1}}\right] \cdots \middle|\mathcal{F}_{t_1}\right]\middle|X_t = x\right] \tag{2.65}$$

for a sufficiently small time-step Δt. As before we would like to get a limiting formula for $\Delta t \to 0$ such that we obtain an evolution of the process $V(X_t, t)$ that does not rely on the knowledge of the function $V(x,t)$ itself. Inspecting the PDE (2.63) shows that two different actions are present: one is given by the ODE via the function $-h$, the other by a projection on the σ-algebra $\mathcal{F}_t$ associated to the current time t via the infinitesimal generator L. Note that in contrast to the linear case the conditional expectations in (2.65) do not commute with the nonlinear mappings $\Phi_{\Delta t}$. In order to still find a feasible formula for $V(X_t, t)$, let us proceed as we have done in the linear case before, cf. (2.39), and apply Itô's formula to $Y_t = V(X_t, t)$ (assuming that $V \in C^{2,1}(\mathbb{R}^d \times [0,T], \mathbb{R})$ is a classical solution to (2.63)) to get

$$Y_t = g(X_T) + \int_t^T h(Y_s)\mathrm{d}s - \int_t^T \sigma^\top \nabla V(X_s, s) \cdot \mathrm{d}W_s. \tag{2.66}$$

Taking conditional expectations w.r.t. to $\mathcal{F}_t$ on both sides then yields

$$\widetilde{Y}_t := \mathbb{E}\left[Y_t|\mathcal{F}_t\right] = \mathbb{E}\left[g(X_T) + \int_t^T h(Y_s)\mathrm{d}s\middle|\mathcal{F}_t\right], \tag{2.67}$$

which indeed seems to be a good candidate for the desired stochastic representation of V. Following this line, we can in fact find a slightly more general way of expressing Y_t compared to (2.66). To this end, consider the random variable appearing in (2.67),

$$\chi = g(X_T) + \int_t^T h(Y_s)\mathrm{d}s, \tag{2.68}$$

which is a functional of Brownian motion $(W_s)_{t \le s \le T}$ and $\mathcal{F}_T$-measurable. Provided that χ is square-integrable there now exists a unique d-dimensional process $(Z_s)_{0 \le s \le T}$ such that

(i) $\mathbb{E}\left[\int_0^T |Z_s|^2 \mathrm{d}s\right] < \infty$,

(ii) $\chi = \mathbb{E}[\chi|\mathcal{F}_t] + \int_t^T Z_s \cdot \mathrm{d}W_s$.

We can therefore write

$$\overline{Y}_t = g(X_T) + \int_t^T h(Y_s)\mathrm{d}s - \int_t^T Z_s \cdot \mathrm{d}W_s, \tag{2.69}$$

which is a more general version of Y_t as defined in (2.66). We should note that both Y_t and Z_t are $\mathcal{F}_t$-adapted, and that, since the process Y_t has a terminal rather than an initial condition, i.e. $Y_T = V(X_T, T) = g(X_T)$, it is not really natural for Y_t to be adapted to the Brownian motion $(W_s)_{0 \le s \le t}$ before time t. In fact, the process Z_t as a factor in front of the Brownian motion can be interpreted as to satisfy this constraint, stressing that we are really looking for a pair of processes (Y, Z).

Now that we have gained some intuition, let us make our formal considerations more precise. Throughout, we will consider the following type of semi-linear PDEs.

Definition 2.19 (Semi-linear PDE)**.** Let $V \in C^{2,1}(\mathbb{R}^d \times [0,T], \mathbb{R})$, we consider PDEs of the form

$$(\partial_t + L)V(x,t) + h(x,t,V(x,t),\sigma^\top \nabla V(x,t)) = 0, \qquad (x,t) \in \mathbb{R}^d \times [0,T), \tag{2.70a}$$

$$V(x,T) = g(x), \qquad x \in \mathbb{R}^d, \tag{2.70b}$$

where $h \in C(\mathbb{R}^d \times [0,T] \times \mathbb{R} \times \mathbb{R}^d, \mathbb{R})$, $g \in C(\mathbb{R}^d, \mathbb{R})$ and L is a differential operator defined in (1.19).

Further, we define the solution to a forward-backward stochastic differential equation (FBSDE) as a triple (X, Y, Z).

Definition 2.20 (FBSDE)**.** We consider the forward-backward stochastic differential equation

$$\mathrm{d}X_s = b(X_s, s)\,\mathrm{d}s + \sigma(X_s, s)\,\mathrm{d}W_s, \qquad X_t = x_{\text{init}}, \tag{2.71a}$$

$$\mathrm{d}Y_s = -h(X_s, s, Y_s, Z_s)\,\mathrm{d}s + Z_s \cdot \mathrm{d}W_s, \qquad Y_T = g(X_T), \tag{2.71b}$$

where the forward process X_s is as in (1.2), the backward processes Y_s and Z_s are progressively measurable[17] and the functions h and g are as in Definition 2.19.

Let us remind ourselves once more that backward processes have a specified target value, but must still be $\mathcal{F}_s$-adapted for all $s \in [0,T]$. This means that they must not be interpreted as time-reversed processes, since those would depend on $\mathcal{F}_T$ for all times $s \in [0,T]$.

Remark 2.21. (Martingale representation theorem and the role of Z) As already hinted at in the introduction of this subsection, one can interpret BSDEs as a nonlinear version of the martingale representation theorem, noting that the presence of Z is crucial in order to ensure the $(\mathcal{F}_s)_{0\le s\le T}$-measurability of Y. In contrast, consider for instance the BSDE

$$\mathrm{d}Y_s = \sigma(Y_s, s)\,\mathrm{d}W_s, \qquad Y_T = \xi, \tag{2.72}$$

for $\xi \in L^2(\mathcal{F}_T)$. Then typically this equation has no $(\mathcal{F}_s)_{0\le s\le T}$-measurable solution. Consider for instance $\sigma = 0$, then $Y_s = \xi$ for all $s \in [0,T]$, which is not $(\mathcal{F}_s)_{0\le s\le T}$-measurable unless $\xi \in \mathcal{F}_0$. Consider now the martingale $Y_t = \mathbb{E}[\xi|\mathcal{F}_t]$, then by the martingale representation theorem [219, Theorem 4.3.4], there exists a unique $Z \in L^2((\mathcal{F}_s)_{0\le s\le T})$ such that

$$Y_t = \xi - \int_t^T Z_s \cdot \mathrm{d}W_s. \tag{2.73}$$

We note that this setting corresponds to choosing $h = 0$ in (2.71b).

Remark 2.22 (Stochastic version of methods of characteristics). In some sense FBSDEs can be interpreted as a stochastic version of the *method of characteristics* for solving PDEs of first order. The idea of the latter is to approach the linear PDE

$$\partial_t V(x,t) + b(x,t) \cdot \nabla V(x,t) = 0, \qquad (x,t) \in \mathbb{R}^d \times [0,T), \tag{2.74a}$$

$$V(x,T) = g(x), \qquad x \in \mathbb{R}^d, \tag{2.74b}$$

via the process $Y_s = V(X_s, s)$ and the system of ODEs

$$\dot{X}_s = b(X_s, s), \qquad X_0 = x_0, \tag{2.75a}$$

$$\dot{Y}_s = 0, \qquad Y_T = g(X_T). \tag{2.75b}$$

This can be compared to Definitions 2.19 and 2.20 with $h = 0$, where the additional second derivatives in the PDE change the ODE (2.75a) to SDE (2.71a) and the additional Itô term in the Y process guarantees adaptedness. When generalizing (2.74) to nonlinear PDEs

$$\partial_t V(x,t) + b(x,t) \cdot \nabla V(x,t) + h(x,t,V(x,t)) = 0, \qquad (x,t) \in \mathbb{R}^d \times [0,T), \tag{2.76a}$$

$$V(x,T) = g(x), \qquad x \in \mathbb{R}^d, \tag{2.76b}$$

then $Z_s = 0$ in (2.71b) would suffice and $(Y_s)_{0\le s\le T}$ would be the solution to the ODE

$$\dot{Y}_s = -h(X_s, s, Y_s), \qquad Y_T = g(X_T). \tag{2.77}$$

Clearly, the randomness of the general BSDE from Definition 2.20 enters through the process X (and therefore through Brownian motion W) and we note once more that the role of the stochastic integral $\int_0^T Z_s \cdot \mathrm{d}W_s$ is to make the process Y adapted, i.e. to again "remove" its randomness. Along those lines we refer to [307, Section 9.4] for an interesting interpretation of Z essentially being the derivative of Y with respect to W.

[17] Progressive measurability implies $\mathcal{F}_s$-adaptedness, the converse is not necessarily true.

In the following, let us state some key results on BSDEs, focusing on existence and uniqueness results as well as on the precise connection to the nonlinear PDE from Definition 2.19. We start with posing appropriate assumptions.

Assumption 2. *For the coefficients in the FBSDE system from Definition 2.20 we assume that*

(i) $b(\mathbf{0},\cdot), \sigma(\mathbf{0},\cdot), h(\mathbf{0},\cdot,0,\mathbf{0}), g(\mathbf{0})$ *are bounded,*

(ii) b, σ, h, g *are uniformly Lipschitz continuous in* (x, y, z)*,*

(iii) b, σ, h *are uniformly Hölder-*$\frac{1}{2}$ *continuous in time.*

Theorem 2.23 (Existence and uniqueness of FBSDE)**.** *Given Assumption 2, there exists a unique solution to the FBSDE system from Definition 2.20.*

Proof. The proof is essentially based on the martingale representation theorem and a Picard iteration scheme. See for instance [90], [281, Theorem 10.2.] or [307, Theorem 4.3.1]. □

Remark 2.24. The BSDE that we have stated in Definition 2.20 is sometimes termed *Markovian BSDE* and can be seen as a special case of the more general formulation

$$\mathrm{d}Y_s = -h(s,\omega,Y_s,Z_s)\,\mathrm{d}s + Z_s \cdot \mathrm{d}W_s, \qquad Y_T = \xi, \tag{2.78}$$

where ξ is $\mathcal{F}_T$-measurable satisfying $\mathbb{E}[|\xi|^2] < \infty$, $\omega \in \Omega$, and h is $\mathcal{P} \otimes \mathcal{B}^d \otimes \mathcal{B}^{d\times n}$-measurable with $\mathcal{P}$ being the predictable σ-algebra and $\mathcal{B}^d$ the Borel σ-algebra[18], and we have in mind that h can depend on continuous paths generated by Brownian motion. We note that many statements (as for instance the well-posedness in Theorem 2.23) hold for this more general case, see for instance [281, Section 10.5] and [307].

Finally, the following statement makes the connection between the FBSDE from Definition 2.20 and the PDE from Definition 2.19 precise.

Theorem 2.25 (Connection between FBSDE and PDE)**.** *Let* $V \in C^{2,1}(\mathbb{R}^d \times [0,T])$ *be a classical solution to the semi-linear PDE* (2.70) *satisfying a linear growth condition and assume that for some constants* $C, \alpha > 0$ *we have* $|\nabla V(t,x)| \leq C(1+|x|^\alpha)$ *for all* $x \in \mathbb{R}^d$*. Then, the pair* (Y, Z) *defined by*

$$Y_t = V(X_t, t), \qquad Z_t = \sigma^\top \nabla V(X_t, t), \qquad 0 \leq t \leq T \tag{2.79}$$

is the solution to the BSDE (2.71b)*.*

Proof. The statement follows from an application of Itô's formula to Y_t, as done in (2.66), see also [234, Theorem 6.3.2]. □

Remark 2.26 (Viscosity solutions)*.* Note that even though $Y_t = V(X_t, t)$ might not be smooth it can often be shown that, given suitable regularity assumptions on the coefficients, it is still a unique solution to the PDE (2.70) in the viscosity sense (cf. Remark 2.7), therefore showing the converse of Theorem 2.25, see for instance [54, 90, 228] and [307, Section 5.5].

Remark 2.27 (Bounded domains)*.* In analogy to Remark 2.17 we can constrain the semi-linear parabolic PDE (2.70) that we have defined on all $\mathbb{R}^d$ to a bounded domain $\mathcal{D} \subset \mathbb{R}^d$. We can for instance consider the elliptic boundary value

$$LV(x) + h(x, V(x), \sigma^\top \nabla V(x)) = 0, \qquad x \in \mathcal{D}, \tag{2.80a}$$

$$V(x) = g(x), \qquad x \in \partial\mathcal{D}, \tag{2.80b}$$

where now the solution does not explicitly depend on time anymore. The corresponding backward equation is then defined as

$$\mathrm{d}Y_s = -h(X_s, Y_s, Z_s)\,\mathrm{d}s + Z_s \cdot \mathrm{d}W_s, \qquad Y_\tau = g(X_\tau), \tag{2.81}$$

where $\tau = \{t > 0 : X_t \notin \mathcal{D}\}$ is a first exit time from the domain. A theoretical justification can for instance be found in [228, Theorem 4.6] and we note that parabolic problems on bounded domains can be defined analogously (see also Section 6.3).

[18] In other words, the process $(h(t,\omega,y,z))_{0\leq t\leq T}$ is predictable for every fixed $(y,z) \in \mathbb{R}\times\mathbb{R}^d$.

Corollary 2.28. *Let $V \in C^{2,1}(\mathbb{R}^d \times [0,T])$ be a classical solution to the semi-linear PDE (2.70) and assume the same as in Theorem 2.25. Then the backward processes*

$$Y_t^v = V(X_t^v, t), \qquad Z_t^v = \sigma^\top \nabla V(X_t^v, t), \qquad 0 \le t \le T \tag{2.82}$$

fulfill the generalized FBSDE system

$$\mathrm{d}X_s^v = (b(X_s^v, s) + \sigma(X_s^v, s) v(X_s^v, s))\,\mathrm{d}s + \sigma(X_s^v, s)\,\mathrm{d}W_s, \qquad X_t^v = x_{\text{init}}, \tag{2.83a}$$

$$\mathrm{d}Y_s^v = (-h(X_s^v, s, Y_s^v, Z_s^v) + v(X_s^v, s) \cdot Z_s^v)\,\mathrm{d}s + Z_s^v \cdot \mathrm{d}W_s, \qquad Y_T^v = g(X_T^v), \tag{2.83b}$$

for any $v \in \mathcal{U}$.

Proof. As for Theorem 2.25 the proof follows by an application of Itô's formula. □

Remark 2.29 (Fully nonlinear PDEs). One can also consider more general nonlinear PDEs of the form

$$\partial_t V(x,t) + h(x, t, V(x,t), \nabla V(x,t), \nabla^2 V(x,t)) = 0, \qquad (x,t) \in \mathbb{R}^d \times [0,T), \tag{2.84a}$$

$$V(x,T) = g(x), \qquad x \in \mathbb{R}^d, \tag{2.84b}$$

where now $h : \mathbb{R}^d \times [0,T] \times \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}^{d\times d} \to \mathbb{R}$ can also depend on the Hessian of V. With

$$Y_s = V(X_s, s), \qquad Z_s = \nabla V(X_s, s), \qquad \Gamma_s = \nabla^2 V(X_s, s), \tag{2.85}$$

this brings the BSDE representation

$$\mathrm{d}Y_s = \left(b(X_s, s) \cdot Z_s + \frac{1}{2}(\sigma\sigma^\top)(X_s, s) : \Gamma_s - h(X_s, s, Y_s, Z_s, \Gamma_s) \right) \mathrm{d}s + \sigma^\top(X_s, s) Z_s \cdot \mathrm{d}W_s, \qquad Y_T = g(X_T). \tag{2.86}$$

See [13, 235] for further details and algorithmic approaches.

In this thesis we are particularly interested in Hamilton-Jacobi-Bellman PDEs, as for instance defined in (2.13) or (2.29). Here usually the nonlinearity of the PDE depends quadratically on the gradient of the solution and is therefore not uniformly Lipschitz continuous such that the usual assumptions on the coefficients, as posed in Assumption 2, do not hold anymore. We must therefore come up with additional assumptions and will refer to analysis that can still prove well-posedness of corresponding BSDEs.

Assumption 3 (Quadratic growth in z)**.** *We assume the following.*

(i) There exists $C > 0$ such that for any $(x,t,y,z) \in \mathbb{R}^d \times [0,T] \times \mathbb{R} \times \mathbb{R}^d$

$$|h(x,t,y,z)| \le C(1 + |y| + |z|^2). \tag{2.87}$$

(ii) There exists $C > 0$ such that for any $(x,t,y_1,y_2,z_1,z_1) \in \mathbb{R}^d \times [0,T] \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}^d$

$$|h(x,t,y_1,z_1) - h(x,t,y_2,z_2)| \le C\left(|y_1 - y_2| + (1 + |y_1| + |y_2| + |z_1| + |z_2|)|z_1 - z_2|\right). \tag{2.88}$$

(iii) $g(X_T)$ is $\mathcal{F}_T$-measurable and $\mathbb{P}(|g(X_T)| > C) = 0$ for some $C > 0$.

Theorem 2.30 (Existence and uniqueness of FBSDE with quadratic growth)**.** *Let Assumption 3 hold. Then the FBSDE system as defined in Definition 2.20 admits a unique solution (Y, Z).*

Proof. A proof can be found in [307, Theorem 7.3.3], which is mainly based on [174]. □

Lastly, let us note that there are other interesting connections between BSDEs and stochastic optimal control problems as for instances explained in [234, Section 6.4] and [281, Section 10.4].

2.2.3 Discretization of forward and backward SDEs

For numerical simulations we need to discretize the forward and backward processes from Definition 2.20. In this section we want to state some basic results on the discretization error that one encounters.

Throughout this work we discretize the forward and backward SDEs with the Euler-Maruyama scheme on a time grid $0 = t_0 < t_1 < \cdots < t_N = T$. For the forward process we define the iterative scheme

$$\widehat{X}_{n+1} = \widehat{X}_n + b(\widehat{X}_n, t_n)\Delta t + \sigma(\widehat{X}_n, t_n)\xi_{n+1}\sqrt{\Delta t}, \qquad \widehat{X}_0 = x. \tag{2.89}$$

where usually the step-size $t_{n+1} - t_n = \Delta t = \frac{T}{N}$ is fixed, and $\xi_{n+1} \sim \mathcal{N}(0, \mathrm{Id}_{d\times d})$.

One is usually interested in how $\widehat{X}_n$ converges to X_{t_n} depending on the step-size Δt. The convergence results of scheme (2.89) are classic and the following theorem states that it is strongly converging with order $\frac{1}{2}$ or 1 depending on whether the diffusion coefficient is x-dependent or not.

Theorem 2.31 (Strong convergence of forward SDE scheme)**.** *Let X be a strong solution to* (1.2)*, $\widehat{X}$ the discretization defined by* (2.89) *and let $C(T)$ be a (time-dependent) constant. It holds*

$$\max_{0\leq n\leq N} \mathbb{E}\left[\left|\widehat{X}_n - X_{t_n}\right|\right] \leq C(T)\sqrt{\Delta t}. \tag{2.90}$$

If the diffusion coefficient σ does not depend on x it holds

$$\max_{0\leq n\leq N} \mathbb{E}\left[\left|\widehat{X}_n - X_{t_n}\right|\right] \leq C(T)\Delta t. \tag{2.91}$$

Proof. See [173]. □

The discretization of the backward process (2.71b) is not so obvious. First note that we can write it in its integrated form for the times $t_n < t_{n+1}$ as

$$Y_{t_{n+1}} = Y_{t_n} - \int_{t_n}^{t_{n+1}} h(X_s, s, Y_s, Z_s)\mathrm{d}s + \int_{t_n}^{t_{n+1}} Z_s \cdot \mathrm{d}W_s. \tag{2.92}$$

In a discrete version we have to replace the integrals with suitable discretizations, where for the deterministic integral we can decide which end point to consider, leading to either of the following two discretization schemes

$$\widehat{Y}_{n+1} = \widehat{Y}_n - h(\widehat{X}_n, t_n, \widehat{Y}_n, \widehat{Z}_n)\Delta t + \widehat{Z}_n \cdot \xi_{n+1}\sqrt{\Delta t}, \tag{2.93a}$$

$$\widehat{Y}_{n+1} = \widehat{Y}_n - h(\widehat{X}_{n+1}, t_{n+1}, \widehat{Y}_{n+1}, \widehat{Z}_{n+1})\Delta t + \widehat{Z}_n \cdot \xi_{n+1}\sqrt{\Delta t}. \tag{2.93b}$$

Of course we could also come up with a scheme mixing the end points in the evaluation of the integral over h. Note that the above scheme relies on an explicit representation of Z in terms of Y, which is often given via the relations $\widehat{Y}_n \approx V(\widehat{X}_n, t_n), \widehat{Z}_n \approx \sigma^\top \nabla V(\widehat{X}_n, t_n)$. Alternatively, one can also define the scheme

$$\widehat{Y}_N = g(\widehat{X}_N), \qquad \widehat{Z}_n = \frac{1}{\sqrt{\Delta t}}\mathbb{E}\left[\widehat{Y}_{n+1}\xi_{n+1}\middle|\widehat{X}_n\right], \qquad \widehat{Y}_n = \mathbb{E}\left[\widehat{Y}_{n+1} + h(\widehat{X}_n, t_n, \widehat{Y}_{n+1}, \widehat{Z}_n)\Delta t\middle|\widehat{X}_n\right]. \tag{2.94}$$

Due to reasons that will become more obvious in Section 6.2 this scheme has been prominent in the analysis of the discretization error and the following theorem states its convergence.

Theorem 2.32 (Discretization of BSDE)**.** *Let (Y, Z) evolve according to the BSDE* (2.71b) *and consider the discretization scheme as in* (2.94)*. If Δt is small enough then*

$$\max_{0\leq n\leq N} \mathbb{E}\left[\left(Y_{t_n} - \widehat{Y}_n\right)^2\right] + \sum_{n=0}^{N-1} \mathbb{E}\left[\int_{t_n}^{t_{n+1}} |Z_s - \widehat{Z}_n|^2 \mathrm{d}s\right] \leq C(T)(1+|x|^2)\Delta t. \tag{2.95}$$

Proof. See [307, Theorem 5.3.3]. □

2.3 Importance sampling and large deviations

In this section we come back to Monte Carlo estimation of observables related to (high-dimensional) diffusion processes as one of the original motivations of this thesis (see Chapter 1). We have argued before that corresponding estimators often suffer from high variances and we therefore introduce importance sampling as a popular method to approach this problem by sampling from an alternative probability measure and reweighting the resulting random variables with a likelihood ratio in order to still produce unbiased estimators for the quantity of interest. Loosely speaking, one can understand this change of the underlying probability measure as making regions that are somehow "more important" appear more often in the simulation. In Section 2.3.1 we define importance sampling in an abstract setting, from which an application to densities is straightforward. We will elevate those considerations to measures on path space and diffusion processes in Section 2.3.2, while elaborating on the goal of reaching zero-variance estimators. In practice such optimal sampling schemes are usually not available and we already refer to Chapter 3, where we will analyze statistical errors when considering suboptimal choices. Since high variances of estimators are particularly common when dealing with so-called rare events, we consider the theory of large deviations as an appropriate framework for studying those phenomena in the context of converging sequences of probability measures. In Section 2.3.3 we will give an introduction to this topic, while especially highlighting the perspective of changes of measures and pointing out connections to importance sampling that will help us to understand potential drawbacks of certain practical methods. Finally, in Section 2.3.4 we will introduce a variational formula that connects our sampling problem to changes of measures from yet another perspective that shall turn out to be fruitful in the identification of optimal importance sampling schemes.

2.3.1 Importance sampling

We shall now first introduce the idea of importance sampling in an abstract setting. To this end, we consider the probability space $(\Omega, \mathcal{F}, \Lambda)$ and the measurable space $(\widetilde{\Omega}, \widetilde{\mathcal{F}})$, on which we want to compute expected values

$$\mathcal{Z} = \mathbb{E}\left[e^{-\mathcal{W}(X)}\right], \tag{2.96}$$

where $X : \Omega \to \widetilde{\Omega}$ is a random variable that is distributed according to the measure $\nu = \Lambda(X^{-1}(\cdot))$, and $\mathcal{W}\colon \widetilde{\Omega} \to \mathbb{R}$ is some functional of X. Usually, we will specify $\widetilde{\Omega}$ to be either $\mathbb{R}^d$ or the path space $C([0,T], \mathbb{R}^d)$.

The idea of importance sampling is to sample instead $\widetilde{X} \in \widetilde{\Omega}$ from another distribution $\widetilde{\nu}$ and weight the samples back according to the corresponding likelihood ratio (or Radon-Nikodym derivative), provided that $\nu \ll \widetilde{\nu}$, namely

$$\mathcal{Z} = \mathbb{E}\left[e^{-\mathcal{W}(\widetilde{X})}\frac{\mathrm{d}\nu}{\mathrm{d}\widetilde{\nu}}(\widetilde{X})\right]. \tag{2.97}$$

One notorious intention of importance sampling is the reduction of the variance of the corresponding Monte Carlo estimator

$$\widehat{\mathcal{Z}}^K = \frac{1}{K}\sum_{k=1}^{K} e^{-\mathcal{W}(\widetilde{X}^{(k)})}\frac{\mathrm{d}\nu}{\mathrm{d}\widetilde{\nu}}(\widetilde{X}^{(k)}), \tag{2.98}$$

where K is the sample size and $\widetilde{X}^{(k)}$ are i.i.d. samples from $\widetilde{\nu}$. We therefore often study the relative error

$$r(\widetilde{\nu}) = \frac{\sqrt{\mathrm{Var}\left(e^{-\mathcal{W}(\widetilde{X})}\frac{\mathrm{d}\nu}{\mathrm{d}\widetilde{\nu}}(\widetilde{X})\right)}}{\mathcal{Z}}, \tag{2.99}$$

noting that the true relative error of the estimator (2.98) is given by $r(\widetilde{\nu})/\sqrt{K}$. It can be readily seen that choosing the optimal proposal measure $\widetilde{\nu} = \nu^*$ defined via

$$\frac{\mathrm{d}\nu^*}{\mathrm{d}\nu} = \frac{e^{-\mathcal{W}}}{\mathcal{Z}} \tag{2.100}$$

yields an unbiased zero-variance estimator. Of course, this estimator is usually infeasible in practice, as $\mathcal{Z}$ is just the quantity we are after, and therefore not available.

Note that the exponential form in (2.96), $e^{-\mathcal{W}}$, constrains our observable to be positive. We make this choice in order to be able to reach a zero variance proposal density without additional tricks, as the optimal proposal measure ν^* defined in (2.100) has to be non-negative. Requiring even strict positivity is convenient in order to get variational dualities that rely on logarithmic transformations, cf. Section 2.3.4. An extension of importance sampling to observables with negative parts can for instance be found in [223].

Many common applications of importance sampling consider measures that admit densities in $\mathbb{R}^d$. However, we shall focus on importance sampling of diffusions, which we will introduce in the next section.

2.3.2 Importance sampling in path space

Let us elevate the abstract importance sampling considerations from the previous subsection to solutions of stochastic differential equations (SDEs) of the form

$$\mathrm{d}X_s = b(X_s, s)\,\mathrm{d}s + \sigma(X_s, s)\,\mathrm{d}W_s, \qquad X_t = x_{\mathrm{init}}, \tag{2.101}$$

on the time interval $s \in [t,T]$, $0 \le t < T < \infty$, just as in (1.2). Again, $b : \mathbb{R}^d \times [t,T] \to \mathbb{R}^d$ denotes the drift coefficient, $\sigma : \mathbb{R}^d \times [t,T] \to \mathbb{R}^{d\times d}$ the diffusion coefficient, $(W_s)_{t\le s\le T}$ standard d-dimensional Brownian motion, and $x_{\mathrm{init}} \in \mathbb{R}^d$ is the (deterministic) initial condition. As in (1.7) our goal is to compute expectations of the form

$$\mathcal{Z} = \mathbb{E}\left[e^{-\mathcal{W}(X)}\right], \qquad \mathcal{W}(X) = \int_t^T f(X_s, s)\mathrm{d}s + g(X_T), \tag{2.102}$$

where $f : \mathbb{R}^d \times [t,T] \to \mathbb{R}, g : \mathbb{R}^d \to \mathbb{R}$ are given functions. We will usually fix the initial time to be $t = 0$, i.e. consider the SDE (2.101) on the interval $[0,T]$. For fixed initial condition $x_{\mathrm{init}} \in \mathbb{R}^d$, let us recall the path space

$$\mathcal{C} = C_{x_{\mathrm{init}}}([0,T], \mathbb{R}^d) = \left\{X : [0,T] \to \mathbb{R}^d \mid X \text{ continuous},\ X_0 = x_{\mathrm{init}}\right\}, \tag{2.103}$$

equipped with the supremum norm and the corresponding Borel-σ-algebra, and denote the set of probability measures on $\mathcal{C}$ by $\mathcal{P}(\mathcal{C})$.

As in the previous section, the idea of importance sampling is to not sample from the original path measure $\mathbb{P} \in \mathcal{P}(\mathcal{C})$ that corresponds to paths of SDE (2.101), but from a different measure $\mathbb{P}^u \in \mathcal{P}(\mathcal{C})$ and weight back accordingly. Just as in (2.97) one then gets an unbiased estimator via

$$\mathcal{Z} = \mathbb{E}\left[e^{-\mathcal{W}(X^u)} \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^u}(X^u)\right], \tag{2.104}$$

where the Radon-Nikodym derivative is now given by Girsanov's theorem (as stated in Theorem B.3) and it turns out that the SDE corresponding to $\mathbb{P}^u$ is just a controlled version of the original one,

$$\mathrm{d}X_s^u = (b(X_s^u, s) + \sigma(X_s^u, s)u(X_s^u, s))\,\mathrm{d}s + \sigma(X_s^u, s)\,\mathrm{d}W_s, \qquad X_t^u = x_{\text{init}}. \tag{2.105}$$

We think of $u : \mathbb{R}^d \times [t, T] \to \mathbb{R}^d$ as a control term steering the dynamics and note (as already hinted at by the notation) the correspondence between u and $\mathbb{P}^u$. As before, our quantity of interest is the relative error, which now depends on the control u:

$$r(u) = \frac{\sqrt{\mathrm{Var}\left(e^{-\mathcal{W}(X^u)} \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^u}(X^u)\right)}}{\mathcal{Z}}. \tag{2.106}$$

Relating to Problem 1.1, we recall that it is a common goal to choose a control that minimizes this relative error. In fact, there exists $u^* \in \mathcal{U}$ that brings the variance, and therefore (2.106), the relative error of the importance sampling estimator, to zero. This is formalized in the following theorem.

Theorem 2.33 (Zero-variance property in path space)**.** *Let*

$$\psi(x, t) = \mathbb{E}\left[e^{-\mathcal{W}(X)} \middle| X_t = x\right]. \tag{2.107}$$

Then the path space measure $\mathbb{P}^{u^*}$ induced by the feedback control

$$u^*(x, s) = \sigma^\top \nabla \log \psi(x, s) \tag{2.108}$$

yields a zero variance estimator, i.e.

$$e^{-\mathcal{W}(X^{u^*})} \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^{u^*}}(X^{u^*}) = \psi(x, 0), \quad \mathbb{P}^{u^*} - a.s. \tag{2.109}$$

Proof. See Appendix C.1. □

Remark 2.34. The above zero-variance statement can also be understood by the following intuitive argument. We know by Jensen's inequality that

$$-\log \mathbb{E}\left[e^{-\mathcal{W}(X)}\right] = -\log \mathbb{E}\left[e^{-\mathcal{W}(X^u)} \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^u}(X^u)\right] \leq \mathbb{E}\left[\mathcal{W}(X^u) - \log \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^u}(X^u)\right] = \mathbb{E}\left[\mathcal{W}(X^u)\right] + \mathrm{KL}(\mathbb{P}^u|\mathbb{P}). \tag{2.110}$$

One can show that equality can be reached via a minimization of the right hand side over u (see for instance Theorem 2.44 or Remark 2.46), namely

$$-\log \mathbb{E}\left[e^{-\mathcal{W}(X)}\right] = \inf_{u \in \mathcal{U}} \left\{\mathbb{E}\left[\mathcal{W}(X^u)\right] + \mathrm{KL}(\mathbb{P}^u|\mathbb{P})\right\}, \tag{2.111}$$

where $\mathcal{U}$ is a suitable set of control functions as for instance defined in (1.5). Now, since the negative logarithm is strictly convex, $-\log(\mathbb{E}[Z]) = \mathbb{E}[-\log(Z)]$ only holds if Z is almost surely constant, which readily yields the variance zero property as stated in (2.109) for $u = u^*$ being the minimizer in (2.111) (cf. [98, Section VI.2]).

As we have discussed in Chapter 1 and particularly in Theorem 1.2, it turns out that there are multiple equivalent perspectives on the problem of finding the optimal importance sampling control u^* in practice. Let us relate to the perspective of conditioning from Problem 1.2 again, which claims that $\mathcal{W}$ induces a *reweighted* path measure $\mathbb{Q}$ on $\mathcal{C}$ via

$$\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}} = \frac{e^{-\mathcal{W}}}{\mathcal{Z}}, \tag{2.112}$$

assuming f and g are such that $\mathcal{Z}$ is finite (which we shall tacitly assume). It turns out that $\mathbb{Q} = \mathbb{P}^{u^*}$ and we realize that the above formula is the same as in (2.100).

Remark 2.35 (Path space importance sampling of trajectory-independent observables). Let us consider the special case of $f = 0$ in the path functional (2.102). Then the optimal change of measure (2.112) is given by

$$\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}(X) = \frac{e^{-g(X_T)}}{\mathbb{E}\left[e^{-g(X_T)}\right]}, \tag{2.113}$$

and we see by inspecting the right-hand side that it does not depend on the entire trajectory, but only on the value at final time T. Since X is a Markov process, it is therefore enough to consider reweightings of the probability density of the process at time T, which we call p_T, namely

$$\frac{q_T}{p_T}(x) = \frac{e^{-g(x)}}{\mathbb{E}\left[e^{-g(x)}\right]}, \tag{2.114}$$

where the expectation is now taken with respect to the density p_T instead of the path measure $\mathbb{P}$. Since usually neither p_T nor the expectation value is known in practice, it is equally hard to solve this problem. There exist multiple alternative ideas on somehow 'transforming' p_T to q_T, e.g. by resampling [77] or 'bridging' [101, 137], however, all methods are especially challenging when the two measures are far apart from each other. For further algorithmic approaches of optimal importance sampling with densities we refer to Chapter 7 and Appendix B.11.

2.3.3 Large deviations theory and its connections to importance sampling

A field that is strongly connected to key ideas in importance sampling is large deviations theory. In a nutshell, this often very technical branch of probability theory provides an appropriate framework for the characterization of the asymptotic concentration of probability measures, in particular empirical measures. As the naming suggests, one is not only interested in "typical" behaviors, but rather in largely deviating phenomena that can be very unlikely to happen. Large deviations theory is an indispensable tool in the analysis of rare events (cf. Section 1.3) and often rather theoretic considerations can lead to practical algorithms for numerical simulations.

In this section we want to introduce some fundamental concepts of large deviations theory, e.g. following [43, 68, 69], having the goal in mind to connect them to importance sampling and other aspects that are related to our guiding problems from Chapter 1. For ease of notation, let us start with one-dimensional random variables $X \in \mathbb{R}$ that are distributed according to some measure ν, noting that multidimensional extensions are straightforward. We want to estimate the expectation

$$\mathcal{Z} = \mathbb{E}[X] \tag{2.115}$$

by its Monte Carlo approximation

$$\widehat{\mathcal{Z}}^K = \frac{1}{K}\sum_{k=1}^{K} X^{(k)}, \tag{2.116}$$

where X_k are i.i.d. from ν. We have already discussed why this can be difficult (cf. Example 1.1), but let us provide yet another illustration for potential sampling issues in the following example.

Example 2.36 (Rare event simulations)**.** *Let us consider the random variable $X = \mathbb{1}_{\{Y>c\}}$, where $Y \in \mathbb{R}$ is some other random variable and $c \in \mathbb{R}$ some constant. The computation of the expectation* (2.115) *then results in computing the probability*

$$p = \mathbb{P}(Y > c) = \mathbb{E}\left[\mathbb{1}_{\{Y>c\}}\right]. \tag{2.117}$$

In this scenario, the Monte Carlo approximation (2.116) *gets particularly challenging if p is small (i.e. if the event $Y > c$ is rare), since the relative error of $\widehat{p}^K := \widehat{\mathcal{Z}}^K$ explodes when p approaches zero, namely*

$$r(\widehat{p}_K) = \frac{\sqrt{p(1-p)}}{\sqrt{K}p} = \frac{\sqrt{\frac{1}{p}-1}}{\sqrt{K}} \xrightarrow{p\to 0} \infty \tag{2.118}$$

for a fixed K. This is a stereotypical problem in rare event estimation.

The above example motivates the question of how to choose K in order to still guarantee good estimators (for fixed $p \neq 0$), or differently put, we may ask the question of how the convergence rate of the sample mean to its expectation depends on the sample size K. Large deviations theory tries to address this question, at least in some specific regime. Before getting there, let us recall two standard approaches for studying the convergence

of the Monte Carlo estimator (2.116) to its mean (2.115). First, assuming that $\mathbb{E}[X] < \infty$, the *strong law of large numbers* assures that

$$\widehat{\mathcal{Z}}^K \xrightarrow{\text{a.s.}} \mathcal{Z} \tag{2.119}$$

for $K \to \infty$, but neither says something about convergence speed nor about any kind of fluctuations of the estimator. A second attempt is the *central limit theorem*, which, now assuming that also $\mathbb{E}\left[X^2\right] < \infty$, states that

$$\sqrt{K}\frac{(\widehat{\mathcal{Z}}^K - \mathcal{Z})}{\sigma} \xrightarrow{\text{d}} \mathcal{N}(0,1) \tag{2.120}$$

as $K \to \infty$, where $\sigma^2 = \text{Var}(X)$, or equivalently,

$$\mathbb{P}\left(\sqrt{K}\frac{(\widehat{\mathcal{Z}}^K - \mathcal{Z})}{\sigma} \geq z\right) \to 1 - \phi(z), \tag{2.121}$$

where ϕ is the cumulant distribution function of a standard normally distributed random variable. With the central limit theorem we can make statements about fluctuations around the mean, but should note that loosely speaking due to the "strechting" with σ only "small fluctuations", i.e. fluctuations close to the mean, can be considered. The aim of large deviations theory on the other hand is to go one step further and study fluctuations that are "far away" from the mean, while at the same time providing some asymptotic convergence speed depending on K. To this end, in analogy to the expression in (2.121), let us study the cumulative distribution function of our estimator $\widehat{\mathcal{Z}}^K$, for which we can readily find an upper bound as

$$\mathbb{P}\left(\widehat{\mathcal{Z}}^K \geq x\right) = \mathbb{E}\left[\mathbb{1}_{\widehat{\mathcal{Z}}^K \geq x}\right] = \mathbb{E}\left[\mathbb{1}_{\alpha K(\widehat{\mathcal{Z}}^K - x) \geq 0}\right] \tag{2.122a}$$

$$\leq \mathbb{E}\left[e^{\alpha K(\widehat{\mathcal{Z}}^K - x)}\right] = \mathbb{E}\left[\prod_{k=1}^{K} e^{\alpha X_k}\right] e^{-\alpha K x} \tag{2.122b}$$

$$= \mathbb{E}\left[e^{\alpha X}\right]^K e^{-\alpha K x} = e^{K(C(\alpha) - \alpha x)}, \tag{2.122c}$$

where we have introduced some $\alpha > 0$ and the cumulant generating function

$$C(\alpha) = \log \mathbb{E}\left[e^{\alpha X}\right]. \tag{2.123}$$

Since α is arbitrary we can minimize w.r.t. this quantity in order to make the bound tighter and get

$$\mathbb{P}\left(\widehat{\mathcal{Z}}^K \geq x\right) \leq e^{\inf_{\alpha > 0}\{K(C(\alpha) - \alpha x)\}} = e^{-KC^*(x)}, \tag{2.124}$$

where

$$C^*(x) = \sup_{\alpha \in \mathbb{R}} \{\alpha x - C(\alpha)\} \in [0, \infty) \tag{2.125}$$

is the Legendre-Fenchel transform of C, which has the property of being convex, non-decreasing on $[\mathcal{Z}, \infty)$ and $C^*(\mathcal{Z}) = 0$. Here we already dropped the requirement $\alpha > 0$ as it turns out that it is in fact not needed.

With (2.124) we have identified an upper bound for the cumulative distribution function that decreases exponentially in K, with C^* specifying the convergence speed. Since we aim at convergence statements similar to (2.121), a corresponding lower bound is needed as well. And indeed, without going into further details about its derivation, the following theorem shows that C^* is just right for quantifying the asymptotic behavior of the Monte Carlo estimator (2.116).

Theorem 2.37 (Cramér)**.** *Let $\widehat{\mathcal{Z}}^K$ be as defined in* (2.116)*. Then for any $x > \mathcal{Z}$ it holds*

$$\lim_{K \to \infty} \frac{1}{K} \log \mathbb{P}\left(\widehat{\mathcal{Z}}^K \geq x\right) = -C^*(x) = -\inf_{y \geq x} C^*(y). \tag{2.126}$$

Proof. See [43, Theorem 3.1.1] and note that the last equality in (2.126) follows from the fact that C^* is non-decreasing on $[\mathcal{Z}, \infty)$ (and will make more sense when we introduce the more general notation in Definition 2.39). □

Remark 2.38 (Convergence on exponential scale)*.* In a more compact way we can write Cramér's theorem as

$$\mathbb{P}\left(\widehat{\mathcal{Z}}^K \geq x\right) \simeq e^{-KC^*(x)} \tag{2.127}$$

for any $x > \mathcal{Z}$. It is important to note that equations (2.126) and (2.127) refer to a convergence on an exponential scale and that subexponential behavior might be hidden. We actually have

$$\mathbb{P}\left(\widehat{\mathcal{Z}}^K \geq x\right) = c_K(x)e^{-KC^*(x)}, \tag{2.128}$$

where the sequence $(c_K)_K$ converges to zero at some subexponential rate, i.e. $\frac{\log c_K}{K} \to 0$ for $K \to \infty$. The term c_K, however, is usually very difficult to compute in practice.

The above observations can be put in a more abstract framework and motivate what is called a *large deviation principle*.

Definition 2.39 (Large deviation principle)**.** Let ν_K be a sequence of probability measures on some space $\mathcal{X}$, let a_K be a sequence of positive real numbers such that $a_K \to \infty$ and let $I : \mathcal{X} \to [0, \infty]$ be a lower semicontinuous functional. The sequence ν_K is said to satisfy a *large deviation principle* with *speed* a_K and *rate* I if and only if for each Borel measurable set $E \subset \mathcal{X}$ it holds

$$-\inf_{x\in E^\circ} I(x) \leq \liminf_{K\to\infty} \frac{1}{a_K}\log(\nu_K(E)) \leq \limsup_{K\to\infty} \frac{1}{a_K}\log(\nu_K(E)) \leq -\inf_{x\in\overline{E}} I(x). \tag{2.129}$$

Comparing to our derivation from above we see that we can identify $\nu_K = \mathbb{P}\left(\widehat{\mathcal{Z}}^K \in \cdot\right), E = [x, \infty), a_K = K$ and $I = C^*$, noting once more that the rate function I is the essential quantity that determines the exponential rate of the convergence that we are interested in.

Inspecting the proof of Cramér's theorem, in particular for the lower bound, reveals a technique that we have seen in our importance sampling considerations from Section 2.3.1 already. The idea is to define a new probability measure by an exponential tilting of the original measure ν, namely by

$$\nu_\alpha := e^{\alpha x - C(\alpha)}\nu = \frac{e^{\alpha x}}{\mathbb{E}\left[e^{\alpha X}\right]}\nu, \tag{2.130}$$

where C is the cumulant generating function as defined in (2.123) and $\alpha \in \mathbb{R}$ is the tilting parameter. We note that ν_α and C can be related via

$$C'(\alpha) = \frac{\mathbb{E}\left[Xe^{\alpha X}\right]}{\mathbb{E}\left[e^{\alpha X}\right]} = \mathbb{E}\left[Xe^{\alpha X - C(\alpha)}\right] = \mathbb{E}_{\nu_\alpha}\left[X\right] \tag{2.131}$$

and similarly

$$C''(\alpha) = \mathrm{Var}_{\nu_\alpha}(X), \tag{2.132}$$

so one can say that C encodes some information of ν_α. The intention of the tilting in (2.130) is to shift the probability mass of the original random variables in such a way that the set under consideration becomes likely. This can be seen by noting that with the choice $\alpha = \alpha^*$ that solves the maximization in the Legendre-Fenchel transformation (2.125) we have

$$C'(\alpha^*) = \mathbb{E}_{\nu_{\alpha^*}}\left[X\right] = \mathbb{E}_{\nu_{\alpha^*}}[\widehat{\mathcal{Z}}] = x, \tag{2.133}$$

i.e. $\widehat{\mathcal{Z}} \approx x$ is no longer rare under ν_{α^*} and we conclude that the change of measure in (2.130) is optimal in some asymptotic sense, as will be explained next.

Remark 2.40 (Rare events and asymptotically optimal variance). In order to illustrate further connections to importance sampling, let us come back to the computations in (2.122), also connecting to Example 2.36, and say we want to estimate

$$p_K(x) = \mathbb{P}\left(\widehat{\mathcal{Z}}^K \geq x\right) \tag{2.134}$$

with $\widehat{\mathcal{Z}}^K$ as in (2.116). In analogy to (2.130) we consider the exponential change of measure

$$\nu_\alpha = e^{\alpha K\widehat{\mathcal{Z}}^K - KC(\alpha)}\nu, \tag{2.135}$$

such that we have

$$p_K(x) = \mathbb{E}_{\nu_\alpha}\left[\mathbb{1}_{\widehat{\mathcal{Z}}^K \geq x}\frac{\mathrm{d}\nu}{\mathrm{d}\nu_\alpha}\right]. \tag{2.136}$$

We can now compute the second moment depending on the tilting parameter α as

$$M_K(x,\alpha) = E_{\nu_\alpha}\left[\mathbb{1}_{\widehat{\mathcal{Z}}^K \geq x}e^{-2K(\alpha\widehat{\mathcal{Z}}^K - C(\alpha))}\right] \tag{2.137a}$$

$$\leq e^{-2K(\alpha x - C(\alpha))}, \tag{2.137b}$$

where we have used the same bound as in (2.122), noting that minimization w.r.t. α brings

$$M_K(x, \alpha^*) \leq e^{-2KC^*(x)}. \tag{2.138}$$

Combining this with our large deviations result as for instance stated in (2.127) and applying Jensen's inequality we get

$$e^{-2KC^*(x)} \simeq p_K^2(x) \leq M_K(x, \alpha^*) \leq e^{-2KC^*(x)}, \tag{2.139}$$

implying

$$\lim_{K\to\infty} \frac{1}{K} \log p_K^2(x) = \lim_{K\to\infty} \frac{1}{K} \log M_K(x, \alpha^*) = -2C^*(x), \tag{2.140}$$

or equivalently

$$p_K^2 \simeq M_K(x, \alpha^*), \tag{2.141}$$

which means that up to subexponential terms the tilting (2.135) is asymptotically optimal in terms of variance reduction. In fact, a guiding principle in the efficient estimation of rare event probabilities by Monte Carlo is that importance sampling based on the change of measure suggested by large deviations theory can reduce the variance by many orders of magnitude. We will later see, however, that importance sampling in a nonasymptotic regime can still lead to large relative errors, since here the subexponential terms do matter (see Remark 2.42, Chapter 3, in particular Section 3.2.2, and e.g. [108]).

For non-independent random variables Cramér's Theorem 2.37 can be generalized to the Gärtner-Ellis theorem. Let us here provide a glimpse on how this leads to large deviations theory of time-continuous stochastic processes, for which we refer to [43, 68, 69] for further details. Instead of having K different random variables, where we are interested in the behavior of sample means when $K \to \infty$, we can now study two kinds of limits: we can either let the time horizon of the stochastic process go to infinity or we can let its noise of go to zero. Let us start with the latter case. To this end, we consider the SDE

$$\mathrm{d}X_s^\eta = b(X_s^\eta)\,\mathrm{d}s + \sqrt{\eta}\,\mathrm{d}W_s, \qquad X_0^\eta = x_{\mathrm{init}}, \tag{2.142}$$

where $\eta > 0$ is a (small) parameter, and where we set the diffusion coefficient to be the identity for notational convenience, but note that a generalization to an arbitrary $\sigma(X_s^\eta)$ in front of the Brownian motion is possible. For $\eta \to 0$ we expect our dynamics to become deterministic and to fulfill the ODE

$$\varphi'(s) = b(\varphi(s)), \qquad \varphi(0) = x_{\mathrm{init}}. \tag{2.143}$$

To be more precise, we assume that (cf. [100, Theorem 1.1])

$$\mathbb{P}\left(\lim_{\eta\to 0} \|X^\eta - \varphi\|_C = 0\right) = 1, \tag{2.144}$$

where $\|f\|_C = \sup_{s\in[0,T]} |f(s)|$ with the shorthand notation $C = C([0,T], \mathbb{R}^d)$ for the set of continuous functions. As before, we are interested in determining the rate of this convergence at an exponential level and (in analogy to Cramér's Theorem 2.37) expect a behavior like

$$\mathbb{P}\left(\|X^\eta - \varphi\|_C < \delta\right) \simeq e^{-\frac{1}{\eta} I(\varphi)} \tag{2.145}$$

for any small enough $\eta, \delta > 0$, where I is an appropriate rate functional in the sense of Definition 2.39. Proving corresponding large deviation results becomes rather technical and without going into further details let us just highlight the following theorem, which proves that a small noise large deviation principle as in (2.145) is indeed valid and specifies the corresponding rate functional.

Theorem 2.41 (Freidlin-Wentzell)**.** *The stochastic process X_s^η defined in* (2.142) *satisfies a large deviation principle with rate function $I : C^1 \to [0, \infty]$ defined by*

$$I(\varphi) = \begin{cases} \frac{1}{2}\int_0^T |\varphi'(s) - b(\varphi(s))|^2 \mathrm{d}s, & \varphi \in H^1, \\ \infty, & \text{otherwise}, \end{cases} \tag{2.146}$$

where $H^1 = \left\{\int_0^T \varphi(s)\mathrm{d}s : \varphi \in L^2([0,T])\right\}$ denotes the space of all absolutely continuous functions with value 0 *at $t = 0$ that possess a square integrable derivative.*

Proof. See [68, Theorem 5.6.3]. □

Comparing to the ODE (2.143), we note that the rate functional is zero if and only if the ODE is fulfilled and that it can therefore be interpreted as a kind of cost penalizing deviations from this most probable path.

Remark 2.42 (Asymptotically optimal variance in small noise regimes). In small noise regimes, one is often interested in computing quantities like

$$\psi^\eta = \mathbb{E}\left[e^{-\frac{1}{\eta}\mathcal{W}(X^\eta)}\right], \tag{2.147}$$

where $\mathcal{W} : C \to \mathbb{R}$ is a suitable path functional as for instance defined in (1.8) (see also Section 3.2.2). A notorious problem is that this observable can have a substantially large second moment

$$M^\eta = \mathbb{E}\left[e^{-\frac{2}{\eta}\mathcal{W}(X^\eta)}\right], \tag{2.148}$$

leading to large relative errors of Monte Carlo estimators. In order to better understand this phenomenon, let us state the following variational relations, sometimes known as Varadhan's integral theorem (see [68, Theorem 4.3.1], [290]),

$$\gamma_1 := -\lim_{\eta\to 0} \eta \log \psi^\eta = \inf_{\substack{\varphi\in C \\ \varphi(0)=x_{\text{init}}}} \{I(\varphi) + \mathcal{W}(\varphi)\} \tag{2.149}$$

and

$$\gamma_2 := -\lim_{\eta\to 0} \eta \log M^\eta = \inf_{\substack{\varphi\in C \\ \varphi(0)=x_{\text{init}}}} \{I(\varphi) + 2\,\mathcal{W}(\varphi)\}, \tag{2.150}$$

where I is the rate functional from Theorem 2.41. Just as in (2.139) and (2.140) we have $2\gamma_1 \geq \gamma_2$ by Jensen's inequality and note that $2\gamma_1 = \gamma_2$ implies that a corresponding estimator is asymptotically optimal on an exponential scale (i.e. $(\psi^\eta)^2 \simeq M^\eta$, sometimes called *log-efficient*), see Remark 2.40. Let us stress again, however, that the relative error of corresponding Monte Carlo estimators can still be large on a subexponential nonasymptotic level, which can be seen by noting that

$$\frac{\sqrt{\mathrm{Var}\left(e^{-\frac{1}{\eta}\mathcal{W}(X^\eta)}\right)}}{\mathbb{E}\left[e^{-\frac{1}{\eta}\mathcal{W}(X^\eta)}\right]} = \sqrt{\frac{M^\eta}{(\psi^\eta)^2} - 1} = \sqrt{e^{\frac{2\gamma_1-\gamma_2+o(1)}{\eta}} - 1}, \tag{2.151}$$

which can blow up even if $2\gamma_1 = \gamma_2$, for example, it can increase exponentially as $\eta^{-\beta}$ for some $\beta \in (0,1)$ (cf. [290]). We will come back to this observation in Section 3.2.2.

Remark 2.43 (Zero noise viscosity approximation). We have seen before (e.g. in Lemma 2.11) that

$$V^\eta(x,t) = -\eta \log \mathbb{E}\left[e^{-\frac{1}{\eta}\mathcal{W}(X^\eta)}\middle| X_t = x\right] \tag{2.152}$$

fulfills the HJB equation

$$\left(\partial_t + \frac{\eta}{2}\Delta + b(x,t)\cdot\nabla\right) V^\eta(x,t) - \frac{1}{2}|\nabla V^\eta(x,t)|^2 + f(x,t) = 0, \qquad V^\eta(x,T) = g(x), \tag{2.153}$$

where f and g are as in (1.8) and we recall that we have set the diffusion matrix to be the identity for notational convenience. One can now show that taking the limit

$$V^0(x,t) := \lim_{\eta\to 0} V^\eta(x,t) \tag{2.154}$$

(as in (2.149)) brings the PDE [96]

$$(\partial_t + b(x,t)\cdot\nabla)V^0(x,t) - \frac{1}{2}|\nabla V^0(x,t)|^2 + f(x,t) = 0, \qquad V^0(x,T) = g(x), \tag{2.155}$$

where the second derivative terms have disappeared, now corresponding to a deterministic rather than a stochastic optimal control problem (compare also to Section 2.1), also resulting in the representation

$$V^0(x,t) = \inf_{\substack{\varphi\in C \\ \varphi(t)=x}} \left\{ \frac{1}{2}\int_t^T |\varphi'(s) - b(\varphi(s))|^2 \mathrm{d}s + \mathcal{W}(\varphi) \right\}, \tag{2.156}$$

which reminds of a deterministic cost functional. We again refer to Section 3.2.2 for further discussion in an importance sampling context.

At the end of this subsection, let us briefly mention the other type of limit that one can take in the large deviation analysis of stochastic processes, namely letting the time horizon go to infinity [94]. To this end, we consider the stochastic process

$$\mathrm{d}X_s = b(X_s)\,\mathrm{d}s + \sigma(X_s)\,\mathrm{d}W_s, \qquad X_t = x_{\mathrm{init}}, \tag{2.157}$$

and can now define the scaled cumulant generating function as

$$C_f(\alpha) = -\lim_{T\to\infty} \frac{1}{T} \log \mathbb{E}\left[e^{-\alpha \int_0^T f(X_s)\mathrm{d}s}\right], \tag{2.158}$$

with some given $f \in C(\mathbb{R}^d, d)$, and define the rate function as

$$I_f(x) = \sup_{\alpha\in\mathbb{R}} \left\{\alpha x - C_f(\alpha)\right\}. \tag{2.159}$$

An interesting connection to Section 2.2 is that $C_f := C_f(1)$ is the principal eigenvalue of the operator $L - f$ appearing in the Feynman-Kac PDE from Theorem 2.14. Indeed, recalling the Feynman-Kac semi-group (e.g. from Remark 2.18),

$$(P_T^f g)(x) = \mathbb{E}\left[g(X_T)e^{-\int_0^T f(X_s)\mathrm{d}s}\middle| X_0 = x\right], \tag{2.160}$$

we expect

$$P_T^f g \sim_{T\to\infty} e^{-T\lambda(f)}, \tag{2.161}$$

where $\lambda(f)$ is the largest eigenvalue of $L - f$. By taking the logarithm and dividing by T, we recover the cumulant C_f, which shows that $C_f = \lambda(f)$ is indeed the largest eigenvalue of $L - f$ (see [94] for further details and [95] for an algorithm that exploits this connection numerically).

2.3.4 Variational characterization of free energy

In this subsection we will take a seemingly different perspective on our sampling problem. We will elaborate on a variational formulation of the free energy functional

$$-\log \mathbb{E}\left[e^{-\mathcal{W}(X)}\right] \tag{2.162}$$

as defined in (1.9), where the path functional $\mathcal{W}$ should be thought of as in (1.8). We will gain further insights about the variational nature of our guiding problem, having in mind that variational characterizations might lead to numerical algorithms in the spirit of machine learning, and we will recover some of the concepts from large deviations theory that we have discussed in the previous section. Eventually we will get back to where we started and once more relate the free energy (2.162) to stochastic optimal control problems as discussed in Section 2.1.

Let us start by stating the following theorem which contains two variational formulas in an abstract setting that are dual to each other.

Theorem 2.44 (Donsker–Varadhan variational principle)**.** *Let $(\Omega, \mathcal{F})$ be a measurable space, let $\mathcal{P}(\Omega)$ be the set of probability measures on $(\Omega, \mathcal{F})$, let $\nu, \widetilde{\nu} \in \mathcal{P}(\Omega)$ be measures and let $\mathcal{W} : \Omega \to \mathbb{R}$ be a measurable functional, then the following Legendre-type dualities hold. We have*

$$-\mathrm{KL}(\widetilde{\nu}|\nu) = \inf_{\mathcal{W}\in B(\Omega)} \left\{\int \mathcal{W}\,\mathrm{d}\widetilde{\nu} + \log \int e^{-\mathcal{W}}\mathrm{d}\nu\right\}, \tag{2.163}$$

where $B(\Omega)$ denotes the set of bounded, real-valued measurable functionals on Ω, and its dual

$$-\log \int e^{-\mathcal{W}}\mathrm{d}\nu = \inf_{\widetilde{\nu}\in\mathcal{P}(\Omega)} \left\{\int \mathcal{W}\,\mathrm{d}\widetilde{\nu} + \mathrm{KL}(\widetilde{\nu}|\nu)\right\}. \tag{2.164}$$

Proof. A proof can for instance be found in [61] or [69], noting that the set $B(\Omega)$ can actually be made larger. To gain some intuition for the second equation (cf. Remark 2.34), we see that Jensen's inequality brings

$$-\log \int e^{-\mathcal{W}}\mathrm{d}\nu = -\log \int e^{-\mathcal{W}} \frac{\mathrm{d}\nu}{\mathrm{d}\widetilde{\nu}}\,\mathrm{d}\widetilde{\nu} = -\log \int e^{-\mathcal{W}-\log\frac{\mathrm{d}\widetilde{\nu}}{\mathrm{d}\nu}}\mathrm{d}\widetilde{\nu} \le \int \left(\mathcal{W} + \log\frac{\mathrm{d}\widetilde{\nu}}{\mathrm{d}\nu}\right)\mathrm{d}\widetilde{\nu} = \int \mathcal{W}\,\mathrm{d}\widetilde{\nu} + \mathrm{KL}(\widetilde{\nu}|\nu), \tag{2.165}$$

where equality is attained when considering the measure $\widetilde{\nu}$ defined by $\frac{\mathrm{d}\nu^*}{\mathrm{d}\widetilde{\nu}} = \frac{e^{-\mathcal{W}}}{\int e^{-\mathcal{W}}\mathrm{d}\nu}$. □

Remark 2.45 (Relation to large deviations theory). The variational expression (2.163) can be equivalently written as

$$\mathrm{KL}(\widetilde{\nu}|\nu) = \sup_{\mathcal{W}\in B(\Omega)} \left\{ \int \mathcal{W}\,\mathrm{d}\widetilde{\nu} - \log \int e^{\mathcal{W}}\mathrm{d}\nu \right\} \tag{2.166}$$

and can therefore be compared to the Legendre-Fenchel transform of the cumulant generating function related to random variables in $\mathbb{R}$ as defined in (2.125). This can be seen by noting that we can write

$$C^*(x) = \sup_{\alpha\in\mathbb{R}} \left\{ \alpha x - \log \mathbb{E}\left[e^{\alpha X}\right] \right\} = \sup_{\alpha\in\mathbb{R}} \left\{ \alpha x - \log \int e^{\alpha x} \mathcal{L}_\nu(X)(\mathrm{d}x) \right\}, \tag{2.167}$$

where $\mathcal{L}_\nu(X)$ is the law of the random variable X, which (with an abuse of notation) is distributed according to ν, as well as

$$\int \mathcal{W}\,\mathrm{d}\widetilde{\nu} = \langle \mathcal{W}, \widetilde{\nu} \rangle, \tag{2.168}$$

and

$$\log \int e^{-\mathcal{W}}\mathrm{d}\nu = \log \int e^{\langle \mathcal{W}, \widetilde{\nu}\rangle} \mathcal{L}_\nu(\delta_X)(\mathrm{d}\widetilde{\nu}), \tag{2.169}$$

where $\mathcal{L}_\nu(\delta_X)$ is the law of the empirical measure δ_X with X being distributed according to ν (cf. [176]). We just made the observation that we can interpret the relative entropy in (2.163) as a rate functional, yielding some sort of abstract version of Cramér's Theorem 2.37, which should be compared to Sanov's theorem, making this connection more precise [176, Theorem 2.4.1].

Remark 2.46 (Path space interpretation). The abstract formulation in Theorem 2.44 allows for Ω being the space of continuous paths, as denoted by $\mathcal{C}$ before, for $\widetilde{\nu}, \nu$ being path space measures $\mathbb{P}^u, \mathbb{P}$ and for $\mathcal{W}$ being a path functional as for instance defined in (1.8) (cf. Remark 2.34). The variational formulation (2.164) can then be translated to

$$-\log \mathbb{E}\left[e^{-\mathcal{W}(X)}\right] = \inf_{\mathbb{P}^u \ll \mathbb{P}} \left\{ \int_{\mathcal{C}} \mathcal{W}\,\mathrm{d}\mathbb{P}^u + \mathrm{KL}(\mathbb{P}^u|\mathbb{P}) \right\}, \tag{2.170}$$

where X is a realization of the SDE (1.2). We well see in Chapter 4 that the perspective of path space measures will be fruitful. Recalling Girsanov's Theorem B.3 and denoting by X^u a solution to the controlled SDE (1.4), we can make the expression explicit in the control u and write

$$-\log \mathbb{E}\left[e^{-\mathcal{W}(X)}\right] = \inf_{u\in\mathcal{U}} \mathbb{E}\left[\mathcal{W}(X^u) + \frac{1}{2}\int_0^T |u(X_s^u, s)|^2 \mathrm{d}s \right] = \inf_{u\in\mathcal{U}} J(u; x_{\mathrm{init}}, 0), \tag{2.171}$$

where the set of admissible controls $\mathcal{U}$ and the cost functional J are as defined in (1.5) and (1.16), see also Proposition 4.7.

Remark 2.47 (Interpretation in statistical physics). The term *free energy* for expression (2.162) comes from statistical physics, and indeed the fact that it is the Legendre transform of the relative entropy is a well-known thermodynamic principle, as the variational formulation (2.164) furnishes the famous relation $F = U - TS$ for the Helmholtz free energy F, with U being the internal energy, T the temperature and S denoting the Gibbs entropy. If we modify the previous assumptions by setting $\mathrm{d}\nu = \mathrm{d}x$, $\mathrm{d}\widetilde{\nu} = \rho\,\mathrm{d}x$ as densities in $\mathbb{R}^d$ and assuming that $\mathcal{W} = \beta E$, where $\beta = (k_B T)^{-1}$ with $k_B > 0$ being Boltzmann's constant and E denoting a smooth potential energy function that is bounded from below and growing at infinity, then

$$\underbrace{-\beta^{-1}\log \int \exp(-\beta E)\,\mathrm{d}x}_{=F} = \min_{\rho>0} \left\{ \underbrace{\int E\rho\,\mathrm{d}x}_{=U} + \underbrace{\beta^{-1}\int \rho\log\rho\,\mathrm{d}x}_{=-TS} \right\} \tag{2.172}$$

with the unique minimizer being the Gibbs-Boltzmann density $\rho^* = \exp(-\beta E)/\mathcal{Z}$ with normalization constant $\mathcal{Z} = \exp(-\beta F)$.

Remark 2.48 (Jarzynski's equality). The Donsker–Varadhan variational principle from Theorem 2.44 shares some features with the non-equilibrium free energy formula of Jarzynski, which relates the Helmholtz equilibrium free energy to averages that are taken over an ensemble of non-equilibrium trajectories generated by forcing the dynamics [156]. An extensive analysis on the relation of the Jarzynski formula to the optimal control of diffusions in the context of variance reduction can be found in [134].

2.4 Neural networks and stochastic optimization

In this final section on the theoretical foundations we shall address a topic that is relevant for actually solving our Problems 1.1-1.5 from Chapter 1. For numerical algorithms two obvious questions are how to approximate the control functions $u : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ as well as how to approach the minimization problems appearing in the variational characterizations. Since Galerkin based approximations usually suffer from the curse of dimensionality (cf. Section 1.5), we will mostly rely on neural networks, which are known to have remarkable approximation properties and have demonstrated impressive numerical performance in various tasks. We will formally introduce feed-forward neural networks and their variants and briefly discuss some recent approximation results. Given their nested structure, minimization cannot be done in closed-form, rather it turns out that gradient descent schemes are the method of choice. Noting that some stochastic optimization tricks can improve performance significantly, we will introduce the *Adam* algorithm as a version of gradient decent that relies on estimated gradient statistics to adjust learning rates.

We start by defining a neural network. It is essentially a complicated function consisting of concatenated affine and nonlinear maps.

Definition 2.49 (Feed-forward neural network)**.** A standard *fully connected feed-forward neural network* is a function $\Phi_\varrho : \mathbb{R}^d \to \mathbb{R}^m$ given by

$$\Phi_\varrho(x) = A_L \varrho(A_{L-1}\varrho(\cdots \varrho(A_1 x + b_1)\cdots) + b_{L-1}) + b_L, \tag{2.173}$$

with matrices $A_l \in \mathbb{R}^{n_l \times n_{l-1}}$, vectors $b_l \in \mathbb{R}^{n_l}, 1 \le l \le L$, $L \in \mathbb{N}$ denoting the depth, and a nonlinear activation function $\varrho : \mathbb{R} \to \mathbb{R}$ that is to be applied componentwise. Note that $n_0 = d$ and $n_L = m$. The collection of matrices A_l and vectors b_l comprises the learnable parameters.

In practice it turns out that adding "skip connections" in between some of the nonlinear activation functions can bring numerical advantages [85, 142]. We therefore define the following version of a feed-forward neural network.

Definition 2.50 (DenseNet)**.** We define the *DenseNet* by

$$\Phi_\varrho(x) = A_L x_L + b_L, \tag{2.174}$$

where x_L is specified recursively by

$$y_{l+1} = \varrho(A_l x_l + b_l), \qquad x_{l+1} = (x_l, y_{l+1})^\top, \tag{2.175}$$

with $A_l \in \mathbb{R}^{n_l \times \sum_{i=0}^{l-1} n_i}, b_l \in \mathbb{R}^l$ for $1 \le l \le L-1$ and $x_1 = x$, $n_0 = d$. Again, the collection of matrices A_l and vectors b_l comprises the learnable parameters.

One reason for the success of neural networks are their approximation properties. This following classic theorem demonstrates that they can approximate any continuous function arbitrarily well.

Theorem 2.51 (Universal approximation theorem)**.** *Let $\varrho : \mathbb{R} \to \mathbb{R}$ be continuous, but not a polynomial. Let $d \ge 1, L = 2$ and $\mathcal{D} \subset \mathbb{R}^d$ be compact. Then for any continuous function $f : \mathbb{R}^d \to \mathbb{R}^m$ and every $\varepsilon > 0$ there exists a feed-forward neural network $\Phi_\varrho : \mathbb{R}^d \to \mathbb{R}^m$ such that*

$$\sup_{x \in \mathcal{D}} |\Phi(x) - f(x)| < \varepsilon. \tag{2.176}$$

Proof. See [191]. □

Remark 2.52. First versions of this theorem (with different conditions on the activation function) using functional analytic arguments relying for instance on the Hahn-Banach theorem and the measure formulation of the Riesz representation theorem can be found in [59, 141]. We see that even "shallow" neural networks only containing one hidden layer posses the universal approximation property. However, one should note that Theorem 2.51 theorem gives no information on the needed "width" of the network, i.e. the sizes of the matrices A_l. For a more quantitative analysis we refer to [205, 236]. Furthermore, note that there also exists a dual version of Theorem 2.51, where neural networks are allowed to have fixed width, however arbitrary depth, see e.g. [203].

Remark 2.53 (High-dimensional approximation). Neural networks cannot break the curse of dimensionality for arbitrary target functions [202, 232, 301], however there are multiple numerical experiments demonstrating that they are particularly well-suited for the approximation of high-dimensional functions. This observation is backed up by recent theoretical analysis demonstrating that the curse can be beaten at least in certain scenarios [8, 22, 53, 88, 157].

Let us denote the learnable parameters (e.g. the entries of the matrices A_l and vectors b_l in Definition 2.49) by $\theta \in \Theta \subset \mathbb{R}^p$. The idea in machine learning is to consider loss functions

$$\mathcal{L} : \Theta \to \mathbb{R} \tag{2.177}$$

that measure the performance of corresponding learning tasks. Permissible loss functions include those that admit a (not necessarily unique [28]) global minimum representing the solution to the problem at hand. They often involve expectation values of random variables and therefore rely on Monte Carlo approximation in practice. The minimization of $\mathcal{L}$ is usually approached via stochastic gradient decent (SGD), going back to [252], where in every iteration step one updates the parameter vector according to

$$\theta_{n+1} = \theta_n - \eta_n g_n, \tag{2.178}$$

with η_n denoting the learning rate and g_n an unbiased estimate of the gradient, i.e. $\mathbb{E}[g_n] = \nabla_\theta \mathcal{L}$. Analysis and improvements of such schemes have been the focus of several studies [37]. One major challenge is that even though losses might be convex in function space, they are usually non-convex in parameter space such that the optimization can get stuck in local minima or take a very long time due to high variances of gradient estimators. Many versions of SGD, often relying on heuristic tricks, have been suggested that try to improve convergence. We shall highlight the *Adam* algorithm [169], somehow combining the idea of gradient aggregation (i.e. reusing previously computed gradients) and gradient scaling in the spirit of second-order methods. We can interpret the algorithm as performing an online adjustment of the learning rate η_n according to a heuristic that takes into account exponential moving averages of the gradients and the squared gradients computed in previous iteration steps. More precisely, one computes

$$m_n = \frac{(1-\beta_1)}{(1-\beta_1^n)} \sum_{i=1}^{n} \beta_1^{n-i} g_i, \qquad v_n = \frac{(1-\beta_2)}{(1-\beta_2^n)} \sum_{i=1}^{n} \beta_2^{n-i} g_i \odot g_i, \tag{2.179}$$

where $g_i \in \mathbb{R}^p$ is the estimator of the gradient at iteration i, $\odot$ denotes elementwise multiplication and $\beta_1, \beta_2 \in [0, 1)$ are hyperparameters. A parameter update is then performed via

$$\theta_{n+1}^{(k)} = \theta_n^{(k)} - \eta_n \frac{m_n^{(k)}}{\sqrt{v_n^{(k)}} + \varepsilon} \tag{2.180}$$

for each $k \in \{1, \dots, p\}$, where $0 < \varepsilon \ll 1$ aims at avoiding a division by zero.

Chapter 3

Nonasymptotic bounds for suboptimal importance sampling

The goal of this chapter is to quantify non-robustness issues of importance sampling. We have introduced importance sampling as a classical method for variance reduction in Section 2.3, where the idea is to sample from an alternative probability measure and reweight the resulting random variables with the likelihood ratio in order to produce an unbiased estimator for the quantity of interest. In theory, under appropriate assumptions, there exists an optimal proposal that yields a zero-variance estimator and therefore removes all the stochasticity from the problem. At the same time, however, any suboptimal choice harbours the risk of actually increasing the relative error of the estimator significantly, therefore counteracting the original intention. To better understand this robustness (or better: fragility) of the optimal proposal in importance sampling is the main goal of this chapter.

In applications, one often faces situations where the probability measures admit densities on a subset of $\mathbb{R}^d$ or a function space like the space of (semi-)continuous trajectories with values in $\mathbb{R}^d$, called path space. We shall put special emphasis on the latter case, specifically on diffusion processes, where additional challenges, such as small noises or metastable dynamics, might occur. We will provide quantitative bounds on the relative error of importance sampling estimators that explain the fragility of importance sampling in these situations. Some of those bounds are formulated on an abstract measurable space, but they can be readily applied to the density case. For the path space measures, we deduce some additional bounds that, in particular, highlight the challenges due to high dimensionality or long trajectories.

The chapter is structured as follows. In Section 3.1 we recall the notions of divergences between proposal and target measures, while refining a bound on the relative error and highlighting robustness issues in high dimensions. In Section 3.2 we translate the bounds from the previous section to this setting and derive an exact formula for the relative error with which we can state novel bounds that allow for interpretations with respect to robustness in higher dimensions and long time horizons. When focusing on PDE methods in Section 3.2.1 we can essentially re-derive bounds from the previous section. In Section 3.2.2 we comment on how our bounds can help to understand potential issues in the small noise regime. Finally, in Section 3.3 we present a couple of numerical examples with which we illustrate the previously discussed issues. Additional auxiliary statements and technical lemmas can be found in Appendix B.4.

Parts of this chapter have been done in collaboration with Carsten Hartmann and published in [129].

3.1 Importance sampling bounds based on divergences

Let us recall from Section 2.3 that one relevant quantity for the study of importance sampling estimators of

$$\mathcal{Z} = \mathbb{E}\left[e^{-\mathcal{W}(\widetilde{X})}\frac{\mathrm{d}\nu}{\mathrm{d}\widetilde{\nu}}(\widetilde{X})\right] \tag{3.1}$$

is the relative error

$$r(\widetilde{\nu}) = \frac{\sqrt{\mathrm{Var}\left(e^{-\mathcal{W}(\widetilde{X})}\frac{\mathrm{d}\nu}{\mathrm{d}\widetilde{\nu}}(\widetilde{X})\right)}}{\mathcal{Z}}, \tag{3.2}$$

as already defined in (2.99), where ν is the base measure, according to which our random variables are distributed, and $\widetilde{\nu}$ is some absolutely continuous proposal measure. It can be readily seen that choosing the optimal proposal measure $\widetilde{\nu} = \nu^*$ defined via

$$\frac{\mathrm{d}\nu^*}{\mathrm{d}\nu} = \frac{e^{-\mathcal{W}}}{\mathcal{Z}} \tag{3.3}$$

yields an unbiased zero-variance estimator, which, however, is infeasible in practice, as $\mathcal{Z}$ is just the quantity we are after, and therefore not available. In this section, we study the relative error when using any other absolutely continuous, suboptimal proposal measure $\widetilde{\nu} \neq \nu^*$. It turns out that divergences between those measures are helpful in this analysis, in particular the following two.

Definition 3.1 (Kullback-Leibler divergence)**.** Let ν_1, ν_2 be probability measures. The Kullback-Leibler (KL) divergence is defined as[19]

$$\mathrm{KL}(\nu_1|\nu_2) = \begin{cases} \mathbb{E}_{\nu_1}\left[\log \frac{\mathrm{d}\nu_1}{\mathrm{d}\nu_2}\right], & \text{if } \nu_1 \ll \nu_2, \\ \infty, & \text{else.} \end{cases} \tag{3.4}$$

Definition 3.2 (χ^2 divergence)**.** Let ν_1, ν_2 be probability measures. The χ^2 divergence is defined as

$$\chi^2(\nu_1|\nu_2) = \begin{cases} \mathbb{E}_{\nu_2}\left[\left(\frac{\mathrm{d}\nu_1}{\mathrm{d}\nu_2}\right)^2 - 1\right], & \text{if } \nu_1 \ll \nu_2, \\ \infty, & \text{else.} \end{cases} \tag{3.5}$$

We start by noting the equivalence of the squared relative error and the χ^2 divergence between the actual and the optimal proposal measure.

Lemma 3.3 (Equivalence with χ^2 divergence)**.** *Let $\widetilde{\nu}$ be a measure that is absolutely continuous with respect to ν, let ν^* be the optimal proposal measure as defined in* (3.3) *and let $r(\widetilde{\nu})$ be the relative error as in* (3.2)*. Then*

$$r^2(\widetilde{\nu}) = \chi^2(\nu^*|\widetilde{\nu}). \tag{3.6}$$

Proof. By using the definition of the χ^2 divergence in the first step, we compute

$$\chi^2(\nu^*|\widetilde{\nu}) = \mathbb{E}_{\widetilde{\nu}}\left[\left(\frac{\mathrm{d}\nu^*}{\mathrm{d}\widetilde{\nu}}\right)^2 - 1\right] = \mathbb{E}_{\widetilde{\nu}}\left[\left(\frac{\mathrm{d}\nu^*}{\mathrm{d}\widetilde{\nu}}\right)^2\right] - \mathbb{E}_{\widetilde{\nu}}\left[\frac{\mathrm{d}\nu^*}{\mathrm{d}\widetilde{\nu}}\right]^2 \tag{3.7a}$$

$$= \mathrm{Var}_{\widetilde{\nu}}\left(\frac{\mathrm{d}\nu^*}{\mathrm{d}\widetilde{\nu}}\right) = \frac{1}{\mathcal{Z}^2}\,\mathrm{Var}\left(e^{-\mathcal{W}(\widetilde{X})}\frac{\mathrm{d}\nu}{\mathrm{d}\widetilde{\nu}}(\widetilde{X})\right) = r^2(\widetilde{\nu}). \tag{3.7b}$$

□

Motivated by known bounds on the χ^2 divergence, we can formulate our first statement, where we quantify the suboptimality by the Kullback-Leibler divergence between the actual and the optimal proposal measure.

Proposition 3.4 (Lower bound on relative error)**.** *Let $\mathcal{W} : \widetilde{\Omega} \to \mathbb{R}$, let $\widetilde{\nu}$ be a measure and let ν^* be the optimal proposal measure as defined in* (3.3)*, then for the relative error* (3.2) *it holds*

$$r(\widetilde{\nu}) \geq \sqrt{e^{\mathrm{KL}(\nu^*|\widetilde{\nu})} - 1}. \tag{3.8}$$

Proof. With Jensens's inequality we have

$$\mathrm{KL}(\nu^*|\widetilde{\nu}) = \mathbb{E}_{\nu^*}\left[\log \frac{\mathrm{d}\nu^*}{\mathrm{d}\widetilde{\nu}}\right] \leq \log \mathbb{E}_{\nu^*}\left[\frac{\mathrm{d}\nu^*}{\mathrm{d}\widetilde{\nu}}\right]. \tag{3.9}$$

Combining this with Lemma 3.3 yields

$$r^2(\widetilde{\nu}) = \mathbb{E}_{\widetilde{\nu}}\left[\left(\frac{\mathrm{d}\nu^*}{\mathrm{d}\widetilde{\nu}}\right)^2 - 1\right] = \mathbb{E}_{\nu^*}\left[\frac{\mathrm{d}\nu^*}{\mathrm{d}\widetilde{\nu}} - 1\right] \geq e^{\mathrm{KL}(\nu^*|\widetilde{\nu})} - 1 \tag{3.10}$$

and therefore the desired statement. □

[19] As a remark on our notation, let us mention that we sometimes endow the expectation operator with a subscript indicating with respect to which measure the expectation is taken, e.g. $\mathbb{E}_\nu$ indicates that the expectation is considered with respect to the measure ν. When explicitly writing down the corresponding random variable, e.g. $\mathbb{E}[X]$, it is usually clear from the context with respect to which measure the expectation shall be understood, and we omit the subscript (see also Remark 4.2).

Remark 3.5 (Bounds on the χ^2 divergence). In the setting of importance sampling the χ^2 divergence also appears in [52]. A bound of the χ^2 divergence that is sometimes used is $\chi^2(\nu^*|\widetilde{\nu}) \geq \mathrm{KL}(\nu^*|\widetilde{\nu})$, which is essentially based on $x \leq e^{x-1}$ and therefore yields a less tight bound compared to Proposition 3.4. The exponential bound we use instead can for instance be found in [78, Theorem 4] and [261, Proposition 4] in a discrete setting; here, a lower bound in terms of the total variation distance is provided as well. [104] offers a continuous version and some other helpful relations between divergences. An application of the bound to importance sampling relative errors can be found in [3] and more analysis with respect to more general f-divergences has been done in [260]. The statement should also be compared to the results in [51], where the required sample size of importance sampling is proved to be exponentially large in the KL divergence between the proposal and the target measure.

Remark 3.6 (Cross-entropy method). Note that the expression $\mathrm{KL}(\nu^*|\widetilde{\nu})$ appearing in (3.8) is exactly the quantity that is minimized in the so-called cross-entropy method [63, 308], which aims at approximating the optimal importance sampling proposal in a family of reference proposals.

Remark 3.7 (Exponential dependence on the dimension). We recall that the KL divergence usually gets larger with increasing state space dimension as can for instance be seen by Lemma B.7 in the appendix, implying that importance sampling is especially difficult in high dimensional settings. Another way of noting bad scaling behavior in high dimensions, similar to Proposition 4.29, is the following. Assume[20]

$$\widetilde{\nu} = \bigotimes_{i=1}^{d} \widetilde{\nu}_i, \qquad \nu^* = \bigotimes_{i=1}^{d} \nu_i^*, \tag{3.11}$$

where each $\widetilde{\nu}_i$, and ν_i^* respectively, shall be identical for $i \in \{1, \dots, d\}$. Then

$$r^2(\widetilde{\nu}) = \mathrm{Var}_{\widetilde{\nu}}\left(\frac{\mathrm{d}\nu^*}{\mathrm{d}\widetilde{\nu}}\right) = \mathbb{E}_{\widetilde{\nu}_i}\left[\left(\frac{\mathrm{d}\nu_i^*}{\mathrm{d}\widetilde{\nu}_i}\right)^2\right]^d - \mathbb{E}_{\widetilde{\nu}_i}\left[\frac{\mathrm{d}\nu_i^*}{\mathrm{d}\widetilde{\nu}_i}\right]^{2d} = \mathbb{E}_{\widetilde{\nu}_i}\left[\left(\frac{\mathrm{d}\nu_i^*}{\mathrm{d}\widetilde{\nu}_i}\right)^2\right]^d - 1 \geq C^d - 1, \tag{3.12}$$

where $C := \mathbb{E}_{\widetilde{\nu}_i}\left[\left(\frac{\nu_i^*}{\widetilde{\nu}_i}\right)^2\right] > 1$ if $\widetilde{\nu} \neq \nu^*$ due to Jensen's inequality. This can be compared to [260, Section 5.2.1], and, to be fair, we should note that also naive sampling, i.e. choosing $\widetilde{\nu} = \nu$, usually leads to an exponential dependency of the relative error on the dimension.

We have so far constructed a lower bound for the relative error. In order to get an upper bound, let us first state the following version of a generalized Jensen inequality, which will turn out to be helpful and is essentially borrowed from [209, Theorem 2].

Proposition 3.8 (Generalized Jensen inequality). *Let λ and ν be measures on $(\widetilde{\Omega}, \widetilde{\mathcal{F}})$,*

$$\mathcal{J}(f, \nu, \varphi) := \mathbb{E}_\nu\left[f(\varphi)\right] - f\left(\mathbb{E}_\nu\left[\varphi\right]\right) \tag{3.13}$$

be the normalized Jensen functional, where $f : \mathbb{R} \to \mathbb{R}$ is convex and $\varphi : \widetilde{\Omega} \to \mathbb{R}$, and let $m = \inf_{E \in \mathcal{F}} \frac{\nu(E)}{\lambda(E)}$, $M = \sup_{E \in \mathcal{F}} \frac{\nu(E)}{\lambda(E)}$. Then

$$m\,\mathcal{J}(f, \lambda, \varphi) \leq \mathcal{J}(f, \nu, \varphi) \leq M\mathcal{J}(f, \lambda, \varphi). \tag{3.14}$$

Proof. See Appendix C.2. □

We can now derive an upper bound as well as a tighter lower bound for the relative error.

Proposition 3.9 (Refined bounds on relative error). *Let $\widetilde{\nu}$ be a measure that is absolutely continuous with respect to ν and let ν^* be the optimal proposal measure as in* (3.3)*. Let m and M be as defined in Proposition 3.8 (with the measures ν and λ being replaced by $\widetilde{\nu}$ and ν^* respectively). Then for the relative error* (3.2) *it holds*

$$\sqrt{e^{m\,\mathrm{KL}(\widetilde{\nu}|\nu^*) + \mathrm{KL}(\nu^*|\widetilde{\nu})} - 1} \leq r(\widetilde{\nu}) \leq \sqrt{e^{M\,\mathrm{KL}(\widetilde{\nu}|\nu^*) + \mathrm{KL}(\nu^*|\widetilde{\nu})} - 1}. \tag{3.15}$$

Proof. Inspired by [262] (which focuses on a discrete probability space) we choose $\nu = \nu^*, \lambda = \widetilde{\nu}, \varphi = \frac{\mathrm{d}\nu^*}{\mathrm{d}\widetilde{\nu}}$ and $f(x) = -\log(x)$ for the expressions in (3.14) in order to get

$$\mathcal{J}(f, \nu^*, \varphi) = -\mathbb{E}_{\nu^*}\left[\log\left(\frac{\mathrm{d}\nu^*}{\mathrm{d}\widetilde{\nu}}\right)\right] + \log\left(\mathbb{E}_{\nu^*}\left[\frac{\mathrm{d}\nu^*}{\mathrm{d}\widetilde{\nu}}\right]\right) = -\mathrm{KL}(\nu^*|\widetilde{\nu}) + \log\left(\chi^2(\nu^*|\widetilde{\nu}) + 1\right), \tag{3.16}$$

$$\mathcal{J}(f, \widetilde{\nu}, \varphi) = -\mathbb{E}_{\widetilde{\nu}}\left[\log\left(\frac{\mathrm{d}\nu^*}{\mathrm{d}\widetilde{\nu}}\right)\right] + \log\left(\mathbb{E}_{\widetilde{\nu}}\left[\frac{\mathrm{d}\nu^*}{\mathrm{d}\widetilde{\nu}}\right]\right) = \mathrm{KL}(\widetilde{\nu}|\nu^*). \tag{3.17}$$

[20] The factorization of the optimal proposal measure ν^* assumes a factorization of the quantity e^{-g}.

With Proposition 3.8 we then get

$$m\,\mathrm{KL}(\widetilde{\nu}|\nu^*) + \mathrm{KL}(\nu^*|\widetilde{\nu}) \leq \log\left(\chi^2(\nu^*|\widetilde{\nu}) + 1\right) \leq M\,\mathrm{KL}(\widetilde{\nu}|\nu^*) + \mathrm{KL}(\nu^*|\widetilde{\nu}) \tag{3.18}$$

and with Lemma 3.3 our statement follows. □

Remark 3.10. One should note that m and M depend on $\widetilde{\nu}$ and ν^*, respectively, and are hard to compute in practice. We have $m \in [0,1]$ and $M \in [1,\infty]$ and indeed it is possible to get $m = 0$ or $M = \infty$. The former case brings back the ordinary Jensen inequality and the lower bound from Proposition 3.9 is then equivalent to the one from Proposition 3.4. The case $M = \infty$ on the other hand yields a trivial upper bound, for which we provide an illustration in Example 3.11.

Example 3.11 (Upper bound for relative error)**.** *In order to illustrate the case where the upper bound in Proposition 3.9 becomes meaningless, consider for instance the measure ν on $[1,\infty) \subset \mathbb{R}$ admitting the one-dimensional density $p(x) = \alpha\frac{1}{x^{\alpha+1}}$ defined for $x \geq 1$. This density is special since for $\alpha \leq 1$ we have $\mathbb{E}[X] = \infty$, however for $\alpha \in (1,2)$ it holds $\mathbb{E}[X] < \infty$, whereas still $\mathbb{E}\left[X^2\right] = \infty$ and therefore $\mathcal{J}(x \mapsto x^2, \nu, \varphi) = \infty$ for $\varphi(x) = x$. Now Proposition 3.8 implies that the upper bound also has to be infinity. Let us illustrate this for the particular choice of the measure λ admitting the density $q(x) = 2\alpha\frac{1}{x^{2\alpha+1}}$. For this choice we have $\mathcal{J}(x \mapsto x^2, \lambda, x \mapsto x) < \infty$ for $\alpha \in (1,2)$, however we compute*

$$M = \sup_{\substack{a,b\in[1,\infty]\\ a\neq b}} \frac{\int_a^b p(x)\mathrm{d}x}{\int_a^b q(x)\mathrm{d}x} \geq \sup_{a\in[1,\infty)} \frac{\int_a^\infty p(x)\mathrm{d}x}{\int_a^\infty q(x)\mathrm{d}x} = \sup_{a\in[1,\infty)} \frac{\frac{1}{a^\alpha}}{\frac{1}{a^{2\alpha}}} = \sup_{a\in[1,\infty)} a^\alpha = \infty. \tag{3.19}$$

In fact Proposition 3.8 implies that one cannot not find any λ for which both $\mathcal{J}(x \mapsto x^2, \lambda, x \mapsto x)$ and M are finite.

To conclude this section, let us illustrate our bounds by looking at a concrete example using Gaussians on $\widetilde{\Omega} = \mathbb{R}^d$ (which should be compared to [206, Section 6]).

Example 3.12 (High-dimensional Gaussians)**.** *Suppose we want to compute $\mathbb{E}\left[e^{-\alpha\cdot X}\right]$, with a given vector $\alpha \in \mathbb{R}^d$, where $X \sim \mathcal{N}(\mu,\Sigma) =: p$ is distributed according to a multidimensional Gaussian with mean $\mu \in \mathbb{R}^d$ and covariance matrix $\Sigma \in \mathbb{R}^{d\times d}$. Then the optimal importance sampling density is given by*

$$p^*(x) = \frac{e^{-\alpha\cdot x}}{\mathcal{Z}}p(x) = \mathcal{N}(x;\mu - \Sigma\alpha,\Sigma). \tag{3.20}$$

If we however sample from a perturbed version

$$\widetilde{p}^\varepsilon := \mathcal{N}(x;\mu - \Sigma(\alpha+\varepsilon),\Sigma) \tag{3.21}$$

with a vector $\varepsilon \in \mathbb{R}^d$, we get the relative error

$$r(\widetilde{p}^\varepsilon) = \frac{1}{\mathcal{Z}}\sqrt{\mathrm{Var}\left(e^{-\alpha\cdot\widetilde{X}}\frac{p}{\widetilde{p}^\varepsilon}(\widetilde{X})\right)} = \sqrt{e^{\varepsilon\cdot\Sigma\varepsilon} - 1}. \tag{3.22}$$

In this particular case, the computations can be compared to the relative error of a log-normally distributed random variable, see Appendix B.4.1. Taking, for instance, $\varepsilon = (\widetilde{\varepsilon},\cdots,\widetilde{\varepsilon})^\top, \Sigma = \mathrm{diag}(\sigma^2,\cdots,\sigma^2)$ yields

$$r(\widetilde{p}^\varepsilon) = \sqrt{e^{d\sigma^2\widetilde{\varepsilon}^2} - 1}, \tag{3.23}$$

where we see an exponential dependence on the variance σ^2, the squared suboptimality parameter $\widetilde{\varepsilon}$ and the dimension d. This implies that, in order to control the relative error in high dimensions, any suboptimal importance sampling estimator needs about $K = \mathcal{O}(e^{d\sigma^2\widetilde{\varepsilon}^2})$ independent realizations to reach convergence. This observation is in agreement with the seminal result of Bengtsson and Bickel [23] that any importance sampling estimator for Gaussians ceases to be asymptotically efficient when $\log(K)/d \to 0$ as $K,d \to \infty$ (see also [192, Thm. 3.1]).

For this example, we can also apply the bound from Proposition 3.4, by noting that $\mathrm{KL}(p^*|\widetilde{p}^\varepsilon) = \frac{1}{2}\varepsilon\cdot\Sigma\varepsilon$, and get

$$r(\widetilde{p}^\varepsilon) \geq \sqrt{e^{\frac{1}{2}\varepsilon\cdot\Sigma\varepsilon} - 1}. \tag{3.24}$$

A comparison to the exact quantity (3.22) reveals that this lower bound is not tight. For an application of Proposition 3.9 we note that also $\mathrm{KL}(\widetilde{p}^\varepsilon|p^*) = \frac{1}{2}\varepsilon\cdot\Sigma\varepsilon$, however m and M are intractable. Still, it is intuitively clear that m becomes smaller and M larger, the more the two Gaussians are apart from each other.

We made the particular choice of $\widetilde{p}^{\varepsilon}$ in (3.21) in order to have an analogy to the path measure setting, which we will discuss in the next section. In fact, the added term $\Sigma\varepsilon$ in (3.21) can be compared to a constant control $\sigma\sigma^{\top}\varepsilon$ in a stochastic process as in (3.25), which, as will be seen in (3.43), yields a completely analog expression for the relative error, noting that standard d-dimensional Brownian motion is distributed according to $W_T \sim \mathcal{N}(0, \Sigma)$ with $\Sigma = T\,\mathrm{Id}_{d\times d}$.

3.2 Suboptimal control of stochastic processes and bounds for the relative error

Let us now apply our abstract bounds from the previous section to importance sampling of diffusions given by

$$\mathrm{d}X_s = b(X_s, s)\,\mathrm{d}s + \sigma(X_s, s)\,\mathrm{d}W_s, \qquad X_t = x_{\mathrm{init}}, \tag{3.25}$$

as in (1.4), by considering their controlled counterparts

$$\mathrm{d}X^u_s = (b(X^u_s, s) + \sigma(X^u_s, s)u(X^u_s, s))\,\mathrm{d}s + \sigma(X^u_s, s)\,\mathrm{d}W_s, \qquad X^u_t = x_{\mathrm{init}}, \tag{3.26}$$

as in (2.105), noting the correspondence between u and the path measure $\mathbb{P}^u$ (see also Section 2.3.2). The importance sampling attempt in (3.1) translates to

$$\mathcal{Z} = \mathbb{E}\left[e^{-\mathcal{W}(X^u)}\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^u}(X^u)\right] \tag{3.27}$$

and we recall from Section 2.3.2 that our quantity of interest, the relative error

$$r(u) = \frac{\sqrt{\mathrm{Var}\left(e^{-\mathcal{W}(X^u)}\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^u}(X^u)\right)}}{\mathcal{Z}}, \tag{3.28}$$

now depends on the control u. Given suitable conditions, there exists $u^* \in \mathcal{U}$ that brings (3.28), the relative error of the importance sampling estimator, to zero, see Theorem 2.33. We recall that an optimal path measure $\mathbb{P}^{u^*} = \mathbb{Q}$ can be defined via

$$\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}} = \frac{e^{-\mathcal{W}}}{\mathcal{Z}}, \tag{3.29}$$

in analogy to (3.3), assuming f and g are such that $\mathcal{Z}$ is finite.

Let us now bring an example that shall illustrate why variance reduction methods are indispensable in certain SDE settings.

Example 3.13 (Rare events of SDEs)**.** *Monte Carlo estimation gets particularly challenging when considering rare events. As a prominent example, let us consider the one-dimensional Langevin dynamics*

$$\mathrm{d}X_s = -\nabla\Psi(X_s)\,\mathrm{d}s + \sqrt{\eta}\,\mathrm{d}W_s, \qquad X_0 = x_{\mathrm{init}}, \tag{3.30}$$

with double well potential $\Psi(x) = \kappa(x^2-1)^2, \kappa > 0$, and noise coefficient $\eta > 0$, as illustrated in Figure 3.1 (cf. Example 1.1 and Sections 4.4.4, 6.2.3.6).

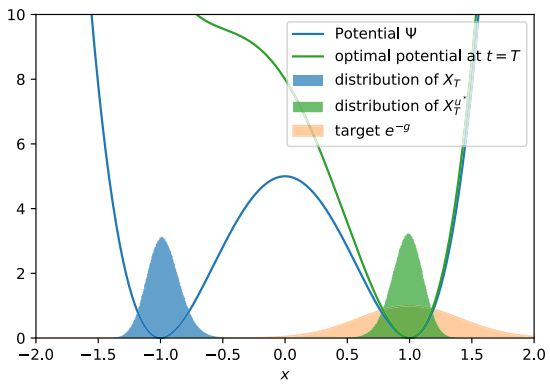


Figure 3.1: Illustration of rare events in a metastable double well potential. We consider the problem described in Example 3.13 with $\kappa = 5, \rho = 3$ on a time horizon $T = 10$ and display the distributions of X_T as well as $X_T^{u^*}$, which is controlled with the optimal importance sampling control u^* yielding a time-dependent optimal potential.

We suppose that the dynamics starts in the left well and choose a function g such that e^{-g} is concentrated in the right well, e.g. $g(x) = \rho(x-1)^2, \rho > 0$. We are interested in computing $\mathbb{E}[\exp(-g(X_T))|X_0 = x_{\text{init}}]$ for, say, $x_{\text{init}} = -1$. To understand the difficulties associated with this sampling problem, let p_T be the law of X_T for some $T > 0$ and recall that the optimal change of measure is given by the (unnormalized) likelihood $\mathrm{d}q_T/\mathrm{d}p_T \propto \exp(-g)$ that is concentrated in the right well (cf. Remark 2.35). However, regions where $\exp(-g)$ is strongly supported have probability close to zero under p_T, for p_T drops to zero quickly for $x > 0$. This can be seen as follows: Let τ be the first exit time of the set $\mathcal{D} = \{x \colon x \leq 0\}$. By Kramer's law [26], the mean first exit time (MFET) satisfies the large deviations asymptotics $\mathbb{E}[\tau] \simeq \exp(2\Delta\Psi/\eta)$ as $\eta \to 0$, where $\Delta\Psi$ is the energy barrier that the dynamics has to overcome to leave the set $\mathcal{D}$, and it turns out that the MFET is independent of the initial condition $x \in \mathcal{D}$. Therefore

$$\lim_{\eta \to 0} \eta \log \mathbb{P}(\tau < T) = -2\Delta\Psi\,, \quad T \ll \mathbb{E}[\tau]\,, \tag{3.31}$$

which is a straightforward consequence of Kramer's law, combined with the Donsker-Varadhan large deviations principle that, for a system of the form (3.30), states that $\mathbb{P}(\tau < T) \simeq 1 - \exp(\lambda_1 T)$ as $T \to \infty$ and $\eta \to 0$, where $\lambda_1 \simeq -1/\mathbb{E}[\tau]$ is the principal eigenvalue of the infinitesimal generator associated with (3.30); see, e.g. [41].

Now, by (3.31)*, we can conclude that $p_T(x) \simeq \exp(-2\Delta\Psi/\eta)$ for $x > 0$. Since p_T is essentially supported on $(-\infty, 1]$, we can approximate $\exp(-g(x))$ by a step function $\mathbb{1}_{\{x \in \mathcal{D}^c\}}$ on $x \in (-\infty, 1]$ and it thus follows that (up to exponentially small error) the relative error for small $\eta > 0$ can be approximated by*

$$r(0) = \sqrt{\frac{\mathbb{E}[\exp(-2g(X_T))] - \mathbb{E}[\exp(-g(X_T))]^2}{\mathbb{E}[\exp(-g(X_T))]^2}} \tag{3.32a}$$

$$\approx \sqrt{\frac{\exp(-2\Delta\Psi/\eta) - \exp(-4\Delta\Psi/\eta)}{\exp(-4\Delta\Psi/\eta)}} \approx \exp(\Delta\Psi/\eta)\,. \tag{3.32b}$$

This kind of exponential behavior is typical for rare event simulation and metastable systems like (3.30). So unless our terminal time T is very large or the energy barrier rather small, X_T is usually mostly supported on the left side of the well and therefore does not overlap very much with e^{-g}, which leads to an extremely large relative error. Note that this problem gets even more severe with growing values of κ and ρ.

We have stated that, given suitable conditions, there exists $u^* \in \mathcal{U}$ that brings (3.28), the relative error of the importance sampling estimator, to zero. However, in practice, u^* is usually not available (just as ν^* is not available in the abstract setting). Let us instead consider the setting where we have the control $u \in \mathcal{U}$ at hand. We want to investigate how the relative error (3.28) behaves depending on how far from optimal u is. For the upcoming analysis, it will turn out that it makes sense to measure the suboptimality and therefore the difference between $\mathbb{P}^u$ and $\mathbb{P}^{u^*}$ in terms of the difference $\delta := u^* - u$. The first statement is an implication of Proposition 3.4.

Corollary 3.14 (Lower bound for relative error on path space)**.** *Consider the path measures $\mathbb{P}^u, \mathbb{P}^{u^*} \in \mathcal{P}(\mathcal{C})$ as previously defined and let $\delta = u^* - u$. For the relative error* (3.28) *it holds*

$$r(u) \geq \left(\exp\left(\mathrm{KL}(\mathbb{P}^{u^*}|\mathbb{P}^u)\right) - 1\right)^{\frac{1}{2}} \tag{3.33}$$

and therefore

$$r(u) \geq \left(\exp\left(\mathbb{E}\left[\frac{1}{2}\int_0^T |\delta(X_s^{u^*}, s)|^2 \mathrm{d}s\right]\right) - 1\right)^{\frac{1}{2}}. \tag{3.34}$$

Proof. The first statement is just Proposition 3.4 with the abstract measures replaced by path measures. The second statement then follows from Girsanov's theorem as stated in Theorem B.3. □

One can of course also transfer the more general bound from Proposition 3.9 to path measures, however, the computations of the quantities m and M seem even more difficult and impractical than in the density case. In order to still find tighter and more applicable bounds, let us now identify an exact formula for the relative error in the SDE setting.

Proposition 3.15 (Formula for path space relative error)**.** *Let X_s^u be the solution to SDE* (3.26) *and let $\delta = u^* - u$. Then the relative error* (3.28) *is given by*

$$r(u) = \left(\mathbb{E}\left[\exp\left(-\int_0^T |\delta(X_s^u, s)|^2 \mathrm{d}s + 2\int_0^T \delta(X_s^u, s) \cdot \mathrm{d}W_s\right)\right] - 1\right)^{\frac{1}{2}}, \tag{3.35}$$

or equivalently

$$r(u) = \left(\mathbb{E}\left[\exp\left(\int_0^T |\delta(X_s^{u+2\delta}, s)|^2 \mathrm{d}s \right) \right] - 1 \right)^{\frac{1}{2}}. \tag{3.36}$$

Proof. The proof can be found in Appendix C.2. Alternatively, the second statement follows as well from Proposition 3.22. □

Remark 3.16. We note that in formula (3.35) the forward process is controlled by u, whereas in (3.36) it is controlled by $u + 2\delta = 2u^* - u$, which of course is usually not available in practice. In the upcoming Corollary 3.18 we will see how we can still make use of the formula.

Remark 3.17. Note that Proposition 3.15 entails Corollary 3.14 since

$$\mathbb{E}_{\mathbb{P}^u}\left[\left(\frac{\mathrm{d}\mathbb{P}^{u^*}}{\mathrm{d}\mathbb{P}^u} \right)^2 \right] = \mathbb{E}_{\mathbb{P}^{u^*}}\left[\frac{\mathrm{d}\mathbb{P}^{u^*}}{\mathrm{d}\mathbb{P}^u} \right] = \mathbb{E}\left[\exp\left(\frac{1}{2}\int_0^T |\delta(X_s^{u^*}, s)|^2 \mathrm{d}s + \int_0^T \delta(X_s^{u^*}, s) \cdot \mathrm{d}W_s \right) \right] \tag{3.37a}$$

$$\geq \exp\left(\mathbb{E}\left[\frac{1}{2}\int_0^T |\delta(X_s^{u^*}, s)|^2 \mathrm{d}s \right] \right). \tag{3.37b}$$

Without the change of the measures as in (3.37a) we obtain

$$\mathbb{E}_{\mathbb{P}^u}\left[\left(\frac{\mathrm{d}\mathbb{P}^{u^*}}{\mathrm{d}\mathbb{P}^u} \right)^2 \right] \geq \exp\left(\mathbb{E}\left[-\int_0^T |\delta(X_s^u, s)|^2 \mathrm{d}s \right] \right), \tag{3.38}$$

where now the process is controlled by u, however this expression has a negative sign in the exponential and is therefore rather useless. The bound

$$\mathbb{E}_{\mathbb{P}^u}\left[\left(\frac{\mathrm{d}\mathbb{P}^{u^*}}{\mathrm{d}\mathbb{P}^u} \right)^2 \right] = \mathbb{E}\left[\exp\left(\int_0^T |\delta(X_s^{u+2\delta}, s)|^2 \mathrm{d}s \right) \right] \geq \exp\left(\mathbb{E}\left[\int_0^T |\delta(X_s^{u+2\delta}, s)|^2 \mathrm{d}s \right] \right), \tag{3.39}$$

on the other hand, seems more useful.

The following corollary derives bounds from the previous Proposition 3.15 that might be useful in practice.

Corollary 3.18 (Bounds for path space relative error)**.** *Let again $\delta = u^* - u$ and let us assume there exist functions $h_1, h_2 : [0,T] \to \mathbb{R}$ such that*

$$h_1(t) \leq |\delta(x,t)| \leq h_2(t) \tag{3.40}$$

for all $x \in \mathbb{R}^d, t \in [0,T]$, then

$$\left(\exp\left(\int_0^T h_1^2(s)\mathrm{d}s \right) - 1 \right)^{\frac{1}{2}} \leq r(u) \leq \left(\exp\left(\int_0^T h_2^2(s)\mathrm{d}s \right) - 1 \right)^{\frac{1}{2}}. \tag{3.41}$$

In particular, if

$$\widetilde{\varepsilon}_1 \leq |\delta_i(x,t)| \leq \widetilde{\varepsilon}_2 \tag{3.42}$$

for all components $i \in \{1, \dots, d\}$ and for all $(x,t) \in \mathbb{R}^d \times [0,T]$ with $\widetilde{\varepsilon}_1, \widetilde{\varepsilon}_2 \in \mathbb{R}$, then

$$\left(e^{d\widetilde{\varepsilon}_1^2 T} - 1 \right)^{\frac{1}{2}} \leq r(u) \leq \left(e^{d\widetilde{\varepsilon}_2^2 T} - 1 \right)^{\frac{1}{2}}. \tag{3.43}$$

Proof. Both statements follow directly from equation (3.36) in Proposition 3.15 by noting that the dependence on the stochastic process and therefore the expectation disappears if we consider bounds on δ that do not depend on x. Two alternative proofs of the corresponding statements can be found in Appendix C.2. □

Remark 3.19. Note that bounding the suboptimality δ for all x can be a strong assumption for practical applications, as often it might vary substantially in x. Still, even those conservative bounds often yield lower bounds that render importance sampling a very challenging endeavor. On the contrary, it seems to be hard to make x-dependent bounds on δ useful due to potentially very complex stochastic dynamics. Let us further note that the bounds in (3.41) imply that errors made over different points in time accumulate, i.e. it does not matter if they have been made at the beginning or the end of a trajectory and neither can they be compensated at later stages.

Another upper bound on the relative error can be derived by means of the Hölder inequality.

Proposition 3.20 (Another bound for path space relative error)**.** *Let $\delta = u^* - u$. For the relative error* (3.26) *it holds*

$$r(u) \leq \left(\mathbb{E}\left[\exp\left((1+\sqrt{2})^2 \int_0^T |\delta(X_s^u, s)|^2 \mathrm{d}s \right) \right]^{\frac{1}{1+\sqrt{2}}} - 1 \right)^{\frac{1}{2}} \tag{3.44}$$

Proof. See Appendix C.2. □

Remark 3.21. Some intuition of the quality of this bound can be gained when for instance assuming that $\delta(x,t) = \varepsilon$ with a constant vector $\varepsilon = (\widetilde{\varepsilon}, \dots, \widetilde{\varepsilon})^\top \in \mathbb{R}^d$. Then this bound yields $r(u) \leq \left(\exp\left((1+\sqrt{2})d\widetilde{\varepsilon}^2 T\right) - 1\right)^{\frac{1}{2}}$, which is less tight than the bound (3.43) in Corollary 3.18. Nevertheless the bound (3.44) is useful in that it only depends on the stochastic process controlled by u, which is a known quantity.

3.2.1 PDE methods for the study of relative errors

Another means of studying the relativ error $r(u)$ are partial differential equations (PDEs). We will formulate a PDE for the relative error (3.2), which might be helpful for future analysis and by which we can rederive bounds from the previous section.

By a slight generalization of [274], one can identify a PDE for the u-dependent second moment (conditioned on $X_t^u = x$),

$$M_u(x,t) = \mathbb{E}\left[e^{-2\,\mathcal{W}(X^u)} \left(\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^u}(X^u) \right)^2 \middle| X_t^u = x \right], \tag{3.45}$$

namely

$$(\partial_t + L - \sigma u(x,t)\cdot\nabla - 2f(x,t) + |u(x,t)|^2) M_u(x,t) = 0, \qquad (x,t) \in \mathbb{R}^d \times [0,T), \tag{3.46a}$$

$$M_u(x,T) = e^{-2g(x)}, \qquad x \in \mathbb{R}^d, \tag{3.46b}$$

where $L = \frac{1}{2}(\sigma\sigma^\top)(x,t) : \nabla^2 + b(x,t)\cdot\nabla$ is the infinitesimal generator associated to the SDE (3.25).

Defining $\delta = u^* - u$, this then immediately leads to the PDE

$$(\partial_t + L + \sigma(\sigma^\top \nabla V(x,t) + \delta(x,t))\cdot\nabla - 2f(x,t) + |\sigma^\top \nabla V(x,t) + \delta(x,t)|^2) M_u(x,t) = 0, \ (x,t) \in \mathbb{R}^d \times [0,T), \tag{3.47a}$$

$$M_u(x,T) = e^{-2g(x)}, \qquad x \in \mathbb{R}^d, \tag{3.47b}$$

which describes the second moment of suboptimal importance sampling. It can be shown that for $\delta = 0$, i.e. under the optimal control $u = u^*$, we recover indeed the zero-variance property of the corresponding importance sampling estimator, see Proposition B.6 in the appendix. In the following statement we construct the PDE that is relevant for the relative error $r(u)$ and re-derive a formula that we have already seen before.

Proposition 3.22 (PDE for the relative error)**.** *Let $\delta = u^* - u$. We consider the second moment as in* (3.45) *and the conditional expectation $\psi(x,t) = \mathbb{E}\left[e^{-\mathcal{W}(X^u)} \middle| X_t = x \right]$, then the function $h_u : \mathbb{R}^d \times [0,T] \to \mathbb{R}$ defined by*

$$h_u(x,t) = \frac{M_u(x,t)}{\psi^2(x,t)}, \tag{3.48}$$

(and related to the relative error by $r(u) = \sqrt{h_u(x,0) - 1}$) solves the PDE

$$\left(\partial_t + L^{u+2\delta} + |\delta(x,t)|^2\right) h_u(x,t) = 0, \qquad (x,t) \in \mathbb{R}^d \times [0,T), \tag{3.49a}$$

$$h_u(x,T) = 1, \qquad x \in \mathbb{R}^d, \tag{3.49b}$$

with $L^{u+2\delta} := L + \sigma(u+2\delta)\cdot\nabla$. This then implies

$$h_u(x,t) = \mathbb{E}\left[\exp\left(\int_t^T |\delta(X_s^{u+2\delta}, s)|^2 \mathrm{d}s \right) \middle| X_t^{u+2\delta} = x \right]. \tag{3.50}$$

Proof. We plug the ansatz

$$M_u(x,t) = h_u(x,t)\psi^2(x,t) = h_u(x,t)e^{-2V(x,t)} \tag{3.51}$$

into the PDE (3.47a). Noting that

$$(\sigma\sigma^\top) : \nabla^2(h_u e^{-2V}) = (\sigma\sigma^\top) : \left(\nabla\left(\nabla h_u e^{-2V} - 2h_u \nabla V e^{-2V}\right)\right) \tag{3.52a}$$

$$= e^{-2V}\left((\sigma\sigma^\top) : \nabla^2 h_u - 4\,\sigma\sigma^\top\nabla V\cdot\nabla h_u + 4\,h_u|\sigma^\top\nabla V|^2 - 2h_u(\sigma\sigma^\top):\nabla^2 V\right), \tag{3.52b}$$

we get the PDE

$$-2h_u \underbrace{\left(\partial_t V + LV - \frac{1}{2}|\sigma^\top\nabla V|^2 + f\right)}_{=0} + \partial_t h_u + Lh_u - \sigma\sigma^\top\nabla V\cdot\nabla h_u + \sigma\delta\cdot\nabla h_u + |\delta|^2 h_u = 0, \tag{3.53}$$

from which the statement follows from the identity $u^* = -\sigma^\top\nabla V$ and a specific Hamilton-Jacobi-Bellman equation that is for instance stated in Problem 1.4. The probabilistic representation (3.50) follows immediately from the Feynman-Kac formula, see Theorem 2.14. □

Remark 3.23. Note that h_u from Proposition 3.22 is related to the relative error (3.28) via $r(u) = \sqrt{h_u(x,0) - 1}$. On the first glance it looks like the PDE (3.49) does not depend on f and g. This is of course not true and we should note that the PDE depends on u^*, which again depends on f and g. Finally, note that with (3.50) we recover the result (3.36) from Proposition 3.15.

3.2.2 Small noise diffusions

A prominent application of importance sampling in stochastic processes can be found in the context of small noise diffusions and rare event simulations (relating to Example 3.13, see also [79, 273, 274, 290] and Section 2.3.3). We model small noises with the smallness parameter $\eta > 0$ by considering the SDEs[21]

$$\mathrm{d}X_s^\eta = b(X_s^\eta, s)\,\mathrm{d}s + \sqrt{\eta}\,\widetilde{\sigma}(X_s^\eta, s)\,\mathrm{d}W_s, \qquad X_t^\eta = x_{\mathrm{init}}, \tag{3.54}$$

and we want to compute quantities like

$$\psi^\eta(x,t) = \mathbb{E}\left[e^{-\frac{1}{\eta}\mathcal{W}(X^\eta)}\middle| X_t^\eta = x\right]. \tag{3.55}$$

If η gets smaller it becomes harder to estimate $\psi^\eta(x,t)$ via Monte Carlo methods as the variance grows exponentially in η. To be more precise (and as already stated in Remark 2.42), by Varadhan's lemma [68, Theorem 4.3.1], using the quantities

$$\gamma_1 := -\lim_{\eta\to 0}\eta\log\mathbb{E}\left[e^{-\frac{1}{\eta}\mathcal{W}(X^\eta)}\right] \qquad \text{and} \qquad \gamma_2 := -\lim_{\eta\to 0}\eta\log\mathbb{E}\left[e^{-\frac{2}{\eta}\mathcal{W}(X^\eta)}\right], \tag{3.56}$$

one gets for the relative error of the uncontrolled process

$$r(0) = \sqrt{e^{\frac{2\gamma_1-\gamma_2+o(1)}{\eta}} - 1}, \tag{3.57}$$

asymptotically as $\eta\to 0$. By Jensen's inequality we have $2\gamma_1 > \gamma_2$ unless $\mathcal{W}$ is a.s. constant, but we note that even for $2\gamma_1 = \gamma_2$ the relative error explodes in the limit $\eta\to 0$. Let us again consider a controlled process

$$\mathrm{d}X_s^{u,\eta} = \left(b(X_s^{u,\eta},s) + \widetilde{\sigma}(X_s^{u,\eta},s)u(X_s^{u,\eta},s)\right)\mathrm{d}s + \sqrt{\eta}\,\widetilde{\sigma}(X_s^{u,\eta},s)\,\mathrm{d}W_s, \qquad X_t^{u,\eta} = x_{\mathrm{init}}, \tag{3.58}$$

and realize that the optimal importance sampling control that yields zero variance,

$$u^* = -\widetilde{\sigma}^\top\nabla V^\eta = \eta\,\widetilde{\sigma}^\top\nabla\log\psi^\eta, \tag{3.59}$$

can be computed via the HJB equation

$$\left(\partial_t + \frac{\eta}{2}(\widetilde{\sigma}\widetilde{\sigma}^\top)(x,t) : \nabla^2 + b(x,t)\cdot\nabla\right)V^\eta(x,t) - \frac{1}{2}|(\widetilde{\sigma}^\top\nabla V^\eta)(x,t)|^2 + f(x,t) = 0, \qquad V^\eta(x,T) = g(x). \tag{3.60}$$

Since solving this PDE is notoriously difficult (especially in high dimensions), various approximations have been suggested that lead to estimators that enjoy log-efficiency or a vanishing relative error in the regime of

[21] To be consistent with the notation from before, we could hide the smallness parameter η in the diffusion coefficient, i.e. $\sigma = \sqrt{\eta}\widetilde{\sigma}$. Then the HJB equation that provides the zero variance control is $(\partial_t + \frac{\eta}{2}(\widetilde{\sigma}\widetilde{\sigma}^\top) : \nabla^2 + b\cdot\nabla)V - \frac{1}{2}|\sqrt{\eta}\widetilde{\sigma}\nabla V|^2 + \frac{1}{\eta}f(x,t) = 0, V(x,T) = \frac{1}{\eta}g(x)$ and the relation $V^\eta = \eta V$ yields HJB equation (3.60).

a vanishing η. However, since log-efficient estimators still often perform badly in practice (as for instances discussed in [6, 108]), in [290] it is suggested to replace u^* by the vanishing viscosity approximation u^0 based on the corresponding HJB equation with $\eta = 0$:

$$u^0 = -\widetilde{\sigma}^\top \nabla V^0, \tag{3.61}$$

where V^0 is the solution to

$$(\partial_t + b(x,t)\cdot\nabla)V^0(x,t) - \frac{1}{2}|(\widetilde{\sigma}^\top \nabla V^0)(x,t)|^2 + f(x,t) = 0, \qquad V^0(x,T) = g(x). \tag{3.62}$$

While it can be shown that, given some regularity assumptions on f and g, it holds [290]

$$\lim_{\eta\to 0} r(u^0) = 0, \tag{3.63}$$

a large relative error for a small, but fixed $\eta > 0$ is still possible. In our notation from before, this situation corresponds to choosing $\delta = u^* - u^0$ and Propositions 3.15 and 3.22 show that

$$r(u^0) = \sqrt{\mathbb{E}\left[\exp\left(\int_0^T |u^* - u^0|^2(X_s^{2u^*-u^0}, s)\mathrm{d}s\right)\right] - 1}. \tag{3.64}$$

Even though this expression converges to zero as $\eta \to 0$ provided that $V \to V^0$ and $u^* \to u^0$ [96], we expect an exponential dependence on the time T and the dimension d for any fixed $\eta > 0$ (cf. our numerical experiment in Section 3.3.4).

In [96] it is proved that

$$\nabla V = \nabla V^0 + \eta \nabla v_1 + o(\eta), \tag{3.65}$$

uniformly on all compact subsets of $\mathbb{R}^d \times (0,T)$, where v_1 solves the PDE stated in Appendix B.4.2. As a consequence, we can write

$$|\nabla V - \nabla V^0| = |\eta \nabla v_1 + o(\eta)| = \eta|\nabla v_1 + o(1)| \tag{3.66}$$

and

$$r(u^0) = \mathbb{E}\left[\exp\left(\eta^2 \int_0^T |(\sigma^\top \nabla v_1)(X_s^{2u^*-u_0^*}, s)|^2 \mathrm{d}s + o(\eta^2)\right)\right]. \tag{3.67}$$

Specifically, if there exist constants $C_1, C_2 > 0$ such that $C_1 < |\nabla v_1(x,t)| < C_2$ for all $(x,t) \in \mathbb{R}^d \times (0,T)$, then the relative error grows exponentially as

$$\sqrt{e^{\eta^2 C_1^2 T + o(\eta^2)} - 1} \le r(u^*) \le \sqrt{e^{\eta^2 C_2^2 T + o(\eta^2)} - 1} \tag{3.68}$$

due to Corollary 3.18. We emphasize, however, that it is not clear under which assumptions this uniform bound can be achieved, given that, in practice, v_1 can be strongly x-dependent as is illustrated with a numerical example in Section 3.3.4.

Remark 3.24. The above considerations show that the relative error is potentially only small if η is (much) smaller than $C_1\sqrt{T}$. This can be compared to equation (5.3) in [274] and in particular to [80], where a concrete example is constructed for which the second moment can be lower bounded by $e^{-\frac{1}{\eta}C_1 + (T-K)C_2}$ for $C_1, C_2, K > 0$, i.e. the time T and the smallness parameter η compete. We illustrate the degeneracy with growing T for a toy example in Figure 3.7.

3.3 Numerical examples

In this section we provide numerical examples that shall illustrate some of the formulas and bounds derived in the previous sections. We particularly demonstrate that importance sampling can be very sensitive to small perturbations of the optimal proposal measure. Here we focus on path space measures and provide several examples of importance sampling of diffusions. The code can be found at `https://github.com/lorenzrichter/suboptimal-importance-sampling`.

3.3.1 Ornstein-Uhlenbeck process

An example where the optimal importance sampling control is analytically computable is the following. Consider the d-dimensional Ornstein-Uhlenbeck process

$$\mathrm{d}X_s = AX_s\,\mathrm{d}s + B\,\mathrm{d}W_s, \qquad X_0 = 0, \tag{3.69}$$

and its controlled version

$$\mathrm{d}X_s^u = (AX_s^u + Bu(X_s^u, s))\,\mathrm{d}s + B\,\mathrm{d}W_s, \qquad X_0^u = 0, \tag{3.70}$$

where $A, B \in \mathbb{R}^{d\times d}$ are given matrices. In (2.102) we set $f = 0$ and $g(x) = \alpha \cdot x$, for a fixed vector $\alpha \in \mathbb{R}^d$, i.e. we want to estimate the quantity

$$\mathcal{Z} = \mathbb{E}\left[e^{-\alpha\cdot X_T}\right]. \tag{3.71}$$

As shown in Appendix B.5.1, the zero-variance importance sampling control is given by

$$u^*(x,t) = -B^\top e^{A^\top(T-t)}\alpha. \tag{3.72}$$

We choose $A = -3\,\mathrm{Id}_{d\times d} + (\xi_{ij})_{1\le i,j\le d}$ and $B = \mathrm{Id}_{d\times d} + (\xi_{ij})_{1\le i,j\le d}$, where $\xi_{ij} \sim \mathcal{N}(0,\sigma^2)$ are i.i.d. random coefficients that are held fixed throughout the simulation. We set $T = 1, \sigma = 1$, $\alpha = (1,\dots,1)^\top$ and first consider the perturbed control

$$u = u^* + (\varepsilon,\dots,\varepsilon)^\top. \tag{3.73}$$

In the two left panels of Figure 3.2 we display a Monte Carlo estimation of the relative error (3.28) using $K = 10^6$ samples and compare it to the formulas from Corollary 3.18 and the bound from Corollary 3.14, once with varying perturbation strength ε, once with varying dimension d. We see that in both cases the simulations agree with our formula, even though for moderate to large deviations from optimality the estimated values of r are observed to fluctuate.

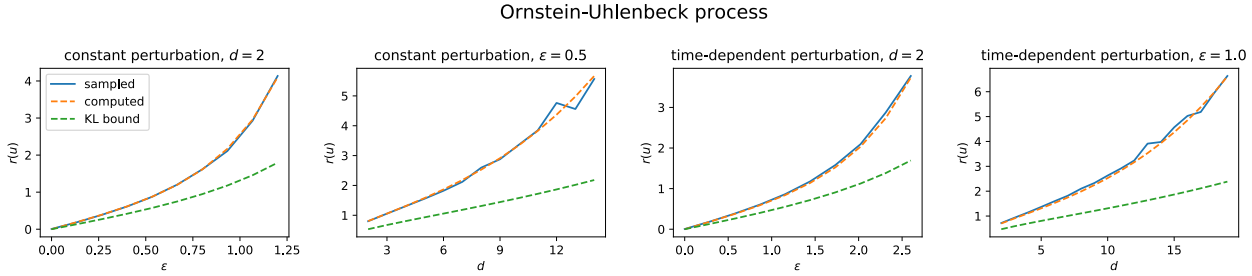


Figure 3.2: Sampled relative error with varying constant or time-dependent perturbation ε and dimension d compared to the formulas derived in Corollary 3.18 and to the lower bound from Corollary 3.14.

Let us now look at an example with a time-dependent perturbation of the optimal control. More specifically, we consider a perturbation that is active only for a certain amount of time $s < T$, namely

$$u(x,t) = u^*(x,t) + (\varepsilon,\dots,\varepsilon)^\top \mathbb{1}_{[0,s]}(t), \tag{3.74}$$

where in our experiment we choose $s = 0.2$. In the two right panels of Figure 3.2 we display the same comparisons as before, however now using formula (3.41) in order to account for the time-dependent nature of the perturbation.

3.3.2 Double well potential

For strongly metastable systems, Monte Carlo estimation is notoriously difficult and variance reduction methods are often indispensable. Importance sampling seems like a method of choice, but we want to illustrate that one has to be very careful with the design of the importance sampling control.

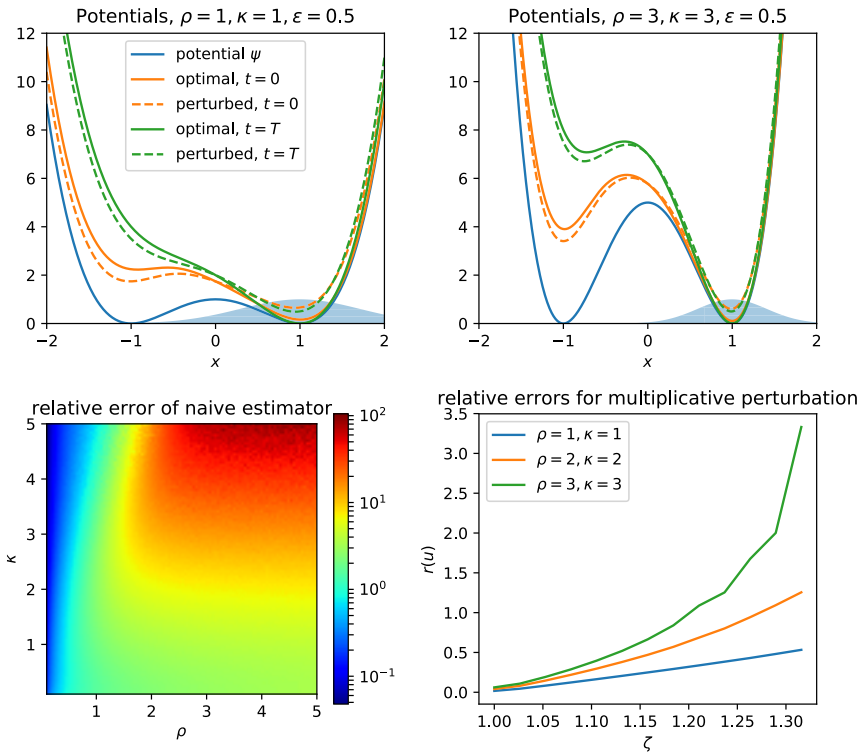


Figure 3.3: Top panels: Double well potentials and optimal tiltings as well as additive perturbations for different values of ρ and κ. Bottom left: Relative error of the naive Monte Carlo estimator for different values of ρ and κ. Bottom right: Relative error depending on the multiplicative perturbation factor ζ.

As in Example 3.13, let us consider the Langevin SDE

$$\mathrm{d}X_s = -\nabla\Psi(X_s)\,\mathrm{d}s + B\,\mathrm{d}W_s, \qquad X_0 = x_{\text{init}}, \tag{3.75}$$

in $d = 1$, where $B \in \mathbb{R}$ is the diffusion coefficient, $\Psi(x) = \kappa(x^2 - 1)^2$ is a double well potential with $\kappa > 0$ and $x = -1$ is the initial condition. For the observable in (2.102) we consider $f = 0$ and $g(x) = \rho(x - 1)^2$, where $\rho > 0$; the terminal time is set to $T = 1$. Note that choosing higher values for ρ and κ accentuates the metastable features, making sample-based estimation of $\mathbb{E}\left[\exp(-g(X_T))\right]$ more challenging. For an illustration, the two top panels of Figure 3.3 show the potential Ψ and the weight from (3.29), $e^{-g(x)}$, for different values of ρ and κ and for $B = 1$. We also plot the 'optimally tilted potentials' $\Psi^* = \Psi + BB^\top V$, noting that $-\nabla\Psi^* = -\nabla\Psi + Bu^*$. In the bottom left panel we show the relative error of the naive estimator depending on different values of κ and ρ.

As before, let us perturb the optimal control, this time both in an additive and multiplicative way, namely

$$u = u^* + \varepsilon = -B^\top\nabla(V - B^{-\top}\varepsilon \cdot x) \qquad \text{and} \qquad u = \zeta u^*, \tag{3.76}$$

where $\varepsilon \in \mathbb{R}^d, \zeta \in \mathbb{R}$ specify the perturbation strengths. In the bottom right panel of Figure 3.3 we show the relative error for the multiplicative perturbation and see that for higher values of ρ and κ the exponential divergence becomes more severe, demonstrating that the robustness issues of importance sampling are particularly present in metastable settings.

Let us now consider perturbations depending either on time or space,

$$u_1(x,t) = u^*(x,t) + \varepsilon\sin(\alpha t) \qquad \text{and} \qquad u_2(x,t) = u^*(x,t) + \varepsilon\sin(\alpha x), \tag{3.77}$$

as illustrated in Figure 3.4 with $\alpha = 50$.

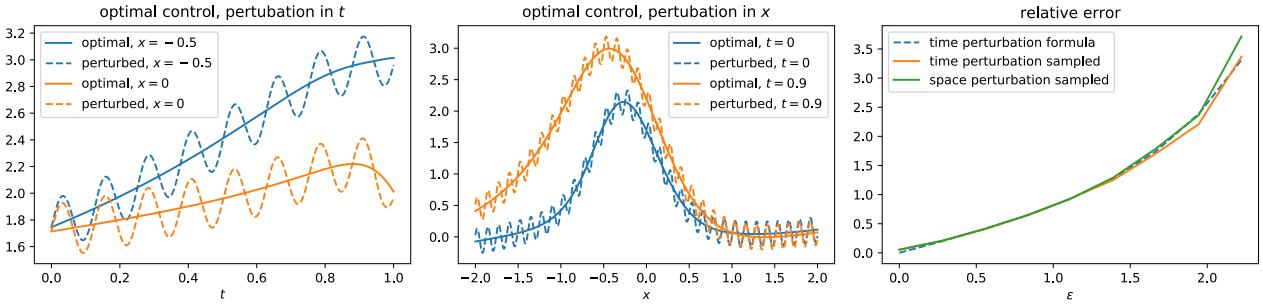


Figure 3.4: Left: Optimal importance sampling control and time perturbation for two different values of x. Middle: Optimal importance sampling control and space perturbation for two different values of t. Right: Relative error of suboptimal importance sampling estimators depending on the perturbation strength ε; here, the dashed line refers to the exact formula (3.78).

In the former case we can analytically compute the relative error due to Corollary 3.18 to be

$$r_1(\varepsilon) = \sqrt{\exp\left(\varepsilon^2\left(\frac{T}{2} - \frac{\sin(2\alpha T)}{4\alpha}\right)\right) - 1}. \tag{3.78}$$

Let us again illustrate how the relative error depends on the perturbation strength ε. In the right panel of Figure 3.4 we can see the agreement of the sampled version with formula (3.78) when considering the time-dependent perturbation. We do not have a formula in the case of a space-dependent perturbation, however we can still observe the exponential dependence on the perturbation strength in the estimated relative error, which is expected for instance from formulas (3.34) and (3.35).

3.3.3 Random stopping times

The suboptimal importance sampling bounds from Section 3.2 can be transferred to problems that involve a random stopping time τ rather than a fixed time horizon T, where mostly $\tau^u = \inf\{t > 0 : X_t^u \notin \mathcal{D}\}$ is defined[22] as the first exit time of a bounded domain $\mathcal{D} \subset \mathbb{R}^d$. However, one has to be careful with applying our formulas and bounds from above, as τ^u itself depends on the law of the process. For an illustration, let us consider a one-dimensional toy example, where the dynamics is a scaled Brownian motion

$$X_t = \sqrt{2}\, W_t \tag{3.79}$$

and we choose $f = 1, g = 0$ in (2.102), such that

$$\mathcal{Z} = \mathbb{E}\left[e^{-\tau}\right]. \tag{3.80}$$

By noting that $\psi(x) = \mathbb{E}\left[e^{-\tau} | X_0 = x\right]$ fulfills the boundary value problem

$$(\Delta - 1)\,\psi(x) = 0, \qquad x \in \mathcal{D}, \tag{3.81a}$$
$$\psi(x) = 1, \qquad x \in \partial\mathcal{D}, \tag{3.81b}$$

we can compute the optimal zero-variance importance sampling control to be

$$u^*(x) = \sqrt{2}\,\nabla \log \psi(x) = \sqrt{2}\,\frac{1 - e^{-2x}}{1 + e^{-2x}}. \tag{3.82}$$

In our experiment, we again perturb the optimal control via

$$u = u^* + \varepsilon. \tag{3.83}$$

Formula (3.36) provides an expression for the relative error, even if T is replaced by a random time τ (which we leave the reader to check for herself), namely

$$r(u) = \left(\mathbb{E}\left[e^{\varepsilon^2 \tau^{2u^* - u}}\right] - 1\right)^{\frac{1}{2}} \geq \left(e^{\varepsilon^2\, \mathbb{E}\left[\tau^{2u^* - u}\right]} - 1\right)^{\frac{1}{2}}, \tag{3.84}$$

[22] We denote with $\tau = \tau^0$ the hitting time of the uncontrolled process X_t.

where it is essential that τ^{2u^*-u} refers to the hitting time of the process $X_t^{2u^*-u}$. We applied Jensen's inequality in the last expression and note that naively assuming

$$r(u) \approx \left(e^{\varepsilon^2 \mathbb{E}[\tau]} - 1\right)^{\frac{1}{2}} \tag{3.85}$$

is usually wrong. Figure 3.5 compares the sampled relative error with the exact formula, the lower bound in (3.84) and the wrong expression (3.85).

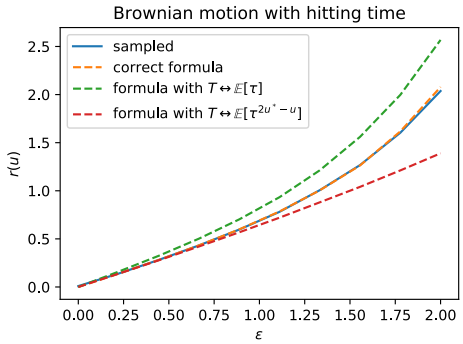


Figure 3.5: Relative error of a quantity involving a random stopping time compared to the exact formula, a lower bound as well as a naive, but usually wrong approximation.

Remark 3.25. Let us note again that estimating quantities involving hitting times gets particularly challenging in rare event settings, where the expected hitting time might become very large, cf. Example 3.13. The relation (3.84) for the relative error then indicates that Monte Carlo estimation becomes especially difficult.

3.3.4 Small noise diffusions

As an example for a small noise diffusion, we consider a modification of a one-dimensional toy example that has been proposed in [290]. We take the scaled Brownian motion

$$X_s^\eta = \sqrt{\eta} W_s, \qquad X_0 = 0.1, \tag{3.86}$$

and want to compute

$$\mathbb{E}\left[e^{-\frac{1}{\eta} g(X_T^\eta)}\right] \tag{3.87}$$

with

$$g(x) = \frac{\alpha}{2}\left(1 - \frac{|x|}{\sqrt{\alpha}}\right)^2 \tag{3.88}$$

for $\alpha > 0$. One readily sees that

$$V^0(x,t) = \frac{\alpha\left(1 - \frac{|x|}{\sqrt{\alpha}}\right)^2}{2(T-t+1)} \tag{3.89}$$

is the unique viscosity solution to the deterministic problem (3.62); we refer to [98] for a discussion of the theory of viscosity solutions. Since an explicit solution $V^*(x,t)$ to the second-order HJB equation (3.60) is not available, we approximate it with finite differences. In Figure 3.6 we show the corresponding controls $u^0(x,s) = -\sigma^\top \nabla V^0(x,t)$ and $u^*(x,s) = -\sigma^\top \nabla V^*(x,t)$ for different values of the noise coefficient η.

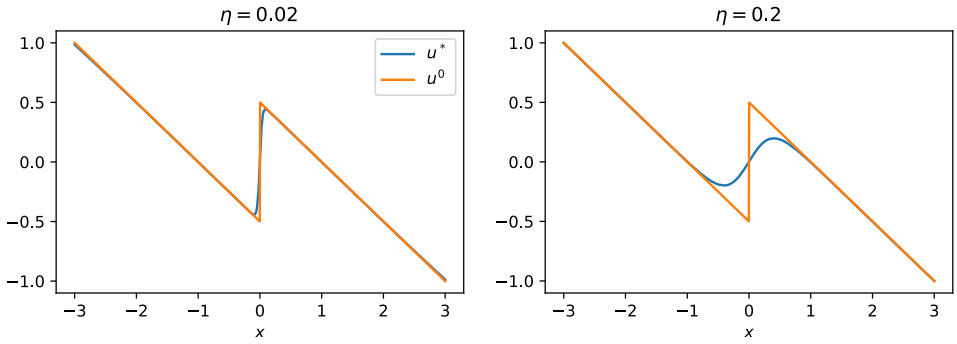


Figure 3.6: For a small noise diffusion problem we display once the optimal control and once the control resulting from the zero-noise approximation with different noise scalings η.

In the middle panel of Figure 3.7 we show the relative error depending on the noise parameter η. Unlike one could expect from (3.68), it seems to not grow exponentially in η, which can be explained by looking at the exponentiated L^2 error, $\exp\left(\mathbb{E}\left[\int_0^T |u^* - u^0|^2(X_s^{u^0}, s)\mathrm{d}s\right]\right)$, which we plot in the left panel. The observation that this does not grow exponentially seems to be rooted in the fact that the suboptimality $\delta = u^* - u^0$ is very different for different values of x. If we vary T, however, we can observe an exponential dependency on the time horizon, as displayed in the right panel of Figure 3.7, again being in accordance with the consideration in Section 3.2.2.

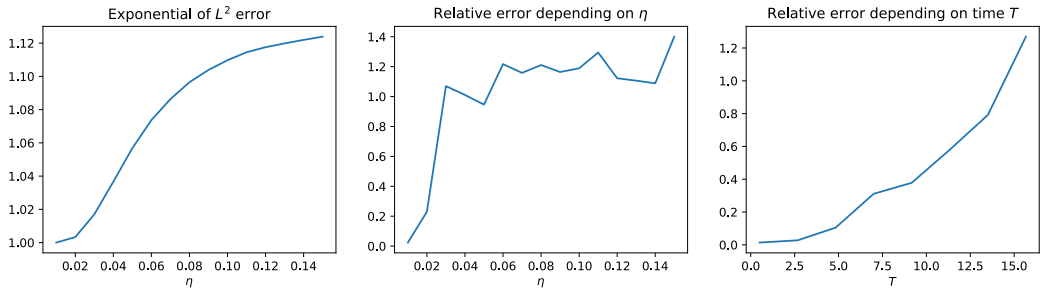


Figure 3.7: Small noise diffusions with vanishing noise coefficient η. Left: Exponential of L^2 error between u^* and u^0 depending on η for $T = 1$. Middle: Relative importance sampling error depending on η. Right: Relative importance sampling error depending on T for $\eta = 0.005$.

Chapter 4

Approximating probability measures on path space

In this chapter we aim at designing and analyzing robust algorithms for numerically solving either of the Problems 1.1-1.5 that we have formulated in Chapter 1. One common feature is that they can be formulated as variational problems that, on a formal level, can be recast as approximation tasks for probability measures on the space of trajectories, i.e. "path space", where a central object is the controlled diffusion

$$\mathrm{d}X_s^u = (b(X_s^u, s) + \sigma(X_s^u, s)u(X_s^u, s))\,\mathrm{d}s + \sigma(X_s^u, s)\,\mathrm{d}W_s, \qquad X_0^u = x_{\mathrm{init}}, \tag{4.1}$$

that we have already specified in (1.4). In order to solve either of our problems we will build on recent machine learning inspired approaches and investigate a class of algorithms that may be termed *iterative diffusion optimization* (IDO) techniques, related, in spirit, to reinforcement learning [238]. Speaking in broad terms, those are characterized by the following outline of steps meant to be executed iteratively until convergence or until a satisfactory control u is found:

1. Simulate K realizations $\{(X_s^{u,(k)})_{0\le s\le T},\ k = 1, \dots, K\}$ of the solution to (4.1).
2. Compute a performance measure and a corresponding gradient associated to the control u, based on $\{(X_s^{u,(k)})_{0\le s\le T},\ k = 1, \dots, K\}$.
3. Modify u according to the gradient obtained in the previous step. Repeat starting from 1.

We note that many algorithmic approaches from the literature can be placed in the IDO framework, in particular some that connect forward-backward SDEs and machine learning [86, 122] as well as some that are rooted in molecular dynamics and optimal control [131, 160, 308]. Crucially, those instances of IDO mainly differ in terms of the performance measure employed in step 2, or, in other words, in terms of an underlying loss function $\mathcal{L}(u)$ constructed on the set of control vector fields. Typically, $\mathcal{L}(u)$ is given in terms of expectations involving the solution to (4.1). Consequently, step 1 can be thought of as providing an empirical estimate of this quantity (and its gradient) based on a sample of size K.

For a principled design and understanding of IDO-like algorithms, it is central to analyze the properties of loss functions and corresponding Monte Carlo estimators, and identify guidelines that promise good performance. Permissible loss functions include those that admit a global minimum representing the solution to the problem at hand. Moreover, suitable loss functions yield themselves to efficient optimization procedures (step 3) such as stochastic gradient descent. In this respect, important desiderata are the absence of local minima as well as the availability of low-variance gradient estimators.

In this chapter, we show that a variety of loss functions can be constructed and analyzed in terms of divergences between probability measures on the path space associated to solutions of (4.1), providing a unifying framework for IDO and extending on previous works in that direction [131, 160, 308]. As this perspective entails the approximation of a target probability measure as a core element, our approach exposes connections to the theory of variational inference [36, 305] (see also Chapter 5). Classical divergences include the relative entropy (or KL-divergence) and its counterpart, the cross-entropy. Motivated by connections to forward-backward SDEs and importance sampling, we propose the novel family of *log-variance* divergences,

$$D_{\widetilde{\mathbb{P}}}^{\mathrm{Var(log)}}(\mathbb{P}_1|\mathbb{P}_2) = \mathrm{Var}_{\widetilde{\mathbb{P}}}\left(\log \frac{\mathrm{d}\mathbb{P}_2}{\mathrm{d}\mathbb{P}_1}\right), \tag{4.2}$$

parametrized by a probability measure $\widetilde{\mathbb{P}}$. Loss functions based on these divergences can be viewed as modifications of those proposed in [86, 122] for solving forward-backward SDEs, essentially replacing second moments by variances, see Section 4.1.2. Moreover, it turns out that the log-variance divergences are closely related to the KL-divergence (see Proposition 4.22), allowing us to draw (perhaps surprising) connections to methods that directly attempt to optimize the dynamics with respect to a control objective.

As the loss functions considered in this section are defined in terms of expected values, practical implementations require appropriate Monte Carlo estimators whose variance directly impacts algorithmic performance. We study the associated relative errors, in particular in high-dimensional settings and for $\mathbb{P}_1 \approx \mathbb{P}_2$, i.e. close to the optimal control. The proposed log-variance divergence and its corresponding standard Monte Carlo estimator turn out to be robust in both settings, in a precise sense that will be developed later.

Let us recall Section 1.5, where we have provided a literature overview related to the connections between different perspectives on the control problem under consideration, in particular summarizing corresponding numerical treatments. As a unifying viewpoint, in Section 4.1 we define viable loss functions through divergences on path space and discuss their connections to the algorithmic approaches encountered in Section 1.5. In particular, we elucidate the relationships of the log-variance divergences with forward-backward SDEs. In the two upcoming sections we analyze properties of the suggested losses, where in Section 4.2 we obtain equivalence relations that hold in an infinite batch size limit and in Section 4.3 we investigate the variances associated to the losses' estimator versions. In the latter case, we consider stability close to the optimal control solution as well as in high dimensional settings. In Section 4.4 we provide numerical examples that demonstrate our findings and in Appendix B.6 we present some further illustrations.

Parts of this chapter are based on joint work with Nikolas Nüsken and have been published in [217].

4.1 Iterative diffusion optimization

In this section we demonstrate that many of the algorithmic approaches encountered in Section 1.5 can be recovered as minimization procedures of certain divergences between probability measures on path space. Similar perspectives (mostly discussing the relative entropy and cross-entropy in Definition 4.1 below) can be found in the literature, see [131, 160, 308]. Recall from Section 1.3 that we denote by $\mathcal{C}$ the space of $\mathbb{R}^d$-valued paths on the time interval $[0,T]$ with fixed initial point $x_{\mathrm{init}} \in \mathbb{R}^d$. As before, the probability measures on $\mathcal{C}$ induced by the stochastic process

$$\mathrm{d}X_s = b(X_s,s)\,\mathrm{d}s + \sigma(X_s,s)\,\mathrm{d}W_s, \qquad X_t = x_{\mathrm{init}}, \tag{4.3}$$

as in (1.2) and its controlled counterpart

$$\mathrm{d}X^u_s = (b(X^u_s,s) + \sigma(X^u_s,s)u(X^u_s,s))\,\mathrm{d}s + \sigma(X^u_s,s)\,\mathrm{d}W_s, \qquad X^u_t = x_{\mathrm{init}}, \tag{4.4}$$

as in (1.4) will be denoted by $\mathbb{P}$ and $\mathbb{P}^u$, respectively. From now on, let us assume that there exists a unique optimal control with convenient regularity properties:

Assumption 4. *The HJB PDE* (1.20) *admits a unique solution* $V \in C_b^{2,1}(\mathbb{R}^d \times [0,T])$. *We set*

$$u^* = -\sigma^\top \nabla V. \tag{4.5}$$

For Assumption 4 to be satisfied, it is sufficient to impose the regularity and boundedness conditions $b, \sigma, f \in C_b^{2,1}(\mathbb{R}^d)$ and $g \in C_b^3(\mathbb{R}^d)$, see[23] [98, Theorem 4.2]. The strong boundedness assumption on V could be weakened and for instance be replaced by the condition $\sigma^\top \nabla V \in \mathcal{U}$. For existence and uniqueness results involving unbounded controls we refer to [99], and for specific examples to Sections 4.4.2 and 4.4.3. In the sense made precise in Theorem 1.2, the control u^* defined above provides solutions to the Problems 1.1-1.5 considered in Chapter 1. Moreover, there exists a corresponding optimal path measure $\mathbb{Q}$ (in the following also called the *target measure*) defined in (1.13) and satisfying $\mathbb{Q} = \mathbb{P}^{u^*}$. We further note that Assumption 4 together with the results from [281, Chapter 11] imply that the solution to the FBSDE (1.21) is unique.

4.1.1 Divergences and loss functions

The SDE (4.4) establishes a measurable map $\mathcal{U} \ni u \mapsto \mathbb{P}^u \in \mathcal{P}(\mathcal{C})$ that can be made explicit in terms of Radon-Nikodym derivatives using Girsanov's theorem (see Theorem B.3 in the Appendix). Consequently, we can elevate

[23] This result requires the boundedness of the controls in $\mathcal{U}$. However, applying [181, Chapter II, Theorem 3.1] to (1.27), we see that ∇V is bounded and hence $\mathcal{U}$ can be restricted appropriately.

divergences between path measures to loss functions on vector fields. To wit, let $D : \mathcal{P}(\mathcal{C}) \times \mathcal{P}(\mathcal{C}) \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ be a divergence[24], where, as before, $\mathcal{P}(\mathcal{C})$ denotes the set of probability measures on $\mathcal{C}$. Then, setting

$$\mathcal{L}_D(u) = D(\mathbb{P}^u|\mathbb{Q}), \qquad u \in \mathcal{U}, \tag{4.6}$$

we immediately see that $\mathcal{L}_D \geq 0$, with Theorem 1.2 implying that $\mathcal{L}_D(u) = 0$ if and only if $u = u^*$. Consequently, an approximation of the optimal control vector field u^* can in principle be found by minimizing the loss $\mathcal{L}_D$. In the remainder of this section, we will suggest possible losses and study some of their properties.

Starting with the KL-divergence as defined in Definition 3.1, we introduce the *relative entropy loss* and the *cross-entropy loss*, corresponding to the divergences

$$D^{\mathrm{RE}}(\mathbb{P}_1|\mathbb{P}_2) = \mathrm{KL}(\mathbb{P}_1|\mathbb{P}_2) \qquad \text{and} \qquad D^{\mathrm{CE}}(\mathbb{P}_1|\mathbb{P}_2) = \mathrm{KL}(\mathbb{P}_2|\mathbb{P}_1). \tag{4.7}$$

Definition 4.1 (Relative entropy and cross-entropy losses)**.** The *relative entropy loss* is given by

$$\mathcal{L}_{\mathrm{RE}}(u) = \mathbb{E}_{\mathbb{P}^u}\left[\log \frac{\mathrm{d}\mathbb{P}^u}{\mathrm{d}\mathbb{Q}}\right], \qquad u \in \mathcal{U}, \tag{4.8}$$

and the *cross-entropy loss* by

$$\mathcal{L}_{\mathrm{CE}}(u) = \mathbb{E}_{\mathbb{Q}}\left[\log \frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}^u}\right], \qquad u \in \mathcal{U}, \tag{4.9}$$

where the target measure $\mathbb{Q}$ has been defined in (1.13).

Remark 4.2 (Notation). Note that, by definition, the expectations in (4.8) and (4.9) are understood as integrals on $\mathcal{C}$, i.e.

$$\mathcal{L}_{\mathrm{RE}}(u) = \int_{\mathcal{C}} \left(\log \frac{\mathrm{d}\mathbb{P}^u}{\mathrm{d}\mathbb{Q}}\right) \mathrm{d}\mathbb{P}^u, \qquad \mathcal{L}_{\mathrm{CE}}(u) = \int_{\mathcal{C}} \left(\log \frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}^u}\right) \mathrm{d}\mathbb{Q}. \tag{4.10}$$

In contrast, the expectation operator $\mathbb{E}$ (without subscript, as used in (1.10) and (1.16), for instance) throughout denotes integrals on the underlying abstract probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0}, \Lambda)$, see also footnote [19] on page 50.

We can further define the variance and log-variance divergences.

Definition 4.3 (Variance divergence)**.** For $\widetilde{\mathbb{P}}, \mathbb{P}_1, \mathbb{P}_2 \in \mathcal{P}(\mathcal{C})$ we define the family of variance divergences parametrized by $\widetilde{\mathbb{P}}$ as

$$D^{\mathrm{Var}}_{\widetilde{\mathbb{P}}}(\mathbb{P}_1|\mathbb{P}_2) = \begin{cases} \mathrm{Var}_{\widetilde{\mathbb{P}}}\left(\frac{\mathrm{d}\mathbb{P}_2}{\mathrm{d}\mathbb{P}_1}\right), & \text{if } \mathbb{P}_1 \sim \mathbb{P}_2 \quad \text{and} \quad \mathbb{E}_{\widetilde{\mathbb{P}}}\left[\left|\frac{\mathrm{d}\mathbb{P}_2}{\mathrm{d}\mathbb{P}_1}\right|\right] < \infty, \\ \infty, & \text{otherwise.} \end{cases} \tag{4.11}$$

Definition 4.4 (Log-variance divergence)**.** For $\widetilde{\mathbb{P}}, \mathbb{P}_1, \mathbb{P}_2 \in \mathcal{P}(\mathcal{C})$ we define the family of variance divergences parametrized by $\widetilde{\mathbb{P}}$ as

$$D^{\mathrm{Var(log)}}_{\widetilde{\mathbb{P}}}(\mathbb{P}_1|\mathbb{P}_2) = \begin{cases} \mathrm{Var}_{\widetilde{\mathbb{P}}}\left(\log \frac{\mathrm{d}\mathbb{P}_2}{\mathrm{d}\mathbb{P}_1}\right), & \text{if } \mathbb{P}_1 \sim \mathbb{P}_2 \quad \text{and} \quad \mathbb{E}_{\widetilde{\mathbb{P}}}\left[\left|\log \frac{\mathrm{d}\mathbb{P}_2}{\mathrm{d}\mathbb{P}_1}\right|\right] < \infty, \\ \infty, & \text{otherwise.} \end{cases} \tag{4.12}$$

For any $\widetilde{\mathbb{P}} \in \mathcal{P}(\mathcal{C})$, it is straightforward to verify that both the variance and the log-variance divergence define indeed divergences on the set of probability measures equivalent to $\widetilde{\mathbb{P}}$.

Remark 4.5. Setting $\widetilde{\mathbb{P}} = \mathbb{P}_1$, the quantity $D^{\mathrm{Var}}_{\mathbb{P}_1}(\mathbb{P}_1|\mathbb{P}_2)$ coincides with the Pearson χ^2-divergence as defined in Definition 3.2 [70, 196] measuring the importance sampling relative error [4], hence relating to Problem 1.1 (see also Lemma 3.3). The divergence $D^{\mathrm{Var(log)}}_{\widetilde{\mathbb{P}}}$ seems to be new; it is motivated by its connections to the forward-backward SDE formulation of optimal control (see Problem 1.5), as will be explained in Section 4.1.2. Let us already mention that inserting the log in (4.11) to obtain (4.12) has the potential benefit of making sample based estimation more robust in high dimensions (see Section 4.3.2). Furthermore, we point the reader to Proposition 4.19 revealing close connections between $D^{\mathrm{Var(log)}}_{\widetilde{\mathbb{P}}}$ and the relative entropy.

Using (4.11) and (4.12) with $\widetilde{\mathbb{P}} = \mathbb{P}^v$, we obtain two additional families of losses, indexed by $v \in \mathcal{U}$:

[24] The defining property of a divergence between probability measures is the equivalence between $D(\mathbb{P}_1|\mathbb{P}_2) = 0$ and $\mathbb{P}_1 = \mathbb{P}_2$. Prominent examples include the KL-divergence and, more generally, the f-divergences [196].

Definition 4.6 (Variance and log-variance losses)**.** For $v \in \mathcal{U}$, the *variance loss* is given by

$$\mathcal{L}_{\mathrm{Var}_v}(u) = \mathrm{Var}_{\mathbb{P}^v}\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}^u}\right), \qquad u \in \mathcal{U}, \tag{4.13}$$

and the *log-variance loss* by

$$\mathcal{L}^{\log}_{\mathrm{Var}_v}(u) = \mathrm{Var}_{\mathbb{P}^v}\left(\log \frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}^u}\right), \qquad u \in \mathcal{U}, \tag{4.14}$$

whenever $\mathbb{E}_{\mathbb{P}^v}\left[\left|\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}^u}\right|\right] < \infty$ or $\mathbb{E}_{\mathbb{P}^v}\left[\left|\log \frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}^u}\right|\right] < \infty$, respectively[25]. The notation $\mathrm{Var}_{\mathbb{P}^v}$ is to be interpreted in line with Remark 4.2.

By direct computations invoking Girsanov's theorem, the losses defined above admit explicit representations in terms of solutions to SDEs of the form (4.3) and (4.4). Crucially, the propositions that follow replace the expectations on $\mathcal{C}$ used in the definitions (4.8), (4.9), (4.11) and (4.12) by expectations on Ω that are more amenable to direct probabilistic interpretation and Monte Carlo simulation (see also Remark 4.2). Recall that the target measure $\mathbb{Q}$ is assumed to be of the type (1.13), where $\mathcal{W}$ has been defined in (1.8). We start with the relative entropy loss:

Proposition 4.7 (Relative entropy loss)**.** *For $u \in \mathcal{U}$, let $(X^u_s)_{0\le s\le T}$ denote the unique strong solution to* (4.4)*. Then*

$$\mathcal{L}_{\mathrm{RE}}(u) = \mathbb{E}\left[\frac{1}{2}\int_0^T |u(X^u_s, s)|^2\,\mathrm{d}s + \int_0^T f(X^u_s, s)\,\mathrm{d}s + g(X^u_T)\right] + \log \mathcal{Z}. \tag{4.15}$$

Proof. See [131, 160]. For the reader's convenience, we provide a self-contained proof in Appendix C.3. □

Remark 4.8. Up to the constant $\log \mathcal{Z}$, the loss $\mathcal{L}_{\mathrm{RE}}$ coincides with the cost functional (1.16) associated to the optimal control formulation in Problem 1.3. The approach of minimizing the KL-divergence between $\mathbb{P}^u$ and $\mathbb{Q}$ as defined in (4.8) is thus directly linked to the perspective outlined in Problem 1.3. We refer to [131, 160] and Remark 2.46 for further details.

The cross-entropy loss admits a family of representations, indexed by $v \in \mathcal{U}$:

Proposition 4.9 (Cross-entropy loss)**.** *For $v \in \mathcal{U}$, let $(X^v_s)_{0\le s\le T}$ denote the unique strong solution to* (4.4)*, with u replaced by v. Then there exists a constant $C \in \mathbb{R}$ (not depending on u in the next line) such that*

$$\mathcal{L}_{\mathrm{CE}}(u) = \frac{1}{\mathcal{Z}}\,\mathbb{E}\Bigg[\left(\frac{1}{2}\int_0^T |u(X^v_s, s)|^2\,\mathrm{d}s - \int_0^T (u\cdot v)(X^v_s, s)\,\mathrm{d}s - \int_0^T u(X^v_s, s)\cdot \mathrm{d}W_s\right) \tag{4.16a}$$

$$\exp\left(-\int_0^T v(X^v_s, s)\cdot \mathrm{d}W_s - \frac{1}{2}\int_0^T |v(X^v_s, s)|^2\,\mathrm{d}s - \mathcal{W}(X^v)\right)\Bigg] + C, \tag{4.16b}$$

for all $u \in \mathcal{U}$.

Proof. See [308] or Appendix C.3 for a self-contained proof. □

Remark 4.10. The appearance of the exponential term in (4.16b) can be traced back to the reweighting

$$D^{\mathrm{CE}}(\mathbb{P}|\mathbb{Q}) = \mathbb{E}_{\mathbb{Q}}\left[\log\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\right)\right] = \mathbb{E}_{\mathbb{P}^v}\left[\log\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\right)\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}^v}\right], \tag{4.17}$$

recalling that $\mathbb{P}^v$ denotes the path measure associated to (4.4) controlled by v. While the choice of v evidently does not affect the loss function, judicious tuning may have a significant impact on the numerical performance by means of altering the statistical error for the associated estimators (see Section 4.1.3). We note that the expression (4.15) for the relative entropy loss can similarly be augmented by an additional control $v \in \mathcal{U}$. However, Proposition 4.29 in Section 4.3.2 discourages this approach and our numerical experiments using a reweighting for the relative entropy loss have not been promising. In general, we feel that exponential terms of the form appearing in (4.16b) often have a detrimental effect on the variance of estimators – this should also be compared to the analysis of the relative error of suboptimal importance sampling estimators in Chapter 3. Therefore, an important feature of both the relative entropy loss and the log-variance loss (see Proposition 4.12) seems to be that expectations can be taken with respect to controlled processes $(X^v_s)_{0\le s\le T}$ without incurring exponential factors as in (4.16b).

[25] These integrability conditions can be checked in practice using the formulas provided in Proposition 4.12 below.

Remark 4.11. Setting $v = 0$ leads to the simplification

$$\mathcal{L}_{\mathrm{CE}}(u) = \frac{1}{\mathcal{Z}}\,\mathbb{E}\left[\left(\frac{1}{2}\int_0^T |u(X_s,s)|^2\,\mathrm{d}s - \int_0^T u(X_s,s)\cdot \mathrm{d}W_s\right)\exp(-\mathcal{W}(X))\right] + C, \tag{4.18}$$

where $(X_s)_{0\le s\le T}$ solves the uncontrolled SDE (4.3). The quadratic dependence of $\mathcal{L}_{\mathrm{CE}}$ on u has been exploited in [308] to construct efficient Galerkin-type approximations of u^*.

Finally, we derive corresponding representations for the variance and log-variance losses:

Proposition 4.12 (Variance-type losses)**.** *For $v \in \mathcal{U}$, let $(X_s^v)_{0\le s\le T}$ denote the unique strong solution to* (4.4)*, with u replaced by v. Furthermore, define*

$$\widetilde{Y}_T^{u,v} = -\int_0^T (u\cdot v)(X_s^v,s)\,\mathrm{d}s - \int_0^T f(X_s^v,s)\,\mathrm{d}s - \int_0^T u(X_s^v,s)\cdot \mathrm{d}W_s + \frac{1}{2}\int_0^T |u(X_s^v,s)|^2\,\mathrm{d}s. \tag{4.19}$$

Then

$$\mathcal{L}_{\mathrm{Var}_v}(u) = \frac{1}{\mathcal{Z}^2}\,\mathrm{Var}\left(e^{\widetilde{Y}_T^{u,v} - g(X_T^v)}\right), \tag{4.20}$$

and

$$\mathcal{L}^{\log}_{\mathrm{Var}_v}(u) = \mathrm{Var}\left(\widetilde{Y}_T^{u,v} - g(X_T^v)\right), \tag{4.21}$$

for all $u \in \mathcal{U}$.

Proof. With $\widetilde{Y}_T^{u,v}$ defined as in (4.19) and using Theorem B.3, we compute for the variance loss

$$\mathcal{L}_{\mathrm{Var}_v}(u) = \mathrm{Var}_{\mathbb{P}^v}\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}^u}\right) = \mathrm{Var}_{\mathbb{P}^v}\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^u}\right) = \frac{1}{\mathcal{Z}^2}\,\mathrm{Var}_{\mathbb{P}^v}\left(e^{\widetilde{Y}_T^{u,v} - g(X_T^v)}\right). \tag{4.22}$$

Similarly, the log-variance loss equals

$$\mathcal{L}^{\log}_{\mathrm{Var}_v}(u) = \mathrm{Var}_{\mathbb{P}^v}\left(\log\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}^u}\right) = \mathrm{Var}_{\mathbb{P}^v}\left(\log\left(\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^u}\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\right)\right) = \mathrm{Var}_{\mathbb{P}^v}\left(\widetilde{Y}_T^{u,v} - g(X_T^v) - \log\mathcal{Z}\right) \tag{4.23a}$$

$$= \mathrm{Var}_{\mathbb{P}^v}\left(\widetilde{Y}_T^{u,v} - g(X_T^v)\right). \tag{4.23b}$$

□

Setting $v = u$ in (4.20) recovers the importance sampling objective in (1.10), i.e. the variance divergence $D^{\mathrm{Var}}_{\mathbb{P}^u}$ encodes the formulation from Problem 1.1. See also [213] and Chapter 3.

Remark 4.13. While different choices of v merely lead to distinct representations for the cross-entropy loss $\mathcal{L}_{\mathrm{CE}}$ according to Proposition 4.9 and Remark 4.10, the variance losses $\mathcal{L}_{\mathrm{Var}_v}$ and $\mathcal{L}^{\log}_{\mathrm{Var}_v}$ do indeed depend on v. However, the property $\mathcal{L}_{\mathrm{Var}_v}(u) = 0 \iff u = u^*$ (and similarly for $\mathcal{L}^{\log}_{\mathrm{Var}_v}$) holds for all $v \in \mathcal{U}$, by construction.

4.1.2 FBSDEs and the log-variance loss

As it turns out, the log-variance loss $\mathcal{L}^{\log}_{\mathrm{Var}_v}$ as computed in (4.21) is intimately connected to the FBSDE formulation in Problem 1.5 (and we already used the notation $\widetilde{Y}_T^{u,v}$ in hindsight). Indeed, setting $v = 0$ in Proposition 4.12 and writing

$$\mathrm{Var}\left(\widetilde{Y}_T^{u,0} - g(X_T^0)\right) = \mathrm{Var}\Big(\underbrace{\widetilde{Y}_T^{u,0} + y_0}_{=:Y_T^{u,0}} - g(X_T^0)\Big), \tag{4.24}$$

for some (at this point, arbitrary) constant $y_0 \in \mathbb{R}$, we recover the forward SDE (1.21a) and the backward SDE (1.21b) from (4.19) in conjunction with the optimality condition $\mathcal{L}^{\log}_{\mathrm{Var}_v}(u) = 0$, using also the identification $u^*(X_s,s) =: -Z_s$ suggested by (1.23). For arbitrary $v \in \mathcal{U}$, following Corollary 2.28, we similarly obtain the generalized FBSDE system

$$\mathrm{d}X_s^v = (b(X_s^v,s) + \sigma(X_s^v,s)v(X_s^v,s))\,\mathrm{d}s + \sigma(X_s^v,s)\,\mathrm{d}W_s, \qquad X_0^v = x_{\mathrm{init}}, \tag{4.25a}$$

$$\mathrm{d}Y_s^{u^*,v} = -f(X_s^v,s)\,\mathrm{d}s + v(X_s^v,s)\cdot Z_s\,\mathrm{d}s + \frac{1}{2}|Z_s|^2\,\mathrm{d}s + Z_s\cdot \mathrm{d}W_s, \qquad Y_T^{u^*,v} = g(X_T^v), \tag{4.25b}$$

again setting

$$Y_T^{u,v} = \widetilde{Y}_T^{u,v} + y_0. \tag{4.26}$$

In this sense, the divergence $D_{\mathbb{P}^v}^{\mathrm{Var(log)}}(\mathbb{P}^u|\mathbb{Q})$ encodes the dynamics (4.25). Let us again insist on the fact that by construction the solution $(Y_s, Z_s)_{0\le s\le T}$ to (4.25) does not depend on $v \in \mathcal{U}$ (the contribution $\sigma(X_s^v, s)v(X_s^v, s)\,\mathrm{d}s$ in (4.25a) being compensated for by the term $v(X_s^v, s)\cdot Z_s\,\mathrm{d}s$ in (4.25b)), whereas clearly $(X_s^v)_{0\le s\le T}$ does. When $u^*(X_s, s) = -Z_s$ is approximated in an iterative manner (see Section 4.4.1), the choice $v = u$ is natural as it amounts to applying the currently obtained estimate for the optimal control to the forward process (4.25a). In this context, the system (4.25) was put forward in [127, Section III.B]. The bearings of appropriate choices for v will be further discussed in Section 4.3.

It is instructive to compare the expression (4.24) for the log-variance loss to the 'moment loss'

$$\mathcal{L}_{\mathrm{moment}}(u, y_0) = \mathbb{E}\left[\left(Y_T^{u,0}(y_0) - g(X_T^0)\right)^2\right] \tag{4.27}$$

suggested in [86, 122] in the context of solving more general nonlinear parabolic PDEs[26]. More generally, we can define

$$\mathcal{L}_{\mathrm{moment}_v}(u, y_0) = \mathbb{E}\left[\left(Y_T^{u,v}(y_0) - g(X_T^v)\right)^2\right] \tag{4.28}$$

as a counterpart to the expression (4.21). This loss is essentially motivated from the FBSDE (4.25) and penalizes deviation from the terminal condition. We will further elaborate on it in Section 6.3.2. Note that unlike the losses considered so far, the moment losses depend on the additional parameter $y_0 \in \mathbb{R}$, which has implications in numerical implementations. Also, these losses do not admit a straightforward interpretation in terms of divergences between path measures. As we show in Proposition 4.22, algorithms based on $\mathcal{L}_{\mathrm{moment}_v}$ are in fact equivalent to their counterparts based on $\mathcal{L}_{\mathrm{Var}_v}^{\log}$ in the limit of infinite batch size when y_0 is chosen optimally or when the forward process is controlled in a certain way. We already anticipate that optimizing an additional parameter y_0 can slow down convergence towards the solution u^* considerably (see Section 4.4).

Remark 4.14. Reversing the argument, the log-variance loss can be obtained from (4.27) by replacing the second moment by the variance and using the translation invariance (4.24) to remove the dependence on y_0. The fact that this procedure leads to a viable loss function (i.e. satisfying $\mathcal{L}(u) = 0 \iff u = u^*$) can be traced back to the fact that the Hamilton-Jacobi PDE (1.20a) is itself translation invariant (i.e. it remains unchanged under the transformation $V \mapsto V + \mathrm{const}$). Following this argument, the log-variance loss can be applied for solving more general PDEs of the form (1.33) in the case when h depends on V only through ∇V. Furthermore, our interpretation in terms of divergences between probability measures on path space remains valid, at least in the case when σ is constant (in the following we let $\sigma = \mathrm{Id}_{d\times d}$ for simplicity)[27]. Indeed, denoting as before the path measure associated to (1.34a) by $\mathbb{P}$, defining the target $\mathbb{Q}$ via $\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}} \propto e^{-g}$, and introducing the approximation $\widetilde{u} \approx -\sigma^\top \nabla V$, the backward SDE (1.34b) induces a $\widetilde{u}$-dependent path measure $\mathbb{P}^{\widetilde{u}}$,

$$\frac{\mathrm{d}\mathbb{P}^{\widetilde{u}}}{\mathrm{d}\mathbb{P}}(X) \propto \exp\left(\int_0^T h(X_s, s, -\widetilde{u}(X_s, s))\,\mathrm{d}s - \int_0^T \widetilde{u}(X_s, s)\cdot\left(b(X_s, s)\,\mathrm{d}s - \mathrm{d}X_s\right)\right), \tag{4.29}$$

assuming that the right-hand side is $\mathbb{P}$-integrable. Replacing $Z \approx -\widetilde{u}$ in (1.34b) and denoting the corresponding process by $Y^{\widetilde{u}}$, we then obtain

$$\mathcal{L}(\widetilde{u}) = \mathrm{Var}_{\mathbb{P}}\left(\log\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}^{\widetilde{u}}}\right) = \mathrm{Var}\left(Y_T^{\widetilde{u}} - g(X_T)\right) \tag{4.30}$$

as an implementable loss function, with straightforward modifications to (1.34) when $\mathbb{P}$ is replaced by $\mathbb{P}^v$, see (4.25). Note, however, that in general the vector field $\widetilde{u}$ does not lend itself to a straightforward interpretation in terms of a control problem. The PDEs treated in [86, 122] and Chapter 6 do not possess the shift-invariance property (that is, h depends on V), and thus the vanishing of (4.30) does not characterize the solution to the PDE (1.33a) uniquely (not even up to additive constants). Uniqueness may be restored by including appropriate terms in (4.30) enforcing the terminal condition (1.33b). Theoretical and numerical properties of such extensions may be fruitful directions for future work, see also Chapter 7.

[26] We have employed the notation $Y_T^{u,0}(y_0)$ in order to stress the dependence on y_0 through (4.26).

[27] For more general diffusion coefficients, we can make similar arguments considering measures on the path space associated to $(W_t)_{t\ge 0}$, however departing slightly from the set-up in this thesis.

4.1.3 Algorithmic outline and empirical estimators

In order to motivate the theoretical analysis in the following sections, let us give a brief overview of algorithmic implementations based on the loss functions developed so far. We refer to Section 4.4.1 for a more detailed account. Recall that by the construction outlined in Section 4.1.1, the solution u^* as defined in (4.5) is characterized as the global minimum of $\mathcal{L}$, where $\mathcal{L}$ represents a generic loss function. Assuming a parametrization $\mathbb{R}^p \ni \theta \mapsto u_\theta$ (derived from, for instance, a Galerkin truncation or a neural network), we apply gradient-descent type methods to the function $\theta \mapsto \mathcal{L}(u_\theta)$, relying on the explicit expressions obtained in Propositions 4.7, 4.9 and 4.12. It is an important aspect that those expressions involve expectations that need to be estimated on the basis of ensemble averages. To approximate the loss $\mathcal{L}_{\mathrm{RE}}$, for instance, we use the estimator

$$\widehat{\mathcal{L}}^{(K)}_{\mathrm{RE}}(u) = \frac{1}{K}\sum_{k=1}^{K}\left[\frac{1}{2}\int_0^T |u(X_s^{u,(k)},s)|^2\,\mathrm{d}s + \int_0^T f(X_s^{u,(k)},s)\,\mathrm{d}s + g(X_T^{u,(k)})\right], \tag{4.31}$$

where $(X_s^{u,(k)})_{0\le s\le T}$, $k=1,\dots,K$ denote independent realizations of the solution to (4.4), and $K\in\mathbb{N}$ refers to the batch size. The estimators $\widehat{\mathcal{L}}^{(K)}_{\mathrm{CE}}(u)$, $\widehat{\mathcal{L}}^{(K)}_{\mathrm{Var}}(u)$, $\widehat{\mathcal{L}}^{\log,(K)}_{\mathrm{Var}}(u)$ and $\widehat{\mathcal{L}}^{(K)}_{\mathrm{moment}_v}(u,y_0)$ are constructed analogously, i.e. the estimator for the cross-entropy loss is given by

$$\widehat{\mathcal{L}}^{(K)}_{\mathrm{CE},v}(u) = \frac{1}{K}\sum_{k=1}^{K}\Bigg[\left(\frac{1}{2}\int_0^T |u(X_s^{v,(k)},s)|^2\,\mathrm{d}s - \int_0^T (u\cdot v)(X_s^{v,(k)},s)\,\mathrm{d}s - \int_0^T u(X^{v,(k)},s)\cdot \mathrm{d}W_s^{(k)}\right) \tag{4.32a}$$

$$\exp\left(-\int_0^T v(X_s^{v,(k)},s)\cdot \mathrm{d}W_s^{(k)} - \frac{1}{2}\int_0^T |v(X_s^{v,(k)},s)|^2\,\mathrm{d}s - \mathcal{W}(X^{v,(k)})\right)\Bigg], \tag{4.32b}$$

the estimator for the variance loss is given by

$$\widehat{\mathcal{L}}^{(K)}_{\mathrm{Var}_v}(u) = \frac{1}{K-1}\sum_{k=1}^{K}\left(e^{\widetilde{Y}_T^{u,v,(k)} - g(X_T^{v,(k)})} - \left(\overline{e^{\widetilde{Y}_T^{u,v} - g(X_T^v)}}\right)\right)^2, \tag{4.33}$$

the estimator for the log-variance loss by

$$\widehat{\mathcal{L}}^{\log(K)}_{\mathrm{Var}_v}(u) = \frac{1}{K-1}\sum_{k=1}^{K}\left(\widetilde{Y}_T^{u,v,(k)} - g(X_T^{v,(k)}) - \left(\overline{\widetilde{Y}_T^{u,v} - g(X_T^v)}\right)\right)^2, \tag{4.34}$$

and the estimator for the moment loss by

$$\widehat{\mathcal{L}}^{(K)}_{\mathrm{moment}_v}(u,y_0) = \frac{1}{K}\sum_{k=1}^{K}\left(\widetilde{Y}_T^{u,v,(k)} + y_0 - g(X_T^{v,(k)})\right)^2. \tag{4.35}$$

In the previous displays, the overline denotes an empirical mean, for example

$$\overline{\widetilde{Y}_T^{u,v} - g(X_T^v)} = \frac{1}{K}\sum_{k=1}^{K}\left(\widetilde{Y}_T^{u,v,(k)} - g(X_T^{v,(k)})\right), \tag{4.36}$$

and $(W_t^{(k)})_{t\ge0}$, $k=1,\dots,K$ denote independent Brownian motions associated to $(X_t^{u,(k)})_{t\ge0}$. By the law of large numbers, the convergence $\widehat{\mathcal{L}}^{(K)}(u)\to\mathcal{L}(u)$ holds almost surely up to additive and multiplicative constants[28], but as we show in Section 4.4, the fluctuations for finite K play a crucial role for the overall performance of the method. The variance associated to empirical estimators will hence be analyzed in Section 4.3.

Remark 4.15. The estimators introduced in this section are standard, and more elaborate constructions, for instance involving control variates [253, Section 4.4.2], can be considered to reduce the variance. We leave this direction for future work. It is noteworthy, however, that the log-variance estimator (4.34) appears to act as a control variate in natural way, see Propositions 4.19, 4.22, 5.6, Lemma 5.5 and Remark 4.23.

Remark 4.16. Note that the estimator $\widehat{\mathcal{L}}^{(K)}_{\mathrm{CE},v}$ depends on $v\in\mathcal{U}$, in contrast to its target $\mathcal{L}_{\mathrm{CE}}$; in other words, the limit $\lim_{K\to\infty}\widehat{\mathcal{L}}^{(K)}_{\mathrm{CE},v}(u)$ does not depend on v. This contrasts the pairs $(\widehat{\mathcal{L}}^{(K)}_{\mathrm{Var}_v},\mathcal{L}_{\mathrm{Var}_v})$ and $(\widehat{\mathcal{L}}^{\log,(K)}_{\mathrm{Var}_v},\mathcal{L}^{\log}_{\mathrm{Var}_v})$, see also Remark 4.10.

[28] More precisely, $\widehat{\mathcal{L}}^{(K)}_{\mathrm{RE}}(u)\to\mathcal{L}_{\mathrm{RE}}(u)-\log\mathcal{Z}$ and $\widehat{\mathcal{L}}^{(K)}_{\mathrm{CE},v}(u)\to\mathcal{Z}(\mathcal{L}_{\mathrm{CE}}(u)-C)$. The fact that the estimators $\widehat{\mathcal{L}}^{(K)}_{\mathrm{RE}}$ and $\widehat{\mathcal{L}}^{(K)}_{\mathrm{CE},v}$ do not depend on the intractable constants $\mathcal{Z}$ and C is crucial for the implementability of the associated methods.

We provide a sketch of the algorithmic procedure in Algorithm 1. Clearly, choosing different loss functions (and corresponding estimators) at every gradient step as indicated leads to viable algorithms. In particular, we have in mind the option of adjusting the forward control $v \in \mathcal{U}$ using the current approximation u_θ. More precisely, denoting by $u_\theta^{(j)}$ the approximation at the j^{th} step, it is reasonable to set $v = u_\theta^{(j)}$ in the iteration yielding $u_\theta^{(j+1)}$. In the remainder of this section, we will focus on this strategy for updating v, leaving differing schemes for future work.

Algorithm 1: Approximation of u^*

Choose a parametrization $\mathbb{R}^p \ni \theta \mapsto u_\theta$.
Initialize u_θ (with a parameter vector $\theta \in \mathbb{R}^p$).
Choose an optimization method *descent*, a batch size $K \in \mathbb{N}$ and a learning rate $\eta > 0$.
repeat
  Choose a loss function $\mathcal{L}$ and a corresponding estimator $\widehat{\mathcal{L}}^{(K)}$.
  Compute $\widehat{\mathcal{L}}^{(K)}(u_\theta)$ according to either (4.31), (4.32), (4.33), (4.34) or (4.35).
  Compute $\nabla_\theta \widehat{\mathcal{L}}^{(K)}(u_\theta)$ using automatic differentiation.
  Update parameters: $\theta \leftarrow \theta - \eta\, \mathit{descent}(\nabla_\theta \widehat{\mathcal{L}}^{(K)}(u_\theta))$.
until *convergence*;
Result: $u_\theta \approx u^*$.

4.2 Equivalence properties in the limit of infinite batch size

In this section we will analyze some of the properties of the losses defined in Section 4.1.1, not taking into account the approximation by ensemble averages described in Section 4.1.3. In other words, the results in this section are expected to be valid when the batch size K used to compute the estimators $\widehat{\mathcal{L}}^{(K)}$ is sufficiently large. The derivatives relevant for the gradient-descent type methodology described in Section 4.1.3 can be computed as follows,

$$\frac{\partial}{\partial \theta_i}\mathcal{L}(u_\theta) = \frac{\delta}{\delta u}\mathcal{L}(u;\phi_i)\Big|_{u=u_\theta}, \qquad \phi_i = \frac{\partial u_\theta}{\partial \theta_i}, \tag{4.37}$$

where $\frac{\delta}{\delta u}\mathcal{L}(u;\phi)$ denotes the Gâteaux derivative in direction ϕ. We recall its definition [269, Section 5.2]:

Definition 4.17 (Gâteaux derivative)**.** Let $u \in \mathcal{U}$ and $\phi \in C_b^1(\mathbb{R}^d \times [0,T], \mathbb{R}^d)$. A loss function $\mathcal{L} : \mathcal{U} \to \mathbb{R}$ is called *Gâteaux-differentiable* at u, if, for all $\phi \in C_b^1(\mathbb{R}^d \times [0,T], \mathbb{R}^d)$, the real-valued function $\varepsilon \mapsto \mathcal{L}(u + \varepsilon\phi)$ is differentiable at $\varepsilon = 0$. In this case we define the *Gâteaux derivative in direction* ϕ to be

$$\frac{\delta}{\delta u}\mathcal{L}(u;\phi) := \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\mathcal{L}(u + \varepsilon\phi). \tag{4.38}$$

Remark 4.18. The functions ϕ_i defined in (4.37) depend on the chosen parametrization for u. In the case when a Galerkin truncation is used, $u_\theta = \sum_i \theta_i \varphi_i$, these coincide with the chosen ansatz functions (i.e. $\phi_i = \varphi_i$). Concerning neural networks, the family $(\phi_i)_i$ reflects the choice of the architecture, the function ϕ_i encoding the response to a a change in the i^{th} weight. For convenience, we will throughout work under the assumption (implicit in Definition 4.17) that the functions ϕ_i are bounded, noting however that this could be relaxed with additional technical effort. Furthermore, note that Definition 4.17 extends straightforwardly to the estimator versions $\widehat{\mathcal{L}}^{(K)}$.

The following result shows that algorithms based on $\frac{1}{2}\mathcal{L}^{\log}_{\mathrm{Var}_v}$ and $\mathcal{L}_{\mathrm{RE}}$ behave equivalently in the limit of infinite batch size, provided that the update rule $v = u$ for the log-variance loss is applied (see the discussion towards the end of Section 4.1.3), and that *'all other things being equal'*, for instance in terms of network architecture and choice of optimizer. Furthermore, we provide an analytical expression for the gradient for future reference.

Proposition 4.19 (Equivalence of log-variance loss and relative entropy loss)**.** *Let $u, v \in \mathcal{U}$ and $\phi \in C_b^1(\mathbb{R}^d \times [0,T], \mathbb{R}^d)$. Then $\mathcal{L}^{\log}_{\mathrm{Var}_v}$ and $\mathcal{L}_{\mathrm{RE}}$ are Gâteaux-differentiable at u in direction ϕ. Furthermore,*

$$\frac{1}{2}\left(\frac{\delta}{\delta u}\mathcal{L}^{\log}_{\mathrm{Var}_v}(u;\phi)\right)\Big|_{v=u} = \frac{\delta}{\delta u}\mathcal{L}_{\mathrm{RE}}(u;\phi) = \mathbb{E}\left[\left(g(X_T^u) - \widetilde{Y}_T^{u,u}\right)\int_0^T \phi(X_s^u, s)\cdot \mathrm{d}W_s\right]. \tag{4.39}$$

Remark 4.20. Proposition 4.19 extends the connection between the cost functional (1.16) and the FBSDE formulation (1.21) exposed in Theorem 1.2. Indeed, the Problems 1.3 and 1.5 do not only agree on identifying the solution u^*; it is also the case that the gradients of the corresponding loss functions agree for $u \neq u^*$.

Moreover, it is instructive to compare the expressions (4.15) and (4.21) (or their sample based variants (4.31) and (4.34)). Namely, computing the derivatives associated to the relative entropy loss entails differentiating both the SDE-solution X^u as well as f and g, determining the running and terminal costs. Perhaps surprisingly, the latter is not necessary for obtaining the derivatives of the log-variance loss, opening the door for gradient-free implementations.

Proof of Proposition 4.19. We present a heuristic argument based on the perspective introduced in Section 4.1.1 and refer to Appendix C.3 for a rigorous proof.

For fixed $\mathbb{P} \in \mathcal{P}(\mathcal{C})$, let us consider perturbations $\mathbb{P}+\varepsilon\mathbb{U}$, where $\mathbb{U}$ is a signed measure with $\mathbb{U}(\mathcal{C}) = 0$. Assuming sufficient regularity, then

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0} D^{\mathrm{RE}}(\mathbb{P}+\varepsilon\mathbb{U}|\mathbb{Q}) = \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0} \mathbb{E}_{\mathbb{P}}\left[\log\left(\frac{\mathrm{d}(\mathbb{P}+\varepsilon\mathbb{U})}{\mathrm{d}\mathbb{Q}}\right)\frac{\mathrm{d}(\mathbb{P}+\varepsilon\mathbb{U})}{\mathrm{d}\mathbb{P}}\right] = \underbrace{\mathbb{E}_{\mathbb{P}}\left[\frac{\mathrm{d}\mathbb{U}}{\mathrm{d}\mathbb{P}}\right]}_{=0} + \mathbb{E}_{\mathbb{P}}\left[\log\left(\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}}\right)\frac{\mathrm{d}\mathbb{U}}{\mathrm{d}\mathbb{P}}\right], \quad (4.40)$$

where the first term on the right-hand side vanishes because of $\mathbb{U}(\mathcal{C}) = 0$. Likewise,

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0} D^{\mathrm{Var(log)}}_{\widetilde{\mathbb{P}}}(\mathbb{P}+\varepsilon\mathbb{U}|\mathbb{Q}) = \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\left(\mathbb{E}_{\widetilde{\mathbb{P}}}\left[\log^2\left(\frac{\mathrm{d}(\mathbb{P}+\varepsilon\mathbb{U})}{\mathrm{d}\mathbb{Q}}\right)\right] - \mathbb{E}_{\widetilde{\mathbb{P}}}\left[\log\left(\frac{\mathrm{d}(\mathbb{P}+\varepsilon\mathbb{U})}{\mathrm{d}\mathbb{Q}}\right)\right]^2\right) \quad (4.41a)$$

$$= 2\mathbb{E}_{\widetilde{\mathbb{P}}}\left[\log\left(\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}}\right)\frac{\mathrm{d}\mathbb{U}}{\mathrm{d}\mathbb{P}}\right] - 2\mathbb{E}_{\widetilde{\mathbb{P}}}\left[\log\left(\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}}\right)\right]\mathbb{E}_{\widetilde{\mathbb{P}}}\left[\frac{\mathrm{d}\mathbb{U}}{\mathrm{d}\mathbb{P}}\right]. \quad (4.41b)$$

For $\widetilde{\mathbb{P}} = \mathbb{P}$, the second term in (4.41b) vanishes (again, because of $\mathbb{U}(\mathcal{C}) = 0$), and hence (4.41b) agrees with (4.40) up to a factor of 2. □

Remark 4.21 (Covariance structure and local minima). It is interesting to note that (4.41) can be expressed as

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0} D^{\mathrm{Var(log)}}_{\widetilde{\mathbb{P}}}(\mathbb{P}+\varepsilon\mathbb{U}|\mathbb{Q}) = \mathrm{Cov}_{\widetilde{\mathbb{P}}}\left(\log\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}}, \frac{\mathrm{d}\mathbb{U}}{\mathrm{d}\mathbb{P}}\right). \quad (4.42)$$

In particular, the derivative is zero for all $\mathbb{U}$ with $\mathbb{U}(\mathcal{C}) = 0$ if and only if $\mathbb{P} = \mathbb{Q}$. In other words, we expect the loss landscape associated to losses based on the log-variance divergence to be free of local minima where the optimization procedure could get stuck. By similar reasoning we expect the same for the relative entropy and cross-entropy losses (compare also to [58, Theorem 2.7.2]). A more refined analysis concerning the former can be found in [195]. The covariance structure will also turn out to be important in statistical properties of the estimator. Note that we can write

$$\mathrm{Cov}_{\widetilde{\mathbb{P}}}\left(\log\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}}, \frac{\mathrm{d}\mathbb{U}}{\mathrm{d}\mathbb{P}}\right) = \mathbb{E}_{\widetilde{\mathbb{P}}}\left[\left(\log\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}} - \mathbb{E}_{\widetilde{\mathbb{P}}}\left[\log\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}}\right]\right)\left(\frac{\mathrm{d}\mathbb{U}}{\mathrm{d}\mathbb{P}} - \mathbb{E}_{\widetilde{\mathbb{P}}}\left[\frac{\mathrm{d}\mathbb{U}}{\mathrm{d}\mathbb{P}}\right]\right)\right] \quad (4.43)$$

and the two 'centerings' indicate potential numerical advantages. We refer to Remark 4.23 and Chapter 5 for further discussions.

In the following proposition, we gather results concerning the moment loss $\mathcal{L}_{\mathrm{moment}_v}$ defined in (4.28). The first statement is analogous to Proposition 4.19 and shows that $\mathcal{L}_{\mathrm{moment}_v}$ and $\mathcal{L}^{\log}_{\mathrm{Var}_v}$ are equivalent in the infinite batch size limit, provided that the update strategy $v = u$ is employed. The second statement deals with the alternative $v \neq u$. In this case, $y_0 = -\log\mathcal{Z}$ (i.e. finding the optimal y_0 according to Theorem 1.2) is necessary for $\mathcal{L}_{\mathrm{moment}_v}$ to identify the correct u^*. Consequently, approximation of the optimal control will be inaccurate unless the parameter y_0 is determined without error.

Proposition 4.22 (Properties of the moment loss)**.** *Let $u, v \in \mathcal{U}$ and $y_0 \in \mathbb{R}$. Then the following holds:*

1. *The losses $\mathcal{L}_{\mathrm{moment},v}(\cdot, y_0)$ and $\mathcal{L}^{\log}_{\mathrm{Var}_v}$ are Gâteaux-differentiable at u, and*

$$\left(\frac{\delta}{\delta u}\mathcal{L}_{\mathrm{moment}_v}(u, y_0; \phi)\right)\Big|_{v=u} = \left(\frac{\delta}{\delta u}\mathcal{L}^{\log}_{\mathrm{Var}_v}(u; \phi)\right)\Big|_{v=u} \quad (4.44)$$

   holds for all $\phi \in C^1_b(\mathbb{R}^d \times [0,T], \mathbb{R}^d)$. In particular, (4.44) is zero at $u = u^*$, independently of y_0.

2. *If $v \neq u$, then*

$$\frac{\delta}{\delta u}\mathcal{L}_{\mathrm{moment}_v}(u, y_0; \phi) = 0 \quad (4.45)$$

   holds for all $\phi \in C^1_b(\mathbb{R}^d \times [0,T], \mathbb{R}^d)$ if and only if $u = u^*$ and $y_0 = -\log\mathcal{Z}$.

Proof. The proof can be found in Appendix C.3. □

Remark 4.23 (Control variates). Inspecting the proofs of Propositions 4.19 and 4.22, we see that the identities (4.39) and (4.44) rest on the expression

$$\frac{\delta}{\delta u}\mathcal{L}(u)\Big|_{v=u} = \mathbb{E}\left[\left(g(X_T^v) - \widetilde{Y}_T^{u,v}\right)\int_0^T \phi(X_s^v, s)\cdot \mathrm{d}W_s\right] + \beta\,\mathbb{E}\left[\int_0^T \phi(X_s^v, s)\cdot \mathrm{d}W_s\right], \tag{4.46}$$

noting the vanishing of last term, where $\beta = -y_0$ for the moment loss and $\beta = -\mathbb{E}\left[g(X_T^u) - \widetilde{Y}_T^{u,u}\right]$ for the log-variance loss. The corresponding Monte Carlo estimators (see Section 4.1.3) hence include terms that are zero in expectation and act as control variates. The general idea of control variates is to add a quantity with known expectation that might correlate with the estimator in such a way that the overall variance is reduced [253, Section 4.4.2]. Here we rely on the Itô integral representing this quantity, i.e. we consider the control variate $\int_0^T \phi(X_s^v, s)\cdot \mathrm{d}W_s$. Using the explicit expression for the derivative in (4.39), the optimal value for the control variate scaling β in terms of variance reduction, as stated in Lemma 5.1, is given by

$$\beta^* = -\frac{\mathrm{Cov}\left(\left(g(X_T^u) - \widetilde{Y}_T^{u,u}\right)\int_0^T \phi(X_s^u, s)\cdot \mathrm{d}W_s, \int_0^T \phi(X_s^u, s)\cdot \mathrm{d}W_s\right)}{\mathrm{Var}\left(\int_0^T \phi(X_s^u, s)\cdot \mathrm{d}W_s\right)} \tag{4.47a}$$

$$= -\mathbb{E}\left[g(X_T^u) - \widetilde{Y}_T^{u,u}\right] - \frac{\mathrm{Cov}\left(g(X_T^u) - \widetilde{Y}_T^{u,u}, \left(\int_0^T \phi(X_s^u, s)\cdot \mathrm{d}W_s\right)^2\right)}{\mathbb{E}\left[\left(\int_0^T \phi(X_s^u, s)\cdot \mathrm{d}W_s\right)^2\right]}, \tag{4.47b}$$

which splits into a ϕ-independent (i.e. shared across network weights) and a ϕ-dependent (i.e. weight-specific) term (see also Lemma 5.5). The ϕ-independent term is reproduced in expectation by the log-variance estimator. Numerical evidence suggests that the ϕ-dependent term is often small and fluctuates around zero, but implementations that include this contribution (based on Monte Carlo estimates) hold the promise of further variance reductions. We note however that determining a control variate for every weight carries a significant computational overhead and that Monte Carlo errors need to be taken into account. Finally, if y_0 in the moment loss differs greatly from $-\mathbb{E}\left[g(X_T^u) - \widetilde{Y}_T^{u,u}\right]$, we expect the corresponding variance to be large, hindering algorithmic performance. In Chapter 5 we have provided a more detailed analysis of the connections between the log-variance divergences and variance reduction techniques in the context of computational Bayesian inference, in particular leading to more rigorous statements on the connection to (optimal) control variate scalings (see Proposition 5.6).

4.3 Finite sample properties and the variance of estimators

In this section we investigate properties of the sample versions of the losses as outlined in Section 4.1.3 and, in particular, study their variances and relative errors. We will highlight two different types of robustness, both of which prove significant for convergence speed and stability concerning practical implementations of Algorithm 1, see the numerical experiments in Section 4.4.

4.3.1 Robustness at the solution u^*

By construction, the optimal control solution u^* represents the global minimum of all considered losses. Consequently, the associated directional derivatives vanish at u^*, i.e.

$$\frac{\delta}{\delta u}\Big|_{u=u^*}\mathcal{L}(u;\phi) = 0, \tag{4.48}$$

for all $\phi \in C_b^1(\mathbb{R}^d \times [0,T], \mathbb{R}^d)$. A natural question is whether similar statements can be made with respect to the corresponding Monte Carlo estimators. We make the following definition.

Definition 4.24 (Robustness at the solution u^*)**.** We say that an estimator $\widehat{\mathcal{L}}^{(N)}$ is *robust at the solution* u^* if

$$\mathrm{Var}\left(\frac{\delta}{\delta u}\Big|_{u=u^*}\widehat{\mathcal{L}}^{(K)}(u;\phi)\right) = 0, \tag{4.49}$$

for all $\phi \in C_b^1(\mathbb{R}^d \times [0,T], \mathbb{R}^d)$ and $K \in \mathbb{N}$.

Robustness at the solution u^* implies that fluctuations in the gradient due to Monte Carlo errors are suppressed close to u^*, facilitating accurate approximation. Conversely, if robustness at u^* does not hold, then the relative error (i.e. the Monte Carlo error relative to the size of the gradients (4.37)) can be large near u^*, potentially incurring instabilities of the gradient-descent type scheme. We refer to Figure 4.12 in the next section and the corresponding discussion for an illustration of this phenomenon.

Proposition 4.25 (Robustness and non-robustness at u^*)**.** *The following holds:*

1. *The variance estimator $\widehat{\mathcal{L}}^{(K)}_{\mathrm{Var}_v}$ and the log-variance estimator $\widehat{\mathcal{L}}^{\log(K)}_{\mathrm{Var}_v}$ are robust at u^*, for all $v \in \mathcal{U}$.*

2. *For all $v \in \mathcal{U}$, the moment estimator $\widehat{\mathcal{L}}^{(K)}_{\mathrm{moment}_v}(\cdot, y_0)$ is robust at u^*, i.e.*

$$\mathrm{Var}\left(\frac{\delta}{\delta u}\Big|_{u=u^*} \widehat{\mathcal{L}}^{(K)}_{\mathrm{moment}_v}(u, y_0; \phi)\right) = 0, \qquad \text{for all } \phi \in C^1_b(\mathbb{R}^d \times [0,T], \mathbb{R}^d), \tag{4.50}$$

   if and only if $y_0 = -\log \mathcal{Z}$.

3. *The relative entropy estimator $\widehat{\mathcal{L}}^{(K)}_{\mathrm{RE}}$ is not robust at u^*. More precisely, for $\phi \in C^1_b(\mathbb{R}^d \times [0,T], \mathbb{R}^d)$,*

$$\mathrm{Var}\left(\frac{\delta}{\delta u}\Big|_{u=u^*} \widehat{\mathcal{L}}^{(K)}_{\mathrm{RE}}(u; \phi)\right) = \frac{1}{K}\,\mathbb{E}\left[\int_0^T |(\nabla u^*)^\top(X_s^{u^*}, s) A_s|^2 \,\mathrm{d}s\right], \tag{4.51}$$

   where $(A_s)_{0\le s\le T}$ denotes the unique strong solution to the SDE

$$\mathrm{d}A_s = (\sigma\phi)(X_s^{u^*}, s)\,\mathrm{d}s + \Big[(\nabla b + \nabla(\sigma u^*))(X_s^{u^*}, s)\Big]^\top A_s\,\mathrm{d}s + A_s \cdot \nabla\sigma(X_s^{u^*}, s)\,\mathrm{d}W_s, \qquad A_0 = 0. \tag{4.52}$$

4. *For all $v \in \mathcal{U}$, the cross-entropy estimator $\widehat{\mathcal{L}}^{(K)}_{\mathrm{CE},v}$ is not robust at u^*.*

Remark 4.26. The fact that robustness of the moment estimator at u^* requires $y_0 = -\log \mathcal{Z}$ might lead to instabilities in practice as this relation is rarely satisfied exactly. Note that the variance of the relative entropy estimator at u^* depends on ∇u^*. We thus expect instabilities in metastable settings, where often this quantity is fairly large. For numerical confirmation, see Figure 4.12 and the related discussion.

Proof. For illustration, we show the robustness of the log-variance estimator $\widehat{\mathcal{L}}^{\log(K)}_{\mathrm{Var}_v}$. The remaining proofs are deferred to Appendix C.3.

By a straightforward calculation (essentially equivalent to (C.37) in Appendix C.3), we see that

$$\frac{\delta}{\delta u}\widehat{\mathcal{L}}^{\log(K)}_{\mathrm{Var}_v}(u;\phi) = \frac{2}{K-1}\sum_{k=1}^K \left[\left(g\left(X_T^{v,(k)}\right) - \widetilde{Y}_T^{u,v,(k)}\right)\frac{\delta \widetilde{Y}_T^{u,v,(k)}}{\delta u}(u;\phi)\right] \tag{4.53a}$$

$$- \frac{2}{K(K-1)}\sum_{k=1}^K \left[\left(g\left(X_T^{v,(k)}\right) - \widetilde{Y}_T^{u,v,(k)}\right)\right]\sum_{k=1}^K \left[\frac{\delta \widetilde{Y}_T^{u,v,(k)}}{\delta u}(u;\phi)\right], \tag{4.53b}$$

where

$$\frac{\delta \widetilde{Y}_T^{u,v,(k)}}{\delta u}(u;\phi) = \int_0^T \phi(X_s^{v,(k)}, s)\cdot \mathrm{d}W_s^{(k)} - \int_0^T (\phi\cdot(u-v))(X_s^{v,(k)}, s)\,\mathrm{d}s. \tag{4.54}$$

The claim now follows from observing that

$$\left(g\left(X_T^{v,(k)}\right) - \widetilde{Y}_T^{u,v,(k)}\right)\Big|_{u=u^*} \tag{4.55}$$

is almost surely constant (i.e. does not depend on k), according to the second equation in (4.25b). □

4.3.2 Stability in high dimensions – robustness under tensorization

In this section we study the robustness of the proposed algorithms in high-dimensional settings. As a motivation, consider the case when the drift and diffusion coefficients in the uncontrolled SDE (4.3) split into separate contributions along different dimensions,

$$b(x,s) = \sum_{i=1}^d b_i(x_i, s), \qquad \sigma(x,s) = \sum_{i=1}^d \sigma_i(x_i, s), \tag{4.56}$$

for $x = (x_1, \dots, x_d) \in \mathbb{R}^d$, and analogously for the running and terminal costs f and g as well as for the control vector field u. It is then straightforward to show that the path measure $\mathbb{P}^u$ associated to the controlled SDE (4.4) and the target measure $\mathbb{Q}$ defined in (1.13) factorize,

$$\mathbb{P}^u = \bigotimes_{i=1}^{d} \mathbb{P}^{u_i}, \qquad \mathbb{Q} = \bigotimes_{i=1}^{d} \mathbb{Q}_i. \tag{4.57}$$

From the perspective of statistical physics, (4.57) corresponds to the scenario where non-interacting systems are considered simultaneously. To study the case when d grows large, we leverage the perspective put forward in Section 4.1.1, recalling that $D(\mathbb{P}|\mathbb{Q})$ denotes a generic divergence. In what follows, we will denote corresponding estimators based on a sample of size K by $\widehat{D}^{(K)}(\mathbb{P}|\mathbb{Q})$, and study the quantity

$$r^{(K)}(\mathbb{P}|\mathbb{Q}) := \frac{\sqrt{\mathrm{Var}\left(\widehat{D}^{(K)}(\mathbb{P}|\mathbb{Q})\right)}}{D(\mathbb{P}|\mathbb{Q})}, \tag{4.58}$$

measuring the relative statistical error when estimating $D(\mathbb{P}|\mathbb{Q})$ from samples, noting that $r^{(K)}(\mathbb{P}|\mathbb{Q}) = \mathcal{O}(K^{-1/2})$. As $r^{(K)}$ is clearly linked to algorithmic performance and stability, we are interested in divergences, corresponding loss functions and estimators whose relative error remains controlled when the number of independent factors in (4.57) increases:

Definition 4.27 (Robustness under tensorization)**.** We say that a divergence $D : \mathcal{P}(\mathcal{C}) \times \mathcal{P}(\mathcal{C}) \to \mathbb{R} \cup \{\infty\}$ and a corresponding estimator $\widehat{D}^{(K)}$ are *robust under tensorization* if, for all $\mathbb{P}, \mathbb{Q} \in \mathcal{P}(\mathcal{C})$ such that $D(\mathbb{P}|\mathbb{Q}) < \infty$ and $K \in \mathbb{N}$, there exists $C > 0$ such that

$$r^{(K)}\left(\bigotimes_{i=1}^{M} \mathbb{P}_i \middle| \bigotimes_{i=1}^{M} \mathbb{Q}_i\right) < C, \tag{4.59}$$

for all $M \in \mathbb{N}$. Here, $\mathbb{P}_i$ and $\mathbb{Q}_i$ represent identical copies of $\mathbb{P}$ and $\mathbb{Q}$, respectively, so that $\bigotimes_{i=1}^{M} \mathbb{P}_i$ and $\bigotimes_{i=1}^{M} \mathbb{Q}_i$ are measures on the product space $\bigotimes_{i=1}^{M} C([0,T], \mathbb{R}^d) \simeq C([0,T], \mathbb{R}^{Md})$.

Clearly, if $\mathbb{P}$ and $\mathbb{Q}$ are measures on $C([0,T], \mathbb{R})$, then M coincides with the dimension of the combined problem.

Remark 4.28. The variance and log-variance divergences defined in (4.11) and (4.12) depend on an auxiliary measure $\widetilde{\mathbb{P}}$. Definition 4.27 extends straightforwardly by considering the product measures $\bigotimes_{i=1}^{d} \widetilde{\mathbb{P}}_i$. In a similar vein, the relative entropy and cross-entropy divergences admit estimators that depend on a further probability measure $\widetilde{\mathbb{P}}$,

$$\widehat{D}^{\mathrm{RE},(K)}_{\widetilde{\mathbb{P}}}(\mathbb{P}|\mathbb{Q}) = \frac{1}{K}\sum_{k=1}^{K}\left[\log\left(\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}}\right)\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\widetilde{\mathbb{P}}}\right](X^{(k)}), \quad \widehat{D}^{\mathrm{CE},(K)}_{\widetilde{\mathbb{P}}}(\mathbb{P}|\mathbb{Q}) = \frac{1}{K}\sum_{k=1}^{K}\left[\log\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\right)\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\widetilde{\mathbb{P}}}\right](X^{k}), \tag{4.60}$$

where $X^{(k)} \sim \widetilde{\mathbb{P}}$, motivated by the identities $D^{\mathrm{RE}}(\mathbb{P}|\mathbb{Q}) = \mathbb{E}_{\widetilde{\mathbb{P}}}\left[\log\left(\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}}\right)\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\widetilde{\mathbb{P}}}\right]$ and $D^{\mathrm{CE}}(\mathbb{P}|\mathbb{Q}) = \mathbb{E}_{\widetilde{\mathbb{P}}}\left[\log\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\right)\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\widetilde{\mathbb{P}}}\right]$. We refer to Remark 4.10 for a similar discussion.

Proposition 4.29. *We have the following robustness and non-robustness properties:*

1. *The log-variance divergence $D^{\mathrm{Var(log)}}_{\widetilde{\mathbb{P}}}$, approximated using the standard Monte Carlo estimator, is robust under tensorization, for all $\widetilde{\mathbb{P}} \in \mathcal{P}(\mathcal{C})$.*
2. *The relative entropy divergence D^{RE}, estimated using $\widehat{D}^{\mathrm{RE},(K)}_{\widetilde{\mathbb{P}}}$, is robust under tensorization if and only if $\widetilde{\mathbb{P}} = \mathbb{P}$.*
3. *The variance divergence $D^{\mathrm{Var}}_{\widetilde{\mathbb{P}}}$ is not robust under tensorization when approximated using the standard Monte Carlo estimator. More precisely, if $\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}$ is not $\widetilde{\mathbb{P}}$-almost surely constant, then, for fixed $K \in \mathbb{N}$, there exist constants $a > 0$ and $C > 1$ such that*

$$r^{(K)}\left(\bigotimes_{i=1}^{M} \mathbb{P}_i \middle| \bigotimes_{i=1}^{M} \mathbb{Q}_i\right) \geq a\, C^M, \tag{4.61}$$

for all $M \geq 1$.

4. *The cross-entropy divergence D^{RE}, estimated using $\widehat{D}^{\mathrm{RE},(K)}_{\widetilde{\mathbb{P}}}$, is not robust under tensorization. More precisely, for fixed $K \in \mathbb{N}$ there exists a constant $a > 0$ such that*

$$r^{(K)}\left(\bigotimes_{i=1}^{M}\mathbb{P}_i \Big| \bigotimes_{i=1}^{M}\mathbb{Q}_i\right) \geq a\left(\sqrt{\chi^2(\mathbb{Q}|\widetilde{\mathbb{P}})+1}\right)^M, \tag{4.62}$$

for all $M \geq 1$. Here

$$\chi^2(\mathbb{Q}|\widetilde{\mathbb{P}}) = \mathbb{E}_{\widetilde{\mathbb{P}}}\left[\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\widetilde{\mathbb{P}}}\right)^2 - 1\right] \tag{4.63}$$

denotes the χ^2-divergence between $\mathbb{Q}$ and $\widetilde{\mathbb{P}}$.

Proof. See Appendix C.3. □

Remark 4.30. Loosely speaking, the reason for the robustness under tensorization of the log-variance loss and the relative entropy loss for the case $\widetilde{\mathbb{P}} = \mathbb{P}$ can be traced back to the fact that measures appear inside the logarithm, which turns products into sums. In particular, Proposition 4.29 suggests that the variance and cross-entropy losses perform poorly in high-dimensional settings as the relative errors (4.61) and (4.62) scale exponentially in M. Numerical support can be found in Section 4.4. For the variance loss, this should also be compared to an analysis on relative errors of suboptimal importance sampling in Chapter 3. We note that in practical scenarios we have that $\widetilde{\mathbb{P}} \neq \mathbb{Q}$ as it is not feasible to sample from the target, and hence $\sqrt{\chi^2(\mathbb{Q}|\widetilde{\mathbb{P}})+1} > 1$.

4.4 Numerical experiments for path space approximations

In this section we illustrate our theoretical results on the basis of numerical experiments. In Section 4.4.1 we discuss computational details of our implementations, complementing the discussion in Section 4.1.3. The Sections 4.4.2 and 4.4.3 focus on the case when the uncontrolled SDE (4.3) describes an Ornstein-Uhlenbeck process and the dimension is comparatively large. In Section 4.4.4 we consider metastable settings (of both low and moderate dimensionality), representative of those typically encountered in rare event simulations (see Example 1.1). We rely on PyTorch as a tool for automatic differentiation and refer to the code at `https://github.com/lorenzrichter/path-space-PDE-solver`.

4.4.1 Computational aspects

The numerical treatment of the Problems 1.1-1.5 using the IDO-methodology is based on the explicit loss function representations in Section 4.1.1, together with a gradient descent scheme relying on automatic differentiation[29]. Following the discussion in Section 4.1.3, a particular instance of an IDO-algorithm is determined by the choice of a loss function, and, in the case of the cross-entropy, moment and variance-type losses, by a strategy to update the control vector field v in the forward dynamics (see Propositions 4.9 and 4.12). As mentioned towards the end of Section 4.1.3, we focus on setting $v = u$ at each gradient step, i.e. to use the current approximation as a forward control. Importantly, we do not differentiate the loss with respect to v; in practice this can be achieved by removing the corresponding variables from the autodifferentiation computational graph (for instance using the `detach` command in the PyTorch package). Including differentiation with respect to v as well as more elaborate choices of the forward control might be rewarding directions for future research.
Practical implementations require approximations at three different stages: first, the time discretization of the SDEs (4.3) or (4.4); second, the Monte Carlo approximation of the losses (as outlined in Section 4.1.3), or, to be precise, the approximation of their respective gradients; and third, the function approximation of either the optimal control vector field u^* or the value function V. Moreover, implementations vary according to the choice of an appropriate gradient descent method.
Concerning the first point, we discretize the SDE (4.4) using the Euler-Maruyama scheme [173] along a time grid $0 = t_0 < \cdots < t_N = T$, namely iterating

$$\widehat{X}^u_{n+1} = \widehat{X}^u_n + \Big(b(\widehat{X}^u_n, t_n) + \sigma(\widehat{X}^u_n, t_n)u(\widehat{X}^u_n, t_n)\Big)\,\Delta t + \sigma(\widehat{X}^u_n, t_n)\xi_{n+1}\sqrt{\Delta t}, \qquad \widehat{X}_0 = x_{\mathrm{init}}, \tag{4.64}$$

where $\Delta t > 0$ denotes the step size, and $\xi_n \sim \mathcal{N}(0, \mathrm{Id}_{d\times d})$ are independent standard Gaussian random variables. Recall that the initial value can be random rather than deterministic (see Remark 1.4). We demonstrate the potential benefit of sampling $\widehat{X}_0$ from a given density in Section 4.4.3.

[29] Note that for the gradients of the process $(X^u_s)_{0\leq s\leq T}$ alternative computational methods can be considered (see [111] for an overview). A numerical analysis of the approach we rely on can be found in [300].

We next discuss the approximation of u^*. First, note that a viable and straightforward alternative is to instead approximate V and compute $u^* = -\sigma^\top \nabla V$ whenever needed (for instance by automatic differentiation), see [241]. However, this approach has performed slightly worse in our experiments, and, furthermore, V can be recovered from u^* by integration along an appropriately chosen curve. To approximate u^*, a classic option is a to use a Galerkin truncation, i.e. a linear combination of ansatz functions

$$u(x, t_n) = \sum_{m=1}^{M} \theta_m^n \varphi_m(x), \tag{4.65}$$

for $n \in \{0, \dots, N-1\}$ with parameters $\theta_m^n \in \mathbb{R}$. Choosing an appropriate set $\{\varphi_m\}_{m=1}^M$ is crucial for algorithmic performance – a task that in high-dimensional settings requires detailed a priori knowledge about the problem at hand. Instead, we focus on approximations of u^* realized by neural networks as defined in Definitions 2.49 and 2.50.

Neural networks are known to be universal function approximators [59, 141], with recent results indicating favorable properties in high-dimensional settings [91, 92, 118, 232, 266]. The control u can be represented by either $u(x,t) = \Phi_\varrho(y)$ with $y = (x,t)^\top$, i.e. using one neural network for both the space and time dependence, or by $u(x, t_n) = \Phi_\varrho^n(x)$, using one neural network per time step. The former alternative led to better performance in our experiments, and the reported results rely on this choice. For the gradient descent step we either choose SGD with constant learning rate [114, Algorithm 8.1] or Adam [114, Algorithm 8.7], [169], a variant that relies on adaptive step sizes and momenta (cf. Section 2.4). Further numerical investigations on network architectures and optimization heuristics can be found in [50].

To evaluate algorithmic choices we monitor the following two performance metrics:

1. The *importance sampling relative error*, namely

$$\delta(u) := \frac{\sqrt{\mathrm{Var}\left(e^{-\mathcal{W}(X^u)} \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^u}(X^u)\right)}}{\mathbb{E}[e^{-\mathcal{W}(X)}]}, \tag{4.66}$$

   where u is the approximated control in the corresponding iteration step. This quantity is zero if and only if $u = u^*$ (cf. Theorem 1.2) and measures the quality of the control in terms of the objective introduced in Problem 1.1. Since its Monte Carlo version fluctuates heavily if u is far away from u^* we usually estimate this quantity with additional samples not being used in the gradient computation.

2. An L^2*-error*,

$$\mathbb{E}\left[\int_0^T |u - u^*_{\mathrm{ref}}|^2(X_s^u, s)\,\mathrm{d}s\right], \tag{4.67}$$

   where u^*_{ref} is computed either analytically or using a finite difference scheme for the HJB PDE (1.20). This quantity is more robust w.r.t. deviations from u^* and therefore we compute the Monte Carlo estimator using just the samples from the training iteration.

4.4.2 Ornstein-Uhlenbeck dynamics with linear costs

Let us consider the controlled Ornstein-Uhlenbeck process

$$\mathrm{d}X_s^u = (AX_s^u + Bu(X_s^u, s))\,\mathrm{d}s + B\,\mathrm{d}W_s, \quad X_0^u = 0, \tag{4.68}$$

where $A, B \in \mathbb{R}^{d\times d}$. Furthermore, we assume zero running costs, $f = 0$, and linear terminal costs $g(x) = \gamma \cdot x$, for a fixed vector $\gamma \in \mathbb{R}^d$. As shown in Section B.5.1, the optimal control is given by

$$u^*(x,t) = -B^\top e^{A^\top (T-t)}\gamma, \tag{4.69}$$

which remarkably does not depend on x. Therefore, not only the variance and log-variance losses are robust at u^* in the sense of Definition 4.24, but also the relative entropy loss, according to (4.51) in Proposition 4.25.

We choose $A = -\mathrm{Id}_{d\times d} + (\xi_{ij})_{1\le i,j\le d}$ and $B = \mathrm{Id}_{d\times d} + (\xi_{ij})_{1\le i,j\le d}$, where $\xi_{ij} \sim \mathcal{N}(0, \nu^2)$ are sampled i.i.d. once at the beginning of the simulation. Note that this choice corresponds to a small perturbation of the product setting from Section 4.3.2. We set $T = 1, \nu = 0.1$, $\gamma = (1, \dots, 1)^\top$ and as function approximation take the DenseNet from Definition 2.50 using two hidden layers, each with a width of $n_1 = n_2 = 30$, and $\varrho = \max(0, x)$ as the nonlinearity. Lastly, we choose the Adam optimizer as a gradient descent scheme. Figure 4.1 shows the algorithm's performance for $d = 1$ with batch size $K = 200$, learning rate $\eta = 0.01$ and step size $\Delta t = 0.01$. We observe that log-variance, relative entropy and moment loss perform similarly and converge well to a suitable approximation. The cross-entropy loss decreases, but at later gradient steps fluctuates more than the other

losses (we note that the fluctuations appear to be less pronounced when using SGD, however at the cost of substantially slowing down the overall speed of convergence). The inferior quality of the control obtained using the cross-entropy loss may be explained by its non-robustness at u^*, see Proposition 4.25.

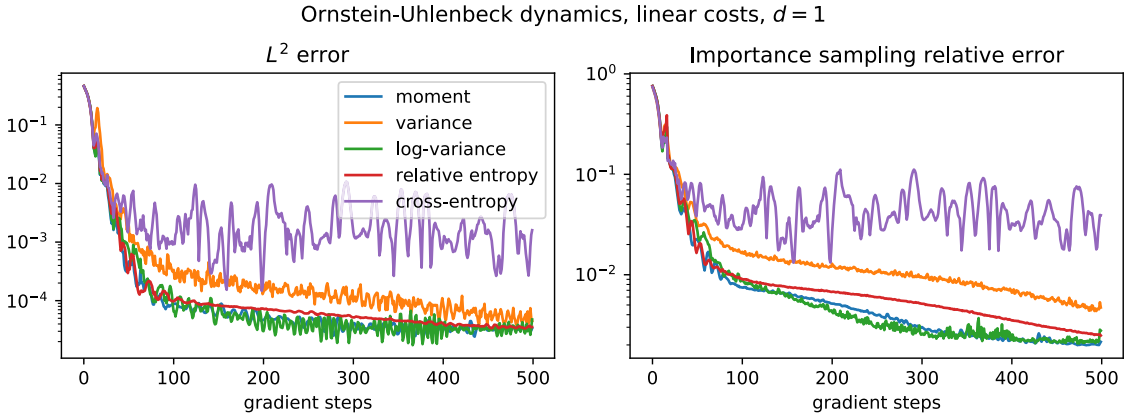


Figure 4.1: Performance of the algorithm using five different loss functions according to the metrics introduced in Section 4.4.1 as a function of the iteration step.

Figure 4.2 shows the algorithm's performance in a high-dimensional case, $d = 40$, where we now choose $K = 500$ as the batch size, $\eta = 0.001$ as the learning rate, $\Delta t = 0.01$ as the time step, and as before rely on a DenseNet with two hidden layers. We observe that relative entropy loss and log-variance loss perform best, and that the moment and cross-entropy losses converge at a significantly slower rate. The variance loss is numerically unstable and hence not represented in Figure 4.2. We encounter similar problems in the subsequent experiments and thus do not consider the variance loss in what follows. In Figure 4.3 we plot some of the components of the 40-dimensional approximated optimal control vector field as well as the analytic solution $u^*_{\text{ref}}(x,t)$ for a fixed value of x and varying time t, showcasing the inferiority of the approximation obtained using the cross-entropy loss. The comparatively poor performance of the cross-entropy and the variance losses can be attributed to their non-robustness with respect to tensorizations, see Section 4.3.2. To further illustrate these results, Figure 4.4 displays the relative error associated to the loss estimators computed from $K = 1.5 \cdot 10^7$ samples in different dimensions. The dimensional dependence agrees with what is expected from Proposition 4.29, but we note that our numerical experiment goes beyond the product case.

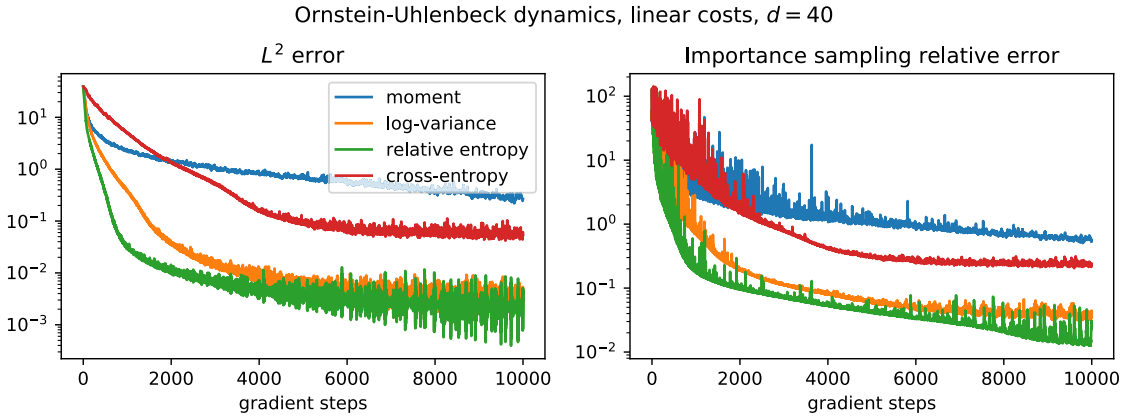


Figure 4.2: Performance of the algorithm using four different loss functions in a high-dimensional setting.

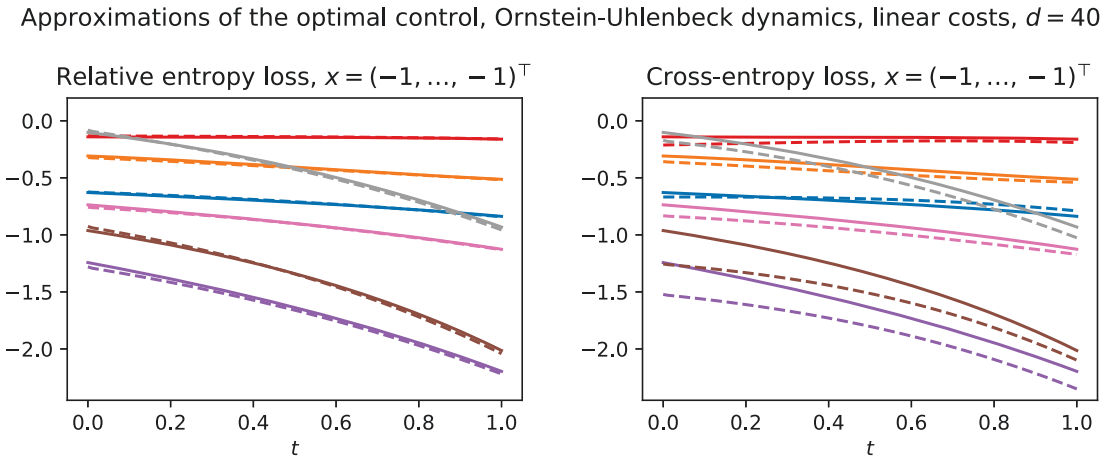


Figure 4.3: Approximation u (dashed lines) and reference solution u^*_{ref} (straight lines) for the optimal control obtained using the relative entropy and cross-entropy losses, respectively. 7 out of the 40 components of u and u^*_{ref} are plotted.

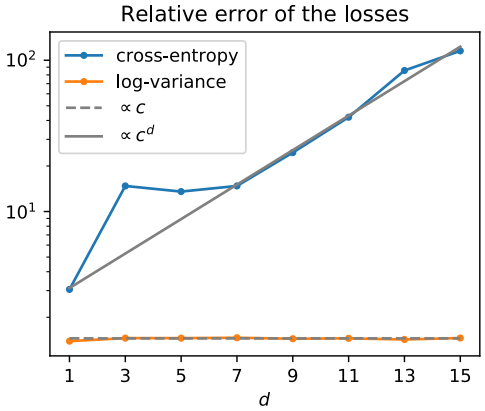


Figure 4.4: Relative error of the log-variance and cross-entropy losses depending on the dimension.

Lastly, let us investigate the effect of the additional parameter y_0 in the moment loss. For a first experiment, we Initialize y_0 with either the naive choice $y_0^{(1)} = 0$, or $y_0^{(2)} = 10$, a starting value which differs considerably from $-\log \mathcal{Z}$ or the optimal choice $y_0^{(3)} = -\log \mathcal{Z} \approx -5.87$. Let us insist that in practical scenarios the value of $-\log \mathcal{Z}$ is usually not known. Additionally, we contrast using Adam and SGD as an optimization routine – in both cases we choose $K = 200$, $\eta = 0.01$, $\Delta t = 0.01$, and the same DenseNet architecture as in the previous experiments.
Figure 4.5 shows that the initialization of y_0 can have a significant impact on the convergence speed. Indeed, with the initialization $y_0 = -\log \mathcal{Z}$, the moment and log-variance losses perform very similarly, in accordance with Proposition 4.22. In contrast, choosing the initial value y_0 such that the discrepancy $|y_0 + \log \mathcal{Z}|$ is large incurs a much slower convergence.
Comparing the two plots in Figure 4.5 shows that the Adam optimizer achieves a much faster convergence overall in comparison to SGD. Moreover, the difference in performance between y_0-initializations is more pronounced when the Adam optimizer is used. The observations in these experiments are in agreement with those in [50].

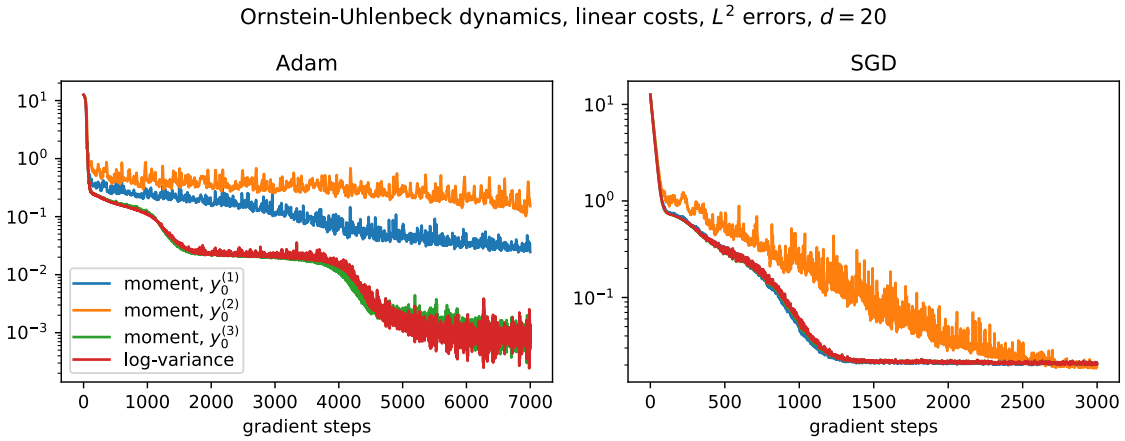


Figure 4.5: Performance of the algorithm with the moment loss and different initializations for y_0, using Adam and SGD.

4.4.3 Ornstein-Uhlenbeck dynamics with quadratic costs

We consider the Ornstein-Uhlenbeck process described by (4.68) with quadratic running and terminal costs, i.e. $f(x,s) = x^\top Px$ and $g(x) = x^\top Rx$, with $P, R \in \mathbb{R}^{d\times d}$. This setting is known as the *linear quadratic Gaussian control* problem [289]. The optimal control is given by [289, Section 6.5]

$$u^*(x,t) = -2B_t^\top F_t\, x, \tag{4.70}$$

where the matrices F_t fulfill the matrix Riccati equation

$$\frac{\mathrm{d}}{\mathrm{d}t}F_t + A_t^\top F_t + F_t A_t - 2F_t B_t B_t^\top F_t + P = 0, \qquad F_T = R. \tag{4.71}$$

In this example, we demonstrate an approach leveraging a priori knowledge about the structure of the solution. Motivated by (4.70), we consider the linear ansatz functions

$$u(x,t_n) = \Upsilon_n x, \tag{4.72}$$

where the entries of the matrices $\Upsilon_n \in \mathbb{R}^{d\times d}$, $n = 0,\dots,N-1$ represent the parameters to be learnt. The matrices A and B are chosen as in Subsection 4.4.2 and we set $P = \frac{1}{2}\mathrm{Id}_{d\times d}$, $R = \mathrm{Id}_{d\times d}$ and $T = 0.5$. Figure 4.6 shows the performance using Adam with learning rate $\eta = 0.001$ and SGD with learning rate $\eta = 0.01$, respectively. The relative entropy losses converges fastest, followed by the log-variance loss. The convergence of the cross-entropy loss is significantly slower, in particular in the SGD case. We also note that the cross-entropy loss diverges if larger learning rates are used. These findings are in line with the results from Proposition 4.29. When SGD is used, the moment loss experiences fluctuations in later gradient steps. This can be explained by the fact that the moment loss is robust at u^* only if $y_0 = -\log \mathcal{Z}$ is satisfied exactly (see Proposition 4.22).

Let us illustrate the potential benefit of sampling X_0 from a prescribed density (see Remark 1.4), here $X_0 \sim \mathcal{N}(0, \mathrm{Id}_{d\times d})$. The overall convergence is hardly affected and the L^2 error dynamics agrees qualitatively with the one shown in Figure 4.6. However, the approximation is more accurate at initial time $t = 0$, see Figure 4.7. This phenomenon appears to be particularly pronounced in this example, as independent ansatz functions are used at each time step.

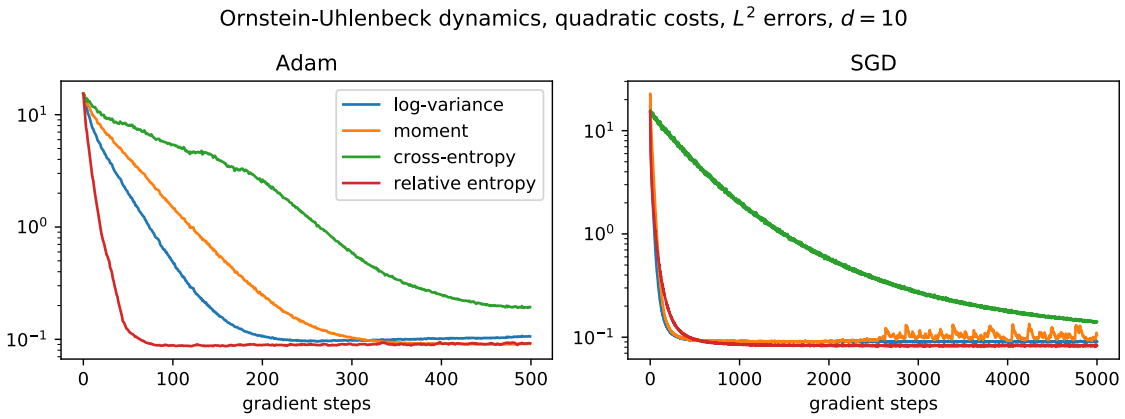


Figure 4.6: Performance of the losses for the Ornstein-Uhlenbeck process with quadratic costs, using Adam and SGD.

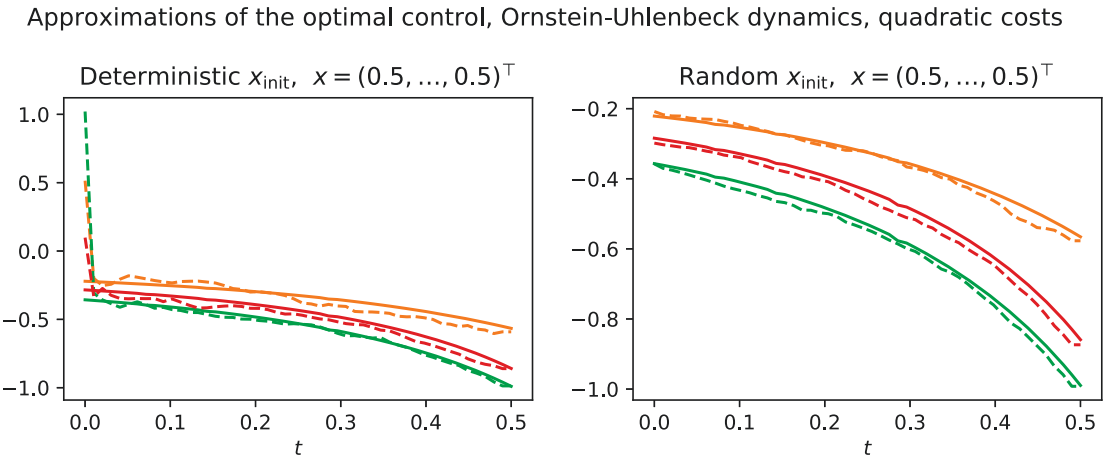


Figure 4.7: Approximation and reference solution of the optimal control with either deterministic or random initializations of x_{init}. Three components of u and u^*_{ref} are plotted.

4.4.4 Metastable dynamics in low and high dimensions

We now come back to the double well potential from Example 1.1 and consider the SDE

$$\mathrm{d}X_s = -\nabla\Psi(X_s)\,\mathrm{d}s + B\,\mathrm{d}W_s, \qquad X_0 = x_{\text{init}}, \tag{4.73}$$

where $B \in \mathbb{R}^{d\times d}$ is the diffusion coefficient, $\Psi(x) = \sum_{i=1}^d \kappa_i(x_i^2 - 1)^2$ is the potential (with $\kappa_i > 0$ being a set of parameters) and $x_{\text{init}} = (-1, \ldots, -1)^\top$ is the initial condition. We consider zero running costs, $f = 0$, terminal costs $g(x) = \sum_{i=1}^d \nu_i(x_i - 1)^2$, where $\nu_i > 0$, and a terminal time $T = 1$. Recall from Example 1.1 that choosing higher values for κ_i and ν_i accentuates the metastable features, making sample-based estimation of $\mathbb{E}\left[\exp(-g(X_T))\right]$ more challenging. For an illustration, Figure 4.8 shows the potential Ψ and the weight at final time e^{-g} (see (1.13)), for different values of ν and κ, in dimension $d = 1$ and for $B = 1$. We furthermore plot the 'optimally tilted potentials' $\Psi^* = \Psi + BB^\top V$, noting that $-\nabla\Psi^* = -\nabla\Psi + Bu^*$. Finally, the right-hand side shows the gradients ∇u^* at initial time $t = 0$.

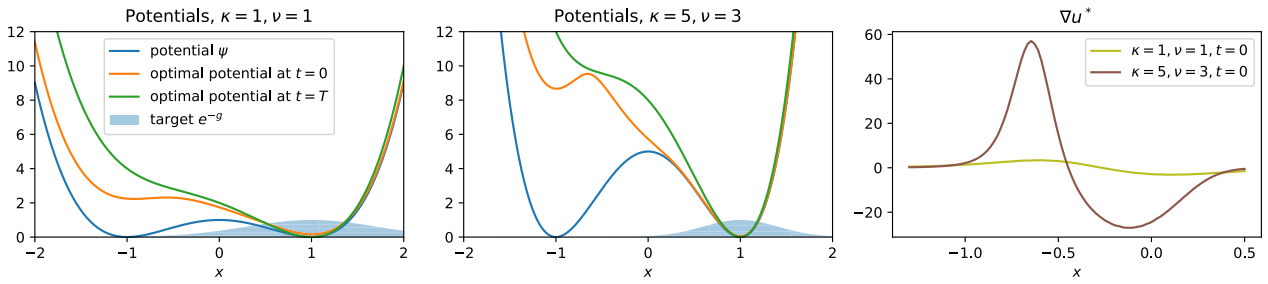


Figure 4.8: The double well potential and the weight e^{-g}, for different values of κ and ν as well as optimal controls (inducing 'tilted potentials') and their gradients.

For an experiment, let us first consider the one-dimensional case, choosing $B = 1$, $\kappa = 5$ and $\nu = 3$. In this setting the relative error associated to the standard Monte Carlo estimator, i.e. the estimator version of (4.66), which we denote by $\widehat{\delta}$, is roughly $\widehat{\delta}(0) = 63.86$ for a batch size of $K = 10^7$ trajectories, from which only about $2 \cdot 10^3$ (i.e. 0.02%) cross the barrier. Given that e^{-g} is supported mostly in the right well, the optimal control u^* steers the dynamics across the barrier. Using an approximation of u^* obtained by a finite difference scheme, we achieve a relative error of $\widehat{\delta}(u^*) = 1.94$ (the theoretical optimum being zero, according to Theorem 1.2) and a crossing ratio of approximately 87.28%.

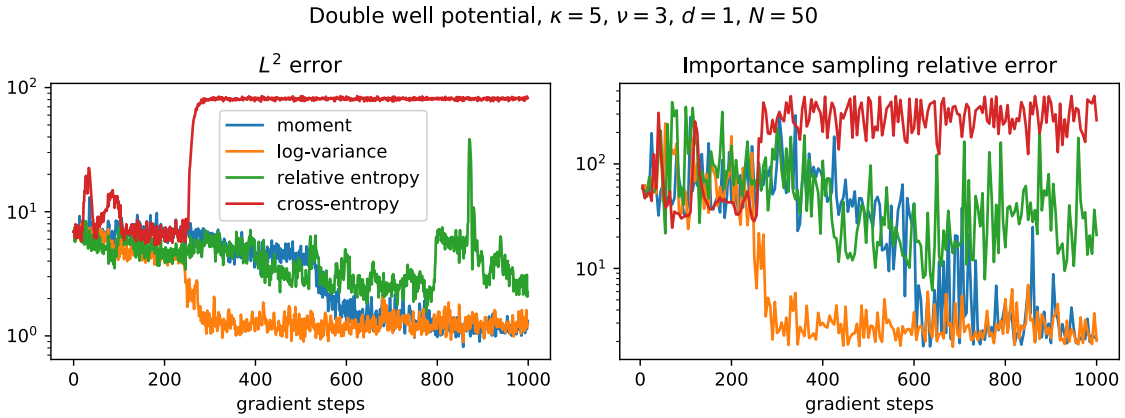


Figure 4.9: Training iterations for the one-dimensional metastable double well example for a small batch size.

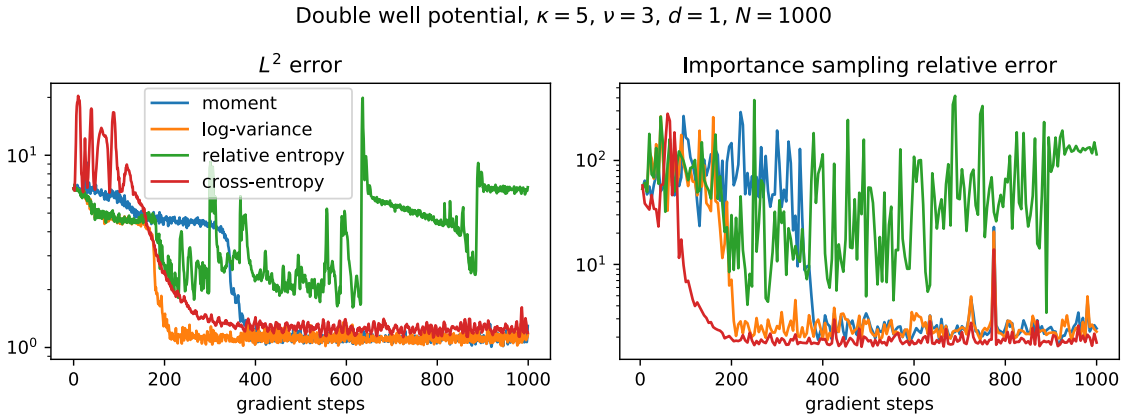


Figure 4.10: Training iterations for the one-dimensional metastable double well example for a large batch size.

To run IDO-based algorithms, we use the standard feed-forward neural network (see Definition 2.49) with the activation function $\varrho = \tanh$ and choose $\Delta t = 0.005$, $\eta = 0.05$. We try batch sizes of $K = 50$ and $K = 1000$ and plot the training progress in Figures 4.9 and 4.10, respectively. In Figure 4.11 we display the approximation obtained using the log-variance loss and compare with the reference solution u^*_{ref}.

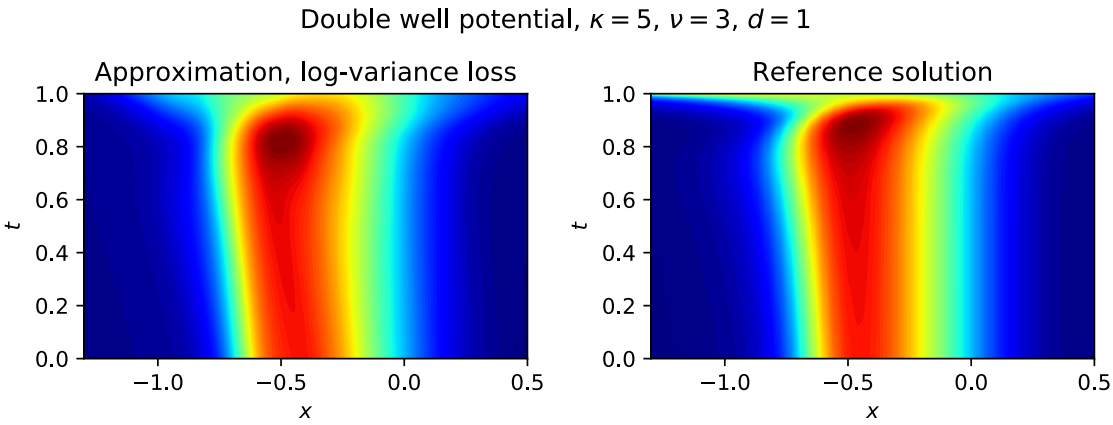


Figure 4.11: Approximation and reference solution for the double well control problem in $d = 1$.

It can be observed that the log-variance and moment losses perform well with both batch sizes, with the log-variance loss however achieving a satisfactory approximation with fewer gradient steps. The cross-entropy loss appears to work well only if the batch size is sufficiently large. We attribute this observation to the non-robustness at u^* (see Proposition 4.25) and, tentatively, to the exponential factor appearing in (4.16b), see Remark 4.10.

The optimization using the relative entropy loss is frustrated by instabilities in the vicinity of the solution u^*. In order to further investigate this aspect we numerically compute the variances of the gradients and the associated relative errors with respect to the mean, using 50 realizations at each gradient step. Figure 4.12 shows the averages of the relative errors and variances over weights in the network[30], confirming that the gradients associated to the log-variance loss have significantly lower variances. This phenomenon is in accordance with Proposition 4.25 (in particular noting that $|\nabla u^*|^2$ is expected to be rather large in a metastable setting, see Figure 4.8) and explains the unsatisfactory behaviour of the relative entropy loss observed in Figures 4.9 and 4.10.

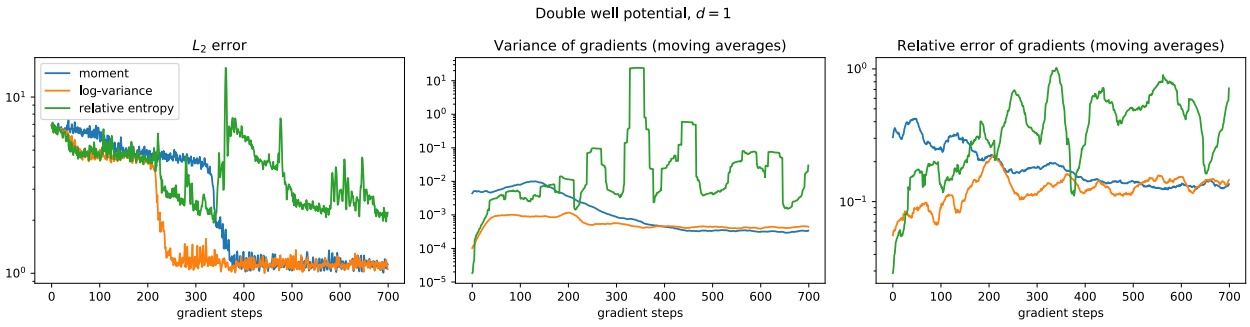


Figure 4.12: We display the L_2 error pertaining to the one-dimensional double well experiment, along with the estimated averages of the variances and relative errors of the gradients along the training iterations for different losses.

Let us now consider the multidimensional setting, namely $d = 10$, where the dynamics exhibits 'highly' metastable characteristics in 3 dimensions and 'weakly' metastable characteristics in the remaining 7 dimensions. To be precise, we set $\kappa_i = 5$, $\nu_i = 3$ for $i \in \{1, 2, 3\}$ and $\kappa_i = 1$, $\nu_i = 1$ for $i \in \{4, \dots, 10\}$. Moreover, we choose the diffusion coefficient to be $B = \mathrm{Id}_{d\times d}$ and conduct the experiment with a batch size of $K = 500$.

In Figure 4.13 we see that only the log-variance loss achieves a reasonable approximation. Interestingly, the training progresses in stages, successively overcoming the potential barriers in the highly metastable directions. On the right-hand side we display the components of the approximated optimal control associated to one highly and one weakly metastable direction, for fixed $t = 0$. We observe that the approximation is fairly accurate, and

[30] In order to lessen the impact of Monte Carlo errors and numerical instabilities, we take moving averages comprising 30 gradient steps and discard partial derivatives with an average magnitude of less than 0.01. We note that the plateaus present in Figure 4.12 are an artefact due to the moving averages, but insist that this procedure does not alter the main results in a qualitative way.

that comparatively large control forces are needed to push the dynamics over the highly metastable potential barrier.

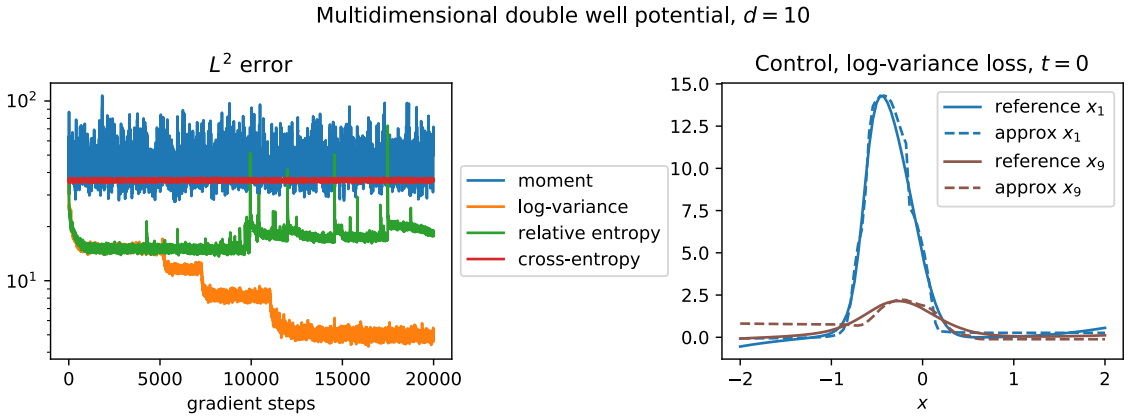


Figure 4.13: Training iterations for the multidimensional metastable double well along with the approximated solution using the log-variance loss, from which we plot two components.

Chapter 5

VarGrad: A low-variance gradient estimator for Bayesian variational inference

In Chapter 4 we have studied variational formulations of certain PDE related problems, which led to the approximation of path space probability measures via minimizing certain divergences. In particular, the idea was to approach a given target by minimizing over a family of alternative measures. Crucially, this led to the novel *log-variance divergence*, for which we have shown remarkable properties. Although we have so far highlighted an application to path space measures, this divergence is valid for any probability measure and therefore can also be applied to densities on $\mathbb{R}^d$, leading to yet other applications. One such application is Bayesian variational inference, where the target measure (or density) is the posterior distribution given by Bayes' theorem. As before, the idea is to approximate this target by minimizing divergences over a family of densities, which is usually done by gradient descent methods in practice. Taking derivatives of a corresponding log-variance loss will then lead to a gradient estimator, which we call *VarGrad*. This chapter is about analyzing VarGrad both theoretically and numerically, ultimately showing its favorable variance properties. In Section 5.1 we will first provide an introduction to Bayesian variational inference, emphasizing why there is indeed need for variance reduction. We note that the used terminology is adapted to the corresponding scientific community and might therefore be slightly different from the other chapters. In Section 5.2 we will introduce VarGrad, which we will subsequently analyze in Section 5.3, in particular showing a closeness to optimally scaled control variate estimators and demonstrating lower variance compared to naive estimators. In Section 5.4 we illustrate these properties on numerical examples.

This chapter is based on joint work with Ayman Boustati, Nikolas Nüsken, Francisco J. R. Ruiz and Ömer Deniz Akyildiz and has been published in [250].

5.1 Background on Bayesian variational inference

We consider the joint density $p(x, z)$ (also called probabilistic model), with $z \in \mathbb{R}^D$ denoting so-called latent variables and $x \in \mathbb{R}^d$ the data, and are interested in computing the posterior $p(z|x) = p(x, z)/p(x)$, where $p(x) = \int p(x, z)\mathrm{d}z$ is the marginal likelihood[31]. For most models of interest, the posterior is hard to compute due to the intractability of the marginal likelihood, and we therefore resort to an approximation. Variational inference approximates the posterior $p(z|x)$ within a parameterized family of distributions $q_\theta(z)$ (with $\theta \in \Theta \subset \mathbb{R}^p$), called variational family[32]. To be precise, it usually finds the parameters θ^* that minimize the KL divergence,

$$\theta^* = \arg\min_{\theta \in \Theta} \mathrm{KL}\left(q_\theta(z)|p(z|x)\right), \tag{5.1}$$

thus, variational inference casts the inference problem as an optimization problem, which can be solved with stochastic optimization tools when the KL divergence is not available in closed form. This optimization problem

[31] Note that by slightly abusing notation we use the same latter for the joint density $p(x, z)$, the posterior $p(z|x)$ and the evidence $p(x)$, however their corresponding arguments should make it clear which object we are referring to. This notation is standard in the machine learning literature on Bayesian variational inference.

[32] Note that we switch the meaning of the letters p and q in order to be consistent with most of the literature on Bayesian variational inference. In contrast to e.g. Chapter 4 now p denotes the target, whereas q the approximating quantity.

is intractable because the KL divergence itself depends on the intractable posterior. Variational inference sidesteps this problem by maximizing instead the *evidence lower bound* (ELBO) given by[33]

$$\mathrm{ELBO}(\theta) = \mathbb{E}_{q_\theta}\left[\log \frac{p(x,z)}{q_\theta(z)}\right], \tag{5.2}$$

which is a lower bound on the marginal likelihood, since

$$\log p(x) = \mathrm{ELBO}(\theta) + \mathrm{KL}\left(q_\theta(z)|p(z|x)\right) \geq \mathrm{ELBO}(\theta). \tag{5.3}$$

Inspecting (5.3), we realize that the maximizer of the ELBO with respect to θ is equivalent to the minimizer of the KL divergence. As the expectation in (5.2) is typically intractable, variational inference uses stochastic optimization to maximize the ELBO. In particular, it forms unbiased Monte Carlo estimators of the gradient $\nabla_\theta \,\mathrm{ELBO}(\theta)$.

In the sequel, we analyze a multi-sample estimator of the gradient of the ELBO. In particular, we focus on an estimator first introduced in [177] and [259], which is based on the so-called score function method [297] with control variates. We first show the connection between this estimator and the log-variance loss as an alternative divergence measure between the variational distribution $q_\theta(z)$ and the exact posterior $p(z|x)$. This divergence, which is different from the standard KL divergence used in variational inference, is defined as the variance, under some arbitrary distribution $r(z)$, of the log-ratio $\log \frac{q_\theta(z)}{p(z|x)}$. As done in the path measure setting in Definition 4.4, we refer to this divergence as the *log-variance loss*. We show that we recover the gradient estimator of [177] and [259] by taking the gradient with respect to the variational parameters θ of the log-variance loss and evaluating the result at $r(z) = q_\theta(z)$. This property suggests a simple algorithm for computing the gradient estimator, based on differentiating through the log-variance loss. We refer to the estimator as *VarGrad*.

We next review the score function method, a Monte Carlo estimator commonly used in variational inference. Instead of the ELBO, we focus on the gradients of the KL divergence with respect to the variational parameters θ directly. These gradients are equal to the gradients of the negative ELBO because the marginal likelihood $p(x)$ does not depend on θ; that is, $\nabla_\theta \,\mathrm{KL}\left(q_\theta(z)|p(z|x)\right) = -\nabla_\theta \,\mathrm{ELBO}(\theta)$.

The score function estimator [45, 226, 244, 297], also known as *Reinforce*, expresses the gradient as an expectation that depends on the log-ratio $\log\left(q_\theta(z)/p(x,z)\right)$ weighted by $\nabla_\theta \log q_\theta(z)$ (known as *score function* in statistics). A formal computation, assuming exchangeability of differentiation and integration, yields

$$\nabla_\theta \,\mathrm{KL}\left(q_\theta(z)|p(z|x)\right) = \int \left(\frac{p(x,z)}{q_\theta(z)}\frac{\nabla_\theta q_\theta(z)}{p(x,z)}q_\theta(z) + \log\left(\frac{q_\theta(z)}{p(x,z)}\right)\frac{\nabla_\theta q_\theta(z)}{q_\theta(z)}q_\theta(z)\right)\mathrm{d}z \tag{5.4a}$$

$$= \nabla_\theta \int q_\theta(z)\mathrm{d}z + \int \log\left(\frac{q_\theta(z)}{p(x,z)}\right)\nabla_\theta \log\left(q_\theta(z)\right) q_\theta(z)\mathrm{d}z, \tag{5.4b}$$

where the first summand vanishes due to $\nabla_\theta \int q_\theta(z)\mathrm{d}z = \nabla_\theta(1) = 0$. This brings the resulting estimator

$$\nabla_\theta \,\mathrm{KL}\left(q_\theta(z)|p(z|x)\right) \approx \widehat{g}_{\mathrm{Reinforce}}(\theta) = \frac{1}{K}\sum_{k=1}^{K} \log\left(\frac{q_\theta(z^{(k)})}{p(x,z^{(k)})}\right)\nabla_\theta \log q_\theta(z^{(k)}), \tag{5.5}$$

where $z^{(k)} \overset{\text{i.i.d.}}{\sim} q_\theta(z)$. It is often observed in practice that this estimator suffers from high variance and therefore additional tricks are needed; [244] for instance suggests to use Rao-Blackwellization, which exploits a potential factorization of the variational distribution, as well as control variates. The idea of the latter is to add a quantity with known expectation that might correlate with the estimator in such a way that the overall variance is reduced. Here we rely on the score function for being this quantity, i.e. we consider the control variate $a \odot \nabla_\theta \log q_\theta(z)$, where $\odot$ denotes the Hadamard (elementwise) product and the idea is to choose the coefficient $a \in \mathbb{R}^p$ as to decrease the estimator's variance, noting that

$$\mathbb{E}\left[\nabla_\theta \log q_\theta(z)\right] = \int \frac{\nabla_\theta q_\theta(z)}{q_\theta(z)} q_\theta(z)\mathrm{d}z = 0. \tag{5.6}$$

We get the unbiased control variate gradient estimator

$$\widehat{g}_{\mathrm{CV}}(\theta) = \widehat{g}_{\mathrm{Reinforce}}(\theta) - a \odot \left(\frac{1}{K}\sum_{k=1}^{K} \nabla_\theta \log q_\theta(z^{(k)})\right). \tag{5.7}$$

It remains the question of how to choose a in practice and we note that there is a theoretical optimal choice yielding minimal variance.

[33] As a further remark on the notation, note that the subscript after the expectation or variance operator indicates the density w.r.t. to which the random variable is drawn, e.g. $\mathbb{E}_q[f(z)] = \int_{\mathbb{R}^d} f(z)q(z)\mathrm{d}z$, cf. footnote [19] on page 50.

Lemma 5.1. *Let $f_\theta(z) := \log \frac{q_\theta(z)}{p(x,z)}$. For each $i \in \{1, \dots, p\}$, the i-th component of the optimal control variate scaling is given by*

$$a_i^* = \frac{\mathrm{Cov}_{q_\theta}\left(f_\theta \partial_{\theta_i} \log q_\theta, \partial_{\theta_i} \log q_\theta\right)}{\mathrm{Var}_{q_\theta}\left(\partial_{\theta_i} \log q_\theta\right)}. \tag{5.8}$$

Proof. For each i we compute

$$\mathrm{Var}_{q_\theta}\left(\widehat{g}_{\mathrm{CV},i}(\theta)\right) = \mathrm{Var}_{q_\theta}\left(\widehat{g}_{\mathrm{Reinforce},i}(\theta)\right) + \frac{{a_i}^2}{S}\mathrm{Var}_{q_\theta}\left(\partial_{\theta_i} \log q_\theta\right) - \frac{2a_i}{S}\mathrm{Cov}_{q_\theta}\left(f_\theta \partial_{\theta_i} \log q_\theta, \partial_{\theta_i} \log q_\theta\right), \tag{5.9}$$

which is a parabola in a_i that opens up. One therefore readily computes the minimal value to be $a_i = a_i^*$ as specified in (5.8). □

For the estimation of $\nabla_\theta \mathrm{KL}\left(q_\theta(z)|p(z|x)\right)$ [177] and [259] leverage the multi-sample estimator by using $K-1$ samples to compute a particular control variate coefficient a and then average over the resulting estimators, which they call *leave-one-out estimator*,

$$\widehat{g}_{\mathrm{LOO}}(\theta) = \frac{1}{K}\sum_{k=1}^{K} \nabla_\theta \log q_\theta(z^{(k)}) \left(f_\theta(z^{(k)}) - \sum_{j \neq k}^{K} f_\theta(z^{(j)}) \right) \tag{5.10a}$$

$$= \frac{1}{K-1}\left(\sum_{k=1}^{K} f_\theta(z^{(k)}) \nabla_\theta \log q_\theta(z^{(k)}) - \bar{f}_\theta \sum_{k=1}^{K} \nabla_\theta \log q_\theta(z^{(k)}) \right), \tag{5.10b}$$

where $z^{(k)} \overset{\text{i.i.d.}}{\sim} q_\theta(z)$ and for simplicity of notation we have defined

$$f_\theta(z) := \log \frac{q_\theta(z)}{p(x,z)} \quad \text{and} \quad \bar{f}_\theta := \frac{1}{K}\sum_{k=1}^{K} f_\theta(z^{(k)}) \approx -\,\mathrm{ELBO}(\theta). \tag{5.11}$$

We note that this method makes no assumptions on the model $p(x,z)$ or the distribution $q_\theta(z)$; the only requirements are to be able to sample from $q_\theta(z)$ and to evaluate $\log q_\theta(z)$ and $\log p(x,z)$.

5.2 The log-variance loss and its connection to VarGrad

We now show the connection between the leave-one-out estimator (5.10) and the log-variance loss. We will introduce the log-variance loss for densities and will refer to the estimator (5.10) as *VarGrad*.

The log-variance loss

In analogy to the path space measure setting, the log-variance loss for densities is defined as the variance, under some arbitrary distribution $r(z)$, of the log-ratio $\log \frac{q_\theta(z)}{p(z|x)}$. It has the property of reproducing the gradients of the KL divergence under certain conditions (see Proposition 5.3 for details). We next give the precise definition of the loss (completely analogous to Definition 4.4).

Definition 5.2. For a given distribution $r(z)$, the log-variance loss $\mathcal{L}_r(\cdot)$ is given by

$$\mathcal{L}_r(q_\theta(z)|p(z|x)) = \frac{1}{2}\mathrm{Var}_r\left(\log\left(\frac{q_\theta(z)}{p(z|x)}\right)\right). \tag{5.12}$$

We refer to the distribution $r(z)$ as the *reference distribution* under which the discrepancy between $q_\theta(z)$ and the posterior $p(z|x)$ is computed. When the support of the reference distribution contains the supports of $q_\theta(z)$ and $p(z|x)$, (5.12) is a divergence[34]; it is zero if and only if $q_\theta(z) = p(z|x)$. The factor $1/2$ in (5.12) is only included because it simplifies some expressions later in this section[35].

We next show that the gradient of the log-variance loss and the gradient of the standard KL divergence coincide under certain conditions. In particular, taking the gradient of (5.12) with respect to the variational parameters *θ and then* evaluating the result for a reference distribution $r(z) = q_\theta(z)$ gives the gradient of the KL divergence. This property is detailed in Proposition 5.3.

[34] More technically, as we assume that $r(z)$, $p(z|x)$, and $q_\theta(z)$ admit densities such that measure-zero sets of $r(z)$ are necessarily measure-zero sets of $p(z|x)$ and $q_\theta(z)$, implying that the divergence is well defined.

[35] Note that compared to Definition 4.4 we introduced a factor $\frac{1}{2}$ in the definition of the log-variance loss in order to relate the corresponding gradient estimator (5.19) to the *Reinforce* estimator (5.5).

Proposition 5.3. *The gradient with respect to θ of the log-variance loss, evaluated at $r(z) = q_\theta(z)$, equals the gradient of the KL divergence,*

$$\nabla_\theta \mathcal{L}_r(q_\theta(z)|p(z|x))\Big|_{r=q_\theta} = \nabla_\theta \,\mathrm{KL}(q_\theta(z)|p(z|x)). \tag{5.13}$$

Proof. We first consider the gradient of the KL divergence. It is given by

$$\nabla_\theta \,\mathrm{KL}(q_\theta(z)|p(z|x)) = \int \nabla_\theta q_\theta(z)\,\mathrm{d}z + \int \log\left(\frac{q_\theta(z)}{p(z|x)}\right)\nabla_\theta q_\theta(z)\,\mathrm{d}z, \tag{5.14}$$

where we can drop the first term since $\int \nabla_\theta q_\theta(z)\,\mathrm{d}z = \nabla_\theta \int q_\theta(z)\,\mathrm{d}z = \nabla_\theta(1) = 0$.

We now consider the gradient of the log-variance loss. Using the definition from (5.12), we see that

$$\nabla_\theta \mathcal{L}_r(q_\theta(z)|p(z|x)) = \frac{1}{2}\nabla_\theta \int \log^2\left(\frac{q_\theta(z)}{p(z|x)}\right) r(z)\mathrm{d}z - \frac{1}{2}\nabla_\theta \left(\int \log\left(\frac{q_\theta(z)}{p(z|x)}\right) r(z)\mathrm{d}z\right)^2 \tag{5.15a}$$

$$= \int \log\left(\frac{q_\theta(z)}{p(z|x)}\right)\frac{\nabla_\theta q_\theta(z)}{q_\theta(z)} r(z)\mathrm{d}z - \left(\int \log\left(\frac{q_\theta(z)}{p(z|x)}\right) r(z)\mathrm{d}z\right)\left(\int \frac{\nabla_\theta q_\theta(z)}{q_\theta(z)} r(z)\mathrm{d}z\right). \tag{5.15b}$$

When we evaluate the gradient at $r(z) = q_\theta(z)$, the right-most term vanishes, since

$$\int \frac{\nabla_\theta q_\theta(z)}{r(z)} r(z)\,\mathrm{d}z = \int \nabla_\theta q_\theta(z)\,\mathrm{d}z = 0. \tag{5.16}$$

Thus, the gradient of the log-variance loss becomes equal to the gradient of the KL divergence. □

Proposition 5.3 implies that we can estimate the gradient of the KL divergence by estimating instead the gradient of the log-variance loss.

Remark 5.4. The result in Proposition 5.3 is obtained by setting $r(z) = q_\theta(z)$ *after* taking the gradient with respect to θ. The same result does not hold if we set $r(z) = q_\theta(z)$ before differentiating.

VarGrad: Derivation of the gradient estimator from the log-variance loss

The leave-one-out estimator in (5.10) [177, 259] is connected to the log-variance loss from above through Proposition 5.3. Firstly, note that the log-variance loss is intractable as it depends on the posterior $p(z|x)$. However, since the marginal likelihood $p(x)$ has zero variance, it can be dropped from the definition in (5.12) yielding

$$\mathcal{L}_r(q_\theta(z)|p(z|x)) = \frac{1}{2}\,\mathrm{Var}_r\left(\log\left(\frac{q_\theta(z)}{p(x,z)}\right)\right) = \frac{1}{2}\,\mathrm{Var}_r\left(f_\theta(z)\right), \tag{5.17}$$

where $f_\theta(z)$ is defined in (5.11).

Next, we can build the estimator of the log-variance loss as the empirical variance of K Monte Carlo samples,

$$\mathcal{L}_r(q_\theta(z)|p(z|x)) \approx \frac{1}{2(K-1)}\sum_{k=1}^{K}\left(f_\theta(z^{(k)}) - \bar{f}_\theta\right)^2, \quad z^{(k)} \overset{\text{i.i.d.}}{\sim} r(z). \tag{5.18}$$

Applying Proposition 5.3 by differentiating through (5.18), we can arrive at the VarGrad estimator, $\widehat{g}_{\text{VarGrad}}(\theta) = \widehat{g}_{\text{LOO}}(\theta) \approx \nabla_\theta \,\mathrm{KL}\left(q_\theta(z)|p(z|x)\right)$, where

$$\widehat{g}_{\text{VarGrad}}(\theta) = \frac{1}{K-1}\left(\sum_{k=1}^{K} f_\theta(z^{(k)})\nabla_\theta \log q_\theta(z^{(k)}) - \bar{f}_\theta \sum_{k=1}^{K}\nabla_\theta \log q_\theta(z^{(k)})\right), \tag{5.19}$$

and $z^{(k)} \overset{\text{i.i.d.}}{\sim} q_\theta(z)$.

The expression for VarGrad in (5.19) is identical to that of the leave-one-out estimator in (5.10) and it can therefore (more or less) be interpreted as a control variate estimator with a particular choice of a. Furthermore, VarGrad is an unbiased estimator of the gradient of the KL divergence (and equivalently the gradient of the ELBO). From a probabilistic programming perspective, setting the reference $r(z) = q_\theta(z)$ *after* differentiating w.r.t. θ amounts to sampling $z^{(k)} \sim q_\theta(z)$ and detaching the resulting samples from the computational graph.

This suggests a novel algorithmic procedure, given in Algorithm 2. Its implementation is simple: we only need the samples $z^{(k)} \sim q_\theta(z)$ and apply the `stop_gradient` operator (assuring that derivatives are not taken with respect to $r(z)$), evaluate the log-ratio $f_\theta(z^{(k)})$ for each sample, and then differentiate through the empirical variance of this log-ratio.

Algorithm 2: Pseudocode for VarGrad

Input: Variational parameters θ, data x, sample size K.
for $k = 1$ **to** K **do**
  $z^{(k)} \leftarrow \texttt{sample}(q_\theta(\cdot))$ (sample from the approximate posterior)
  $z^{(k)} \leftarrow \texttt{stop\_gradient}(z^{(k)})$ (detach the samples from the computational graph)
  $f_\theta^{(k)} \leftarrow \log q_\theta(z^{(k)}) - \log p(x, z^{(k)})$ (an estimate of the negative ELBO)
end
$\widehat{\mathcal{L}} \leftarrow \frac{1}{2}\,\mathrm{Var}(\{f_\theta^{(k)}\}_{k=1}^K)$ (an estimate of the log-variance loss)
Result: Gradient of $\widehat{\mathcal{L}}$ (differentiate through the loss w.r.t. θ)

5.3 Analytical results for VarGrad

We now study some properties of $\widehat{g}_{\mathrm{VarGrad}}$ in comparison to other estimators based on the score function method. We analyze the difference δ^{CV} between the control variate coefficient of VarGrad (called a^{VarGrad}) and the optimal one stated in Lemma 5.1. The former can be approximated cheaply and unbiasedly, while a standard Monte Carlo estimator of the latter is biased and often exhibits high variance. Furthermore, we establish that the difference δ^{CV} is negligible in certain settings, in particular when $\mathrm{KL}(q_\theta(z)|p(z|x))$ is either very large or close to zero; thus in these settings the control variate coefficient of VarGrad is close to the optimal coefficient. Later we show that a simple relation between δ^{CV} and the ELBO is sufficient to *guarantee* that $\widehat{g}_{\mathrm{VarGrad}}$ has lower variance than $\widehat{g}_{\mathrm{Reinforce}}$ when the number of Monte Carlo samples is large enough.

Analysis of the control variate coefficients

As mentioned before, in [244] it is proposed to modify $\widehat{g}_{\mathrm{Reinforce}}$ using a score function control variate yielding (5.7),

$$\widehat{g}_{\mathrm{CV}}(\theta) = \widehat{g}_{\mathrm{Reinforce}}(\theta) - a \odot \left(\frac{1}{K} \sum_{k=1}^{K} \nabla_\theta \log q_\theta(z^{(k)}) \right), \tag{5.20}$$

where a is a vector chosen so as to reduce the variance of the estimator. We recover VarGrad as in (5.19), up to a factor of proportionality, by setting the control variate coefficient $a = \bar{f}_\theta \mathbf{1}$ in (5.7), where $\mathbf{1}$ is a vector of ones. The proportionality relation is $\frac{K-1}{K}\widehat{g}_{\mathrm{CV}} = \widehat{g}_{\mathrm{VarGrad}}$. In terms of variance reduction, we recall the optimal coefficient a^* from Lemma 5.1.

We next show that the coefficients of VarGrad, a^{VarGrad}, are close to the optimal coefficients a^*. For this, we first relate a^{VarGrad} to a^* in Lemma 5.5.

Lemma 5.5. *We can write the optimal control variate coefficient as the expected value of* a^{VarGrad} *plus a* control variate correction *term* δ^{CV}*, i.e.,*

$$a^* = \mathbb{E}_{q_\theta}[a^{\mathrm{VarGrad}}] + \delta^{\mathrm{CV}} = -\mathrm{ELBO}(\theta) + \delta^{\mathrm{CV}}, \tag{5.21}$$

where $a^{\mathrm{VarGrad}} = \bar{f}_\theta$ *and the components of the correction term* δ^{CV} *are given by*

$$\delta_i^{\mathrm{CV}} = \frac{\mathrm{Cov}_{q_\theta}\left(f_\theta, (\partial_{\theta_i} \log q_\theta)^2\right)}{\mathrm{Var}_{q_\theta}\left(\partial_{\theta_i} \log q_\theta\right)}. \tag{5.22}$$

Proof. First, notice that $\mathrm{Var}_{q_\theta}\left(\partial_{\theta_i} \log q_\theta\right) = \mathbb{E}_{q_\theta}[(\partial_{\theta_i} \log q_\theta)^2]$ since $\mathbb{E}_{q_\theta}[\partial_{\theta_i} \log q_\theta] = 0$. We then compute

$$a_i^* = \frac{\mathbb{E}_{q_\theta}\left[f_\theta (\partial_{\theta_i} \log q_\theta)^2\right]}{\mathbb{E}_{q_\theta}\left[\left(\partial_{\theta_i} \log q_\theta\right)^2\right]} \tag{5.23a}$$

$$= \frac{\mathbb{E}_{q_\theta}\left[f_\theta (\partial_{\theta_i} \log q_\theta)^2\right] - \mathbb{E}_{q_\theta}\left[f_\theta\right] \mathbb{E}_{q_\theta}\left[(\partial_{\theta_i} \log q_\theta)^2\right] + \mathbb{E}_{q_\theta}\left[f_\theta\right] \mathbb{E}_{q_\theta}\left[(\partial_{\theta_i} \log q_\theta)^2\right]}{\mathbb{E}_{q_\theta}\left[\left(\partial_{\theta_i} \log q_\theta\right)^2\right]} \tag{5.23b}$$

$$= \mathbb{E}_{q_\theta}[\bar{f}_\theta] + \delta_i^{\mathrm{CV}}. \tag{5.23c}$$

In the last line we have used the fact that $\mathbb{E}_{q_\theta}[f_\theta] = \mathbb{E}_{q_\theta}[\bar{f}_\theta]$. □

According to Lemma 5.5, the difference between the optimal control variate coefficient and the (expected) VarGrad coefficient is equal to the correction δ^{CV}. We hypothesize that direct Monte Carlo estimation of δ^{CV} in (5.22) (or similarly for (5.8)) suffers from high variance because it takes the form of a fraction[36] (see for instance Section 5.4.2). Moreover, estimating (5.22) by taking the ratio of two Monte Carlo estimators gives a biased estimate. Furthermore, estimating δ_i^{CV} with the same samples as the ones used for estimating the score function yields a biased gradient estimator [106, Section 4].

We next show that in certain settings the correction term δ^{CV} becomes negligible, implying that $\widehat{g}_{\mathrm{VarGrad}}$ and $\widehat{g}_{\mathrm{Reinforce}}$ equipped with the optimal control variate coefficients behave almost identically. We provide empirical evidence of this finding in Section 5.4 (and in Section 5.4.2 for the Gaussian case).

Proposition 5.6 (δ^{CV} is small in comparison to $\mathbb{E}_{q_\theta}[a^{\mathrm{VarGrad}}]$ if the KL divergence between $q_\theta(z)$ and $p(z|x)$ is large or small)**.** *Assume that $q_\theta(z)$ has lighter tails than the posterior $p(z|x)$, in the sense that there exists a constant $C > 0$ such that*

$$\sup_z \frac{q_\theta(z)}{p(z|x)} < C. \tag{5.24}$$

Furthermore, define the kurtosis of the score function,

$$\mathrm{Kurt}[\partial_{\theta_i} \log q_\theta] = \frac{\mathbb{E}_{q_\theta}[(\partial_{\theta_i} \log q_\theta)^4]}{(\mathbb{E}_{q_\theta}[(\partial_{\theta_i} \log q_\theta)^2])^2}, \tag{5.25}$$

and assume that it is bounded, $\mathrm{Kurt}[\partial_{\theta_i} \log q_\theta] < \infty$. Then, the ratio between the control variate correction δ^{CV} and the expected control variate coefficient of VarGrad can be upper bounded by

$$\left|\frac{\delta_i^{\mathrm{CV}}}{\mathbb{E}_{q_\theta}[a^{\mathrm{VarGrad}}]}\right| \leq \frac{2\sqrt{C\,\mathrm{Kurt}[\partial_{\theta_i} \log q_\theta]}}{\left|\sqrt{\mathrm{KL}(q_\theta(z)|p(z|x))} - \frac{\log p(x)}{\sqrt{\mathrm{KL}(q_\theta(z)|p(z|x))}}\right|}. \tag{5.26}$$

Proof. Note that

$$\left|\frac{\delta_i^{\mathrm{CV}}}{\mathbb{E}_{q_\theta}[a^{\mathrm{VarGrad}}]}\right| = \left|\frac{\mathrm{Cov}_{q_\theta}\left(f_\theta, (\partial_{\theta_i} \log q_\theta)^2\right)}{\mathbb{E}_{q_\theta}[f_\theta]\,\mathrm{Var}_{q_\theta}\left(\partial_{\theta_i} \log q_\theta\right)}\right| = \left|\frac{\mathbb{E}_{q_\theta}\left[(f_\theta - \mathbb{E}_{q_\theta}[f_\theta])\,(\partial_{\theta_i} \log q_\theta)^2\right]}{\mathbb{E}_{q_\theta}[f_\theta]\mathbb{E}_{q_\theta}\left[(\partial_{\theta_i} \log q_\theta)^2\right]}\right|, \tag{5.27}$$

where we have used the fact that $\mathbb{E}_{q_\theta}[\partial_{\theta_i} \log q_\theta] = 0$. From

$$\mathbb{E}[f_\theta] = -\mathrm{ELBO}(\theta) = \mathrm{KL}(q_\theta(z)|p(z|x)) - \log p(x),$$

and using the Cauchy-Schwarz inequality, (5.27) can be bounded from above by

$$\frac{\left(\mathrm{Var}_{q_\theta}\left(\log \frac{q_\theta(z)}{p(z|x)}\right)\right)^{1/2}}{|\mathrm{KL}(q_\theta(z)|p(z|x)) - \log p(x)|}\left(\frac{\mathbb{E}_{q_\theta}[(\partial_{\theta_i} \log q_\theta)^4]}{(\mathbb{E}_{q_\theta}[(\partial_{\theta_i} \log q_\theta)^2])^2}\right)^{1/2}. \tag{5.28}$$

The second factor equals $\sqrt{\mathrm{Kurt}[\partial_{\theta_i} \log q_\theta]}$. To bound the first factor, notice that

$$\left(\mathrm{Var}_{q_\theta}\left(\log \frac{q_\theta(z)}{p(z|x)}\right)\right)^{1/2} \leq \left(\mathbb{E}_{q_\theta}\left[\log^2 \frac{q_\theta(z)}{p(z|x)}\right]\right)^{1/2} = \left(\mathbb{E}_{q_\theta}\left[\log^2 \frac{p(z|x)}{q_\theta(z)}\right]\right)^{1/2} \tag{5.29a}$$

$$\leq \left(2\mathbb{E}_{q_\theta}\left[\exp\left(\left|\log \frac{p(z|x)}{q_\theta(z)}\right|\right) - 1 - \left|\log \frac{p(z|x)}{q_\theta(z)}\right|\right]\right)^{1/2}, \tag{5.29b}$$

where we have used the estimate

$$e^x - 1 - x = \sum_{n=0}^{\infty} \frac{x^n}{n!} - 1 - x = \sum_{n=2}^{\infty} \frac{x^n}{n!} \geq \frac{1}{2}x^2, \quad x \geq 0, \tag{5.30}$$

with $x = \left|\log \frac{p(z|x)}{q_\theta(z)}\right|$. We now use [103, Lemma 8.3] to bound (5.29b) from above by

$$2\sqrt{C}h(q_\theta(z) \mid p(z|x)), \tag{5.31}$$

[36] Monte Carlo estimators of fractions are not straightforward. As a simple example, consider the ratio of two independent Gaussian random variables, each with zero mean and unit variance. The ratio follows a Cauchy distribution, which has infinite variance.

where

$$h(q_\theta(z)|p(z|x)) = \sqrt{\int \left(\sqrt{q_\theta(z)} - \sqrt{p(z|x)}\right)^2 \mathrm{d}z} \tag{5.32}$$

is the Hellinger distance. From [247, Lemma A.3.5] we have the bound $h(q_\theta(z) \mid p(z|x)) \leq \sqrt{\mathrm{KL}(q_\theta(z) \mid p(z|x))}$. Combining these estimates we arrive at the claimed result. □

Remark 5.7. The variational approximation $q_\theta(z)$ typically underestimates the spread of the posterior $p(z|x)$ [36], and so the assumption in (5.24) is typically satisfied in practice after a few iterations of the optimization algorithm. The kurtosis $\mathrm{Kurt}[\partial_{\theta_i} \log q_\theta]$ quantifies the weight of the tails of the variational approximation in terms of the score function. In Section 5.4.1 we analyze the kurtosis of exponential family distributions and show that it is uniformly bounded for Gaussian variational families.

Remark 5.8. The upper bound in (5.26) allows us to identify two regimes. When $\mathrm{KL}(q_\theta(z)|p(z|x))$ is large, the bound asserts that the relative error satisfies

$$\left|\frac{\delta_i^{\mathrm{CV}}}{\mathbb{E}_{q_\theta}[a^{\mathrm{VarGrad}}]}\right| \lessapprox \mathcal{O}\left(\mathrm{KL}(q_\theta(z)|p(z|x))^{-1/2}\right), \tag{5.33}$$

as the second term in the denominator of (5.26) becomes negligible. This can happen in the early stages of the optimization process, in which case we can conclude that δ^{CV} is expected to be small. Since the KL divergence increases with the dimensionality of the latent variable z (see Appendix B.7), (5.26) also implies that the ratio becomes smaller as the number of latent variables grows. Moreover, if the *minimum* KL divergence between the variational family and the true posterior is large (i.e., if the best candidate in the variational family is still far away from the target), the correction term δ_i^{CV} can be negligible during the whole optimization procedure, which is often the case in practice.
In the regime where $\mathrm{KL}(q_\theta(z)|p(z|x))$ approaches zero (i.e., towards the end of the optimization process if the variational family is well specified and includes the posterior), then (5.26) implies that

$$\left|\frac{\delta_i^{\mathrm{CV}}}{\mathbb{E}_{q_\theta}[a^{\mathrm{VarGrad}}]}\right| \lessapprox \mathcal{O}\left(\mathrm{KL}(q_\theta(z)|p(z|x))^{1/2}\right). \tag{5.34}$$

In this regime, the error w.r.t. the optimal control variate coefficient decreases with the KL divergence. The estimates in (5.33) and (5.34) combined suggest that the relative error remains bounded throughout the optimization. We will verify this proposition experimentally in Section 5.4.

Variance of the estimator

Now we provide a result that guarantees that the variance of $\widehat{g}_{\mathrm{VarGrad}}$ is smaller than the variance of $\widehat{g}_{\mathrm{Reinforce}}$ when the number of Monte Carlo samples is large enough.

Proposition 5.9. *Consider the two gradient estimators $\widehat{g}_{\mathrm{Reinforce}}(\theta)$ and $\widehat{g}_{\mathrm{VarGrad}}(\theta)$, each with K Monte Carlo samples, as defined in* (5.5) *and* (5.19)*, respectively. If*

$$-\frac{\delta_i^{\mathrm{CV}}}{\mathbb{E}_{q_\theta}[a^{\mathrm{VarGrad}}]} = \frac{\delta_i^{\mathrm{CV}}}{\mathrm{ELBO}(\theta)} < \frac{1}{2} \tag{5.35}$$

then there exists $K_0 \in \mathbb{N}$ such that

$$\mathrm{Var}\left(\widehat{g}_{\mathrm{VarGrad},i}(\theta)\right) \leq \mathrm{Var}\left(\widehat{g}_{\mathrm{Reinforce},i}(\theta)\right), \qquad \text{for all} \quad K \geq K_0. \tag{5.36}$$

Proof. See Appendix C.4. □

If the correction δ^{CV} is negligible in the sense of Proposition 5.6, then the assumption in (5.35) is satisfied and Proposition 5.9 guarantees that VarGrad has lower variance than Reinforce when K is large enough. We arrive at the following corollary, which also considers the dimensionality of the latent variables. The main assumption that the KL divergence increases with the dimension of the latent space is supported by the result in Appendix B.7.

Corollary 5.10. *Let K be the number of samples and D the dimension of the latent variable z. Furthermore, let the assumptions of Proposition 5.6 be satisfied and assume that $\mathrm{KL}(q_\theta(z)|p(z|x))$ is strictly increasing in D and goes to infinity for $D \to \infty$. Then, there exist $K_0, D_0 \in \mathbb{N}$ such that*

$$\mathrm{Var}\left(\widehat{g}_{\mathrm{VarGrad},i}(\theta)\right) \leq \mathrm{Var}\left(\widehat{g}_{\mathrm{Reinforce},i}(\theta)\right), \qquad \text{for all } K \geq K_0 \text{ and } D \geq D_0. \tag{5.37}$$

Proof. Note that with Proposition 5.6 we have

$$\left| \frac{\delta_i^{\mathrm{CV}}}{\mathbb{E}_{q_\theta}[a^{\mathrm{VarGrad}}]} \right| \to 0 \tag{5.38}$$

for $D \to \infty$, assuming that $\mathrm{KL}(q_\theta(z)|p(z|x))$ is strictly increasing in D. Therefore, for large enough D, the condition from Proposition 5.9 (see (5.35)), is fulfilled and the statement follows immediately. □

We provide further intuition on the condition in (5.35) with the analysis in Section 5.4.2.

Work related to VarGrad

In the last few years, many gradient estimators of the ELBO have been proposed; see [212] for a comprehensive review. Among those, the score function estimators [45, 226, 244, 297] and the reparameterization estimators [168, 249, 280], as well as combinations of both [214, 257], are arguably the most widely used. NVIL [210] and MuProp [120] are unbiased gradient estimators for training stochastic neural networks.

Other gradient estimators are specific for discrete-valued latent variables. The concrete relaxation [154, 204] described a way to form a biased estimator of the gradient, which REBAR [282] and RELAX [116] use as a control variate to obtain an unbiased estimator. Other recent estimators have been proposed by [57, 188, 231, 267, 302, 303], and [74]. In Section 5.4, we compare VarGrad with some of these estimators, showing that it exhibits a favorable performance versus computational complexity trade-off.

The VarGrad estimator was first introduced by [177] and [259]. It also relates to VIMCO [211] in that it is a leave-one-out estimator. In this chapter, we have described an alternative derivation of VarGrad, based on the log-variance loss.

The log-variance loss defines an alternative divergence between the approximate and the exact posterior distributions. In the context of optimal control of diffusion processes and related forward-backward stochastic differential equations, it arises naturally to quantify the discrepancy between measures on path space, see Chapter 4. Other forms of alternative divergences have also been explored in previous work; for example the χ^2-divergence [70], the Rényi divergence [194], the Langevin-Stein [243], the α-divergence [138], other f-divergences [292], a contrastive divergence [256], and also the inclusive KL [215], see also Section 5.4.3.

5.4 Numerical experiments for VarGrad

In order to verify the properties of VarGrad empirically, we test it on two popular models: a Bayesian logistic regression on a synthetic dataset and a discrete variational autoencoder (DVAE) [168, 258] on a fixed binarization of Omniglot [186]. Let us first explain the two different experiments in detail[37].

In the Bayesian logistic regression we define the discrete likelihood for $x \in \{0, 1\}$ to be the probability mass function of a Bernoulli random variable, i.e.

$$p(x|z) = b(x; \sigma(Yz)) \tag{5.39}$$

with (the componentwise application)

$$b(x; \xi) := \xi^x (1 - \xi)^{1-x}, \qquad \sigma(y) = \frac{1}{1 + e^{-y}}, \tag{5.40}$$

$z \in \mathbb{R}^D$ being the latent variables (which in this case are the coefficients of a logistic regression) and $Y \in \mathbb{R}^{n \times D}$ a given design matrix whose entries are once generated uniformly on $[-1, 1]$. We choose $N = 100$ and will vary the dimension D. We consider a Gaussian prior $p(z) = \mathcal{N}(z; \mu, \Sigma)$, where in our experiment we choose $\mu = (0, \dots, 0)^\top, \Sigma = \mathrm{diag}(25, \dots, 25, 1)$. For the variational distribution we choose $q_\theta(z) = \mathcal{N}(z; \widetilde{\mu}, \widetilde{\Sigma})$, with $\widetilde{\Sigma} = \mathrm{diag}(\widetilde{\sigma}_1, \dots, \widetilde{\sigma}_D)$, i.e. $\theta = (\widetilde{\mu}_1, \dots, \widetilde{\mu}_D, \widetilde{\sigma}_1, \dots, \widetilde{\sigma}_D)^\top$. We train the models using stochastic gradient descent with a learning rate of 0.001.

As a second example we consider a discrete variational autoencoder (DVAE), following the setup in [204], which was also replicated in [116] and [282]. We fix the prior to be $p(z) = b(z; 0.5)$, where here and in the sequel b is applied componentwise. We consider data $x \in \mathbb{R}^m$, latent variables $z \in \mathbb{R}^D$ and define

$$p_\vartheta(x|z) = b(x; \Phi_\vartheta(z)) \qquad \text{and} \qquad q_\theta(z) = b(z; \widetilde{\Phi}_\theta(x)), \tag{5.41}$$

[37] The code is available at `https://github.com/aboustati/vargrad`.

where $\Phi_\vartheta : \mathbb{R}^D \to \mathbb{R}^m$ and $\widetilde{\Phi}_\theta : \mathbb{R}^m \to \mathbb{R}^D$ are neural networks with parameters ϑ and θ respectively.

One speaks of stochastic binary layers[38] when considering the graphical model

$$p_\vartheta(x, z) = p_\vartheta(x, z_1, \dots, z_r) = p_{\vartheta_r}(x|z_r) \left(\prod_{i=1}^{r-1} p_{\vartheta_i}(z_{i+1}|z_i) \right) p(z_1). \tag{5.42}$$

Let us choose $r = 2$ in the sequel, i.e.

$$p_\vartheta(x, z) = p_\vartheta(x, z_1, z_2) = p_{\vartheta_2}(x|z_2) p_{\vartheta_1}(z_2|z_1) p(z_1) \tag{5.43}$$

with $z = (z_1, z_2), \vartheta = (\vartheta_1, \vartheta_2)$ and

$$p_{\vartheta_2}(x|z_2) = b(x; \Phi_{2,\vartheta_2}(z_2)), \qquad p_{\vartheta_1}(z_2|z_1) = b(z_2; \Phi_{1,\vartheta_1}(z_1)), \qquad p(z_1) = b(z_1; 0.5) \tag{5.44}$$

and analogously

$$q_\theta(z) = q_\theta(z_1, z_2) = q_{\theta_1}(z_1|z_2) q_{\theta_2}(z_2) \tag{5.45}$$

with $\theta = (\theta_1, \theta_2)$ and

$$q_{\theta_1}(z_1|z_2) = b(z_1; \widetilde{\Phi}_{1,\theta_1}(z_2)), \qquad q_{\theta_2}(z_2) = b(z_2; \widetilde{\Phi}_{2,\theta_2}(x)). \tag{5.46}$$

As before we want to reach $p_\vartheta(z|x) \approx q_\theta(z)$, i.e. we want to maximize the lower bound

$$\mathrm{ELBO}(\vartheta, \theta) = \mathbb{E}_{q_\theta} \left[\log \frac{p_\vartheta(x, z)}{q_\theta(z)} \right] \tag{5.47}$$

w.r.t. θ, and now also simultaneously w.r.t. ϑ in order to improve the model p_ϑ. We note that we can write

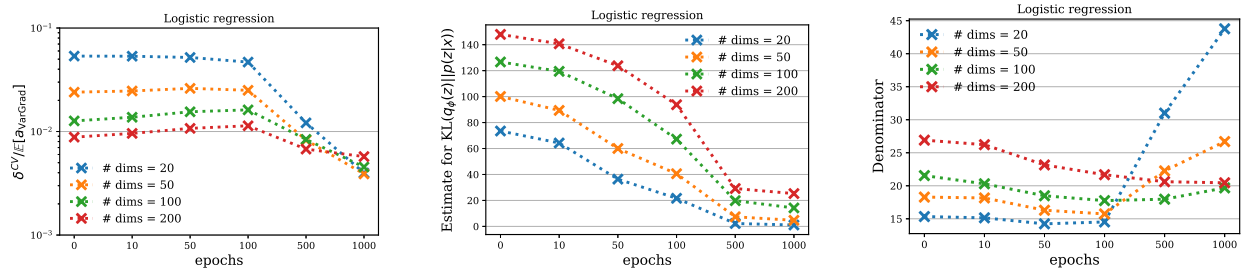
$$\nabla_\theta \,\mathrm{ELBO}(\vartheta, \theta) = -\nabla_\theta \,\mathrm{KL}(q_\theta(z)|p_\vartheta(x, z)) = -\nabla_\theta \mathcal{L}_r(q_\theta(z)|p_\vartheta(z|x))\Big|_{r=q_\theta} \tag{5.48}$$

$$\nabla_\vartheta \,\mathrm{ELBO}(\vartheta, \theta) = -\nabla_\vartheta \,\mathrm{KL}(q_\theta(z)|p_\vartheta(x, z)) = \mathbb{E}_{q_\theta} \left[\nabla_\vartheta \log p_\vartheta(x, z) \right], \tag{5.49}$$

so we perform the optimization w.r.t. θ with VarGrad, whereas the optimization w.r.t. ϑ is done with the usual KL estimator, which usually does not suffer from high variance, as we do not have to differentiate w.r.t. parameters that determine the random variables. For the data we take a binarization of Omniglot [186], where we binarize at the standard cut-off of 0.5. We use the standard train/test splits for this dataset. We use the *two layers linear* architecture, which has two stochastic binary layers with 200 units each, as used in [204]. For this model, the decoders mirror the corresponding encoders. For training the models, we use the Adam optimizer [169] with learning rates $0.001, 0.0005$ and 0.0001.

Closeness to the optimal control variate

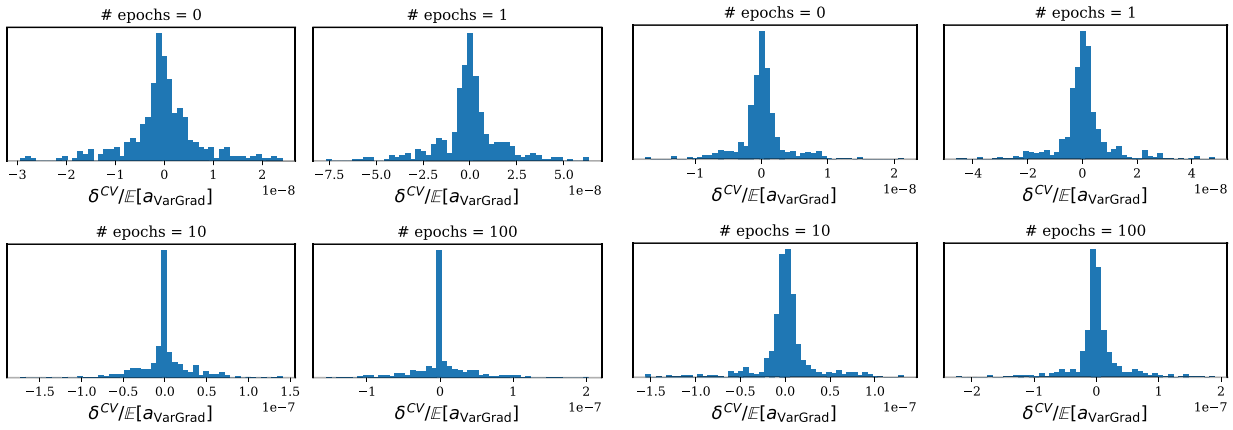
In Section 5.3 we analytically showed that VarGrad is close to the optimal control variate, and in particular that the ratio $\left|\delta_i^{\mathrm{CV}} / \mathbb{E}_{q_\theta}[a^{\mathrm{VarGrad}}]\right|$ can be small over the whole optimization procedure. This behavior is expected to be even more pronounced with growing dimensionality of the latent space. In Figure 5.1, we confirm this result by showing the ratio $\left|\delta_i^{\mathrm{CV}} / \mathbb{E}_{q_\theta}[a^{\mathrm{VarGrad}}]\right|$ for the logistic regression model. We also show the KL divergence along the iterations and the denominator of the bound in (5.26); see Figure 5.1 for the details.



(a) (b) (c)

Figure 5.1: Illustration of Proposition 5.6 and Remark 5.8 on the logistic regression model. In (a), we show that the ratio $\left|\delta_i^{\mathrm{CV}} / \mathbb{E}_{q_\theta}[a^{\mathrm{VarGrad}}]\right|$ is small and uniformly bounded over epochs, illustrating that the VarGrad estimator stays close to the optimal control variate coefficients during the whole optimization procedure. Additionally, this ratio decreases with increasing dimensionality of the latent variables. In (b), we display an estimate of the KL divergence across epochs and demonstrate the beneficial effect of higher dimensions, since the bound of (5.26) is expected to scale like $\mathcal{O}(\mathrm{KL}^{-1/2})$ in the early phase. In (c), we plot an estimate of the denominator of the bound (5.26), which increases or stays constant over epochs, demonstrating that the ratio in (5.26) stays stable (and small) over the epochs.

[38] Not to be confused with layers in a neural network.



(a) Two-layer linear DVAE. (b) nonlinear DVAE.

Figure 5.2: The distribution of $\frac{\delta_i^{\text{CV}}}{\mathbb{E}[a_{\text{VarGrad}}]}$ associated with the biases of two DVAE models with 200 latent dimensions trained on Omniglot using VarGrad. The estimates are obtained with 2000 Monte Carlo samples. The ratio $\frac{\delta_i^{\text{CV}}}{\mathbb{E}[a_{\text{VarGrad}}]}$ is consistently small throughout the optimization procedure.

In Figure 5.2, we provide further evidence that this ratio is also small when fitting DVAE. Indeed, we observe that the ratio $\delta_i^{\text{CV}} / \mathbb{E}[a^{\text{VarGrad}}]$ is typically very small and is distributed around zero during the whole optimization procedure.

Variance reduction and computational cost

In Figure 5.3 we show the variance of different gradient estimators throughout optimization in the logistic regression setting. We realize a significant improvement of VarGrad compared to the standard Reinforce estimator (5.5). In fact, we observe only a small difference between the variance of VarGrad and the variance of an *oracle estimator* based on Reinforce with access to the optimal control variate coefficient a^*. Figure 5.3 also shows the variance of the *sampled estimator*, which is based on Reinforce with an estimate of the optimal control variate; this confirms the difficulty of estimating it in practice. (A similar trend can be observed for the DVAE, where VarGrad is compared to a wider list of estimators from the DVAE literature.) All methods use $K = 4$ Monte Carlo samples, and the control variate coefficient is estimated with either 2 extra samples (*sampled estimator*) or 1000 samples (*oracle estimator*).

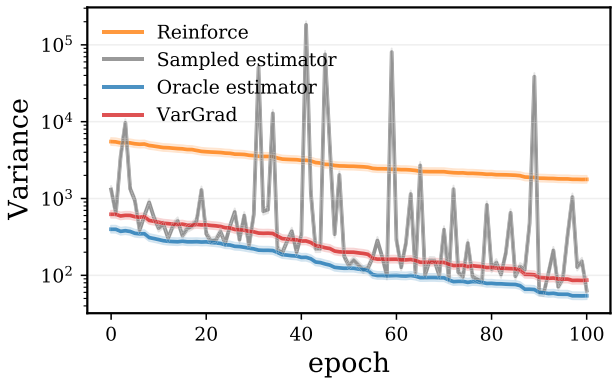


Figure 5.3: Estimates of the variance of the gradient component w.r.t. the posterior mean of one of the weights for the logistic regression model. The variance of VarGrad is close to the *oracle estimator* based on Reinforce with access to the optimal control variate coefficient a^*. Moreover, the *sampled estimator* (based on Reinforce with an estimate of a^*) shows the difficulty of estimating the optimal control variate coefficient in practice.

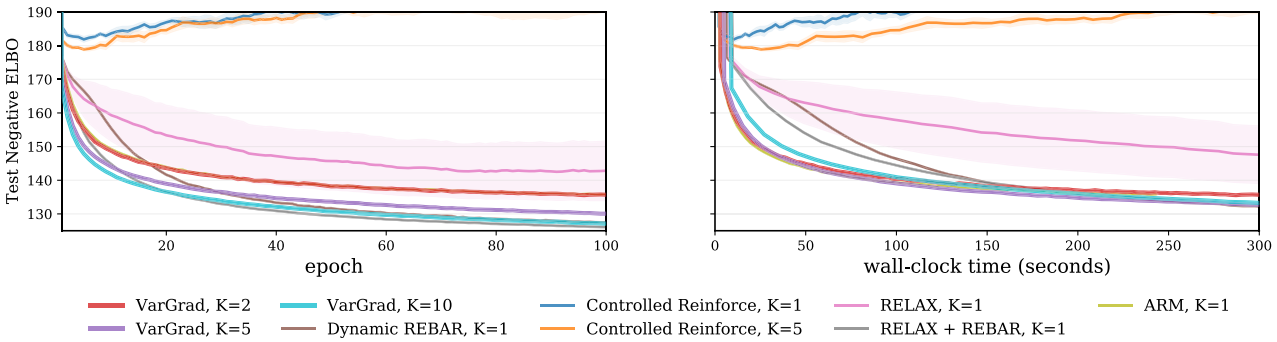


Figure 5.4: optimization trace versus epoch (left) and wall-clock time (right) for a two-layer linear DVAE on a fixed binarization of Omniglot. The plot compares VarGrad to Reinforce with score function control variates [244], dynamic REBAR [282], RELAX, RELAX + REBAR [116] and ARM [303]. The number of samples used to compute each gradient estimator is given in the figure legend. VarGrad demonstrates favorable scalability and performance when compared to the other estimators.

Finally, Figure 5.4 compares VarGad with other estimators by training a DVAE on Omniglot. The figure shows the negative ELBO as a function of the epoch number (left plot) and against the wall-clock time (right plot). The negative ELBO is computed on the standard test split and the optimization uses Adam [169] with learning rate of 0.001. VarGrad achieves similar performance to state-of-the-art estimators, such as REBAR [282], RELAX [116], and ARM [303], while being simpler to implement (see Algorithm 2) and without any tunable hyperparameters.

In Figure 5.5 we present additional results on the practical variance reduction that VarGrad induces in the two layer linear DVAE. Here, we compare with various other estimators from the literature. VarGrad achieves considerable variance reduction over the adaptive (RELAX) and non-adaptive (controlled Reinforce) model-agnostic estimators. Structured adaptive estimators such as Dynamic REBAR and RELAX + REBAR start with a higher variance at the beginning of the optimization, which reduces towards the end. ARM, which uses antithetic sampling, achieves the most reduction; however, it is only applicable to models with Bernoulli latent variables. Notably, the extra variance reduction seen in some of the methods does not translate to better optimization performance on this example.

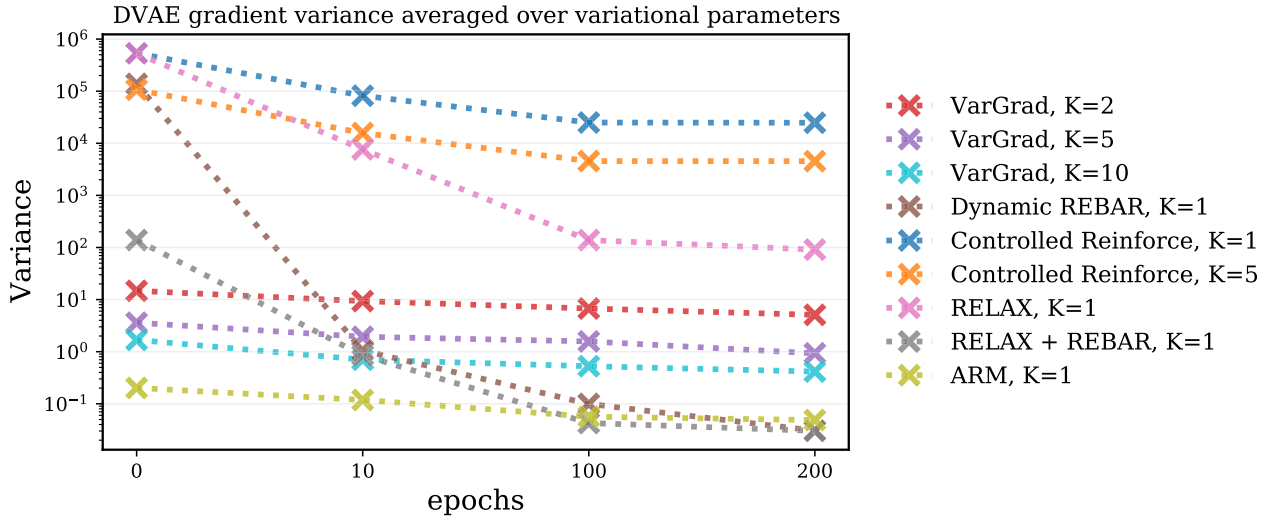


Figure 5.5: Estimates of the gradient variance of the DVAE at 4 points during the optimization for different gradient estimators. The plot compares VarGrad to Reinforce with score function control variates [244], dynamic REBAR [282], RELAX, RELAX + REBAR [116] and ARM [303]. The number of samples used to compute each gradient estimator is given in the figure legend.

5.4.1 Results on the kurtosis of the score for exponential families

Here we provide a more explicit expression for $\mathrm{Kurt}[\partial_{\theta_i} \log q_\theta]$ in the case when $q_\theta(z)$ is given by an exponential family, i.e. $q_\theta(z) = h(z) \exp\left(\theta^\top T(z) - A(\theta)\right)$, where $T(z)$ is the vector of sufficient statistics and $A(\theta)$ denotes the log-partition function. As an application, we show that in the Gaussian case, $\mathrm{Kurt}[\partial_{\theta_i} \log q_\theta]$ is uniformly bounded across the whole variational family.

Lemma 5.11. *Let $q_\theta(z) = h(z) \exp\left(\theta^\top T(z) - A(\theta)\right)$. Then*

$$\mathrm{Kurt}[\partial_{\theta_i} \log q_\theta] = \frac{\mathbb{E}_{q_\theta}\left[(T_i(z) - m_i)^4\right]}{\mathbb{E}_{q_\theta}\left[(T_i(z) - m_i)^2\right]^2}, \tag{5.50}$$

where $m_i = \mathbb{E}_{q_\theta}[T_i(z)]$ denotes the mean of the sufficient statistics. In particular, $\mathrm{Kurt}[\partial_{\theta_i} \log q_\theta]$ does not depend on $h(z)$ or $A(\theta)$.

Proof. The claim follows by direct calculation. Indeed,

$$\partial_{\theta_i} \log q_\theta(z) = T_i(z) - \frac{\partial A}{\partial \theta_i}(\theta). \tag{5.51}$$

It is left to show that $\frac{\partial A}{\partial \theta_i}(\theta) = \mu_i$. For this, notice that the normalization condition

$$\int h(z) \exp\left(\theta^\top T(z) - A(\theta)\right) \mathrm{d}z = 1 \tag{5.52}$$

implies

$$\int h(z) \left(T_i(z) - \frac{\partial A(\theta)}{\partial \theta_i}\right) \exp\left(\theta^\top T(z) - A(\theta)\right) \mathrm{d}z = 0 \tag{5.53}$$

by taking the derivative w.r.t. θ_i. The left-hand side equals $\mathbb{E}_{q_\theta}[T_i(z)] - \frac{\partial A}{\partial \theta_i}(\theta)$, and so the claim follows. □

Lemma 5.12. *Let $q_\theta(z)$ be the family of one-dimensional Gaussian distributions. Then there exists a constant $C > 0$ such that*

$$\mathrm{Kurt}[\partial_{\theta_i} \log q_\theta] < C \tag{5.54}$$

for all i and all $\theta \in \Theta$. In fact, it is possible to take $C = 15$.

Proof. For the Gaussian family, the sufficient statistics are given by $T_1(z) = z$ and $T_2(z) = z^2$. We have that

$$\frac{\mathbb{E}_{\mathcal{N}(\mu,\sigma^2)}\left[(T_1(z) - m_1)^4\right]}{\mathbb{E}_{\mathcal{N}(\mu,\sigma^2)}\left[(T_1(z) - m_1)^2\right]^2} = \frac{\mathbb{E}_{\mathcal{N}(\mu,\sigma^2)}\left[(z - \mu)^4\right]}{\mathbb{E}_{\mathcal{N}(\mu,\sigma^2)}\left[(z - \mu)^2\right]^2} = 3, \tag{5.55}$$

by the well-known fact the standard kurtosis of any univariate Gaussian is 3. A lengthy but straightforward computation shows that

$$\frac{\mathbb{E}_{\mathcal{N}(\mu,\sigma^2)}\left[(T_2(z) - m_2)^4\right]}{\mathbb{E}_{\mathcal{N}(\mu,\sigma^2)}\left[(T_2(z) - m_2)^2\right]^2} = \frac{3(4\mu^4 + 20\mu^2\sigma^2 + 5\sigma^4)}{(2\mu^2 + \sigma^2)^2}, \tag{5.56}$$

which is maximized for $\mu = 0$, taking the value 15. □

Lemma 5.12 shows that the kurtosis term in our bound (5.26) can be bounded for Gaussian families. This result is expected to extend to the multivariate cases as well. We note that we observe in our experiments that the bound is finite in a variety of cases.

5.4.2 Illustrations in the Gaussian case

In the case when $q(z)$ and $p(z|x)$ are (diagonal) Gaussians we can gain some intuition on the performance of VarGrad by computing the relevant quantities analytically. The principal insights obtained from the examples presented in this section can be summarized as follows: Firstly, in certain scenarios the Reinforce estimator does indeed exhibit a lower variance in comparison with VarGrad (although the advantage is very modest and only materializes for a restricted set of parameters). This finding illustrates that the conditions in equations (5.35) and (5.36) (the latter referring to $K \geq K_0$) cannot be dropped without replacement from the formulation of Proposition 5.9. Secondly, in line with the results from Section 5.4, the relative error $\delta_i^{\mathrm{CV}} / \mathbb{E}[a^{\mathrm{VarGrad}}]$ decreases with increased dimensionality. Moreover, the variance associated to computing the optimal control variate coefficients a^* is significant and increases considerably with the number of latent variables.

Comparing the variances of Reinforce and VarGrad

In order to understand when the variance of VarGrad is smaller than the variance of the Reinforce estimator we first consider the one-dimensional Gaussian case $q(z) = \mathcal{N}(z; \mu, \sigma^2)$ and $p(z|x) = \mathcal{N}(z; \widetilde{\mu}, \widetilde{\sigma}^2)$ and analyze the derivative w.r.t. μ. A lengthy calculation shows that

$$\Delta_{\mathrm{Var}}(\mu, \widetilde{\mu}, \sigma^2, \widetilde{\sigma}^2, K) := \mathrm{Var}(\widehat{g}_{\mathrm{Reinforce},\mu}) - \mathrm{Var}(\widehat{g}_{\mathrm{VarGrad},\mu}) \tag{5.57a}$$

$$= \frac{1}{4K\sigma^4\widetilde{\sigma}^2}\left((\mu-\widetilde{\mu})^4 + 2(\mu-\widetilde{\mu})^2\left(\frac{3K-7}{K-1}\sigma^2 - 3\widetilde{\sigma}^2\right) + \frac{5K-7}{K-1}\left(\sigma^2-\widetilde{\sigma}^2\right)^2\right) \tag{5.57b}$$

$$\approx \frac{1}{4K\sigma^4\widetilde{\sigma}^2}\left(\Delta_\mu^4 + 6\Delta_\mu^2\Delta_{\sigma^2} + 5\Delta_{\sigma^2}^2\right), \tag{5.57c}$$

where the last line holds for large K with $\Delta_\mu := \mu - \widetilde{\mu}$ and $\Delta_{\sigma^2} := \sigma^2 - \widetilde{\sigma}^2$.

For an illustration, let us vary the above parameters. First, let us fix $\sigma^2 = \widetilde{\sigma}^2 = 1$. We note from (5.57c) that in this case we expect VarGrad to have lower variance regardless of Δ_μ as long as K is large enough. In Figure 5.6 we see that this is in fact the case, however a different result can be observed for small K, which is again in accordance with (5.57b).

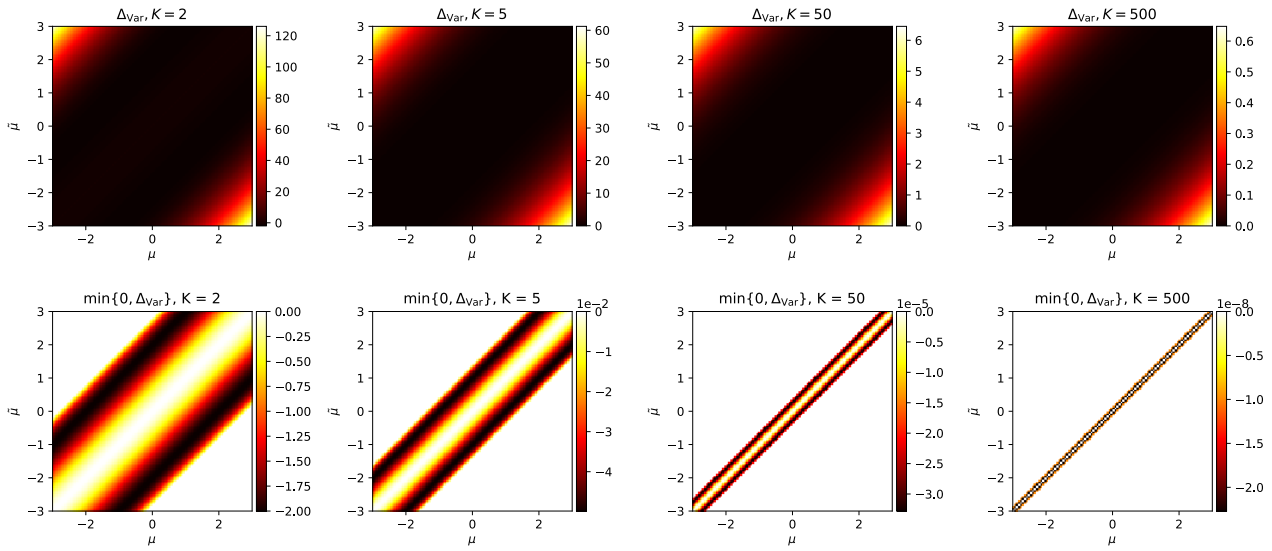


Figure 5.6: We compare the variance of the reinforce estimator with the variance of VarGrad. VarGrad is often better even for small K – for large K this can be guaranteed with Proposition 5.9.

Next, we consider arbitrary σ^2 and $\widetilde{\sigma}^2$, but fixed $\mu = 1, \widetilde{\mu} = 2$. In Figure 5.7 we observe that the variance of VarGrad is smaller for most values of σ^2 and $\widetilde{\sigma}^2$. However, even for large K there remains a region where the Reinforce estimator is superior. In fact, one can compute the condition for this to happen to be $\Delta_{\sigma^2} \in \left[-\Delta_\mu^2, -\frac{1}{5}\Delta_\mu^2\right]$, which can be compared with the condition in (5.35) in Proposition 5.9.

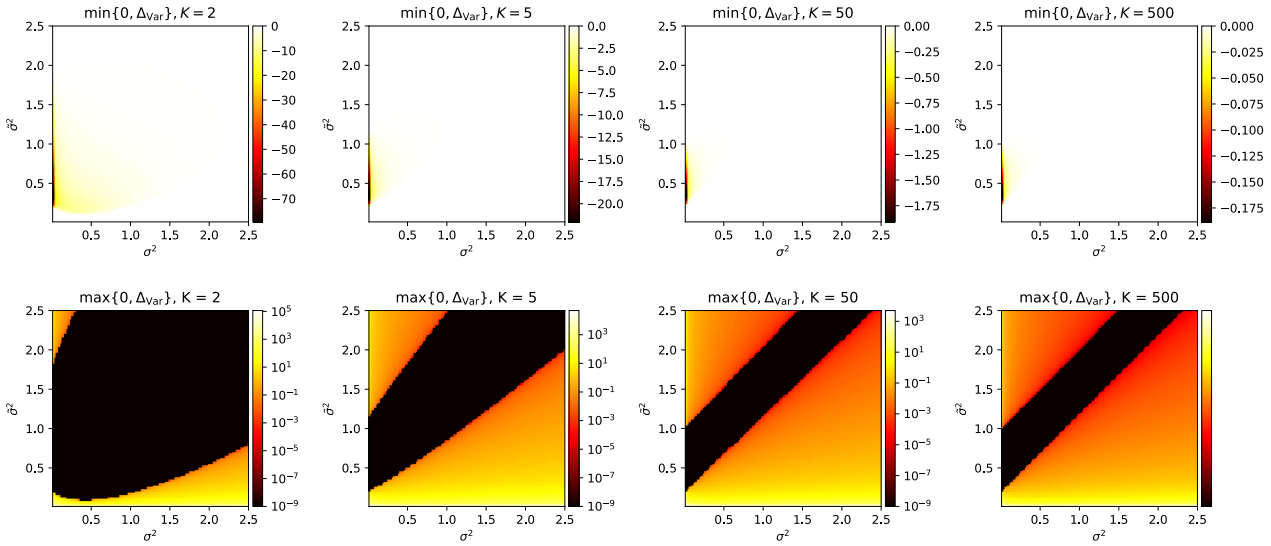


Figure 5.7: Variance comparison with varying σ^2 and $\widetilde{\sigma}^2$. VarGrad only wins outside a certain region, however, if so, then potentially by orders of magnitude.

In Figure 5.8 we display the variance differences Δ_{Var} as functions of Δ_μ and Δ_{σ^2}, approximated according to (5.57c), for the same fixed values as before and see that they are bounded from below, but not from above.

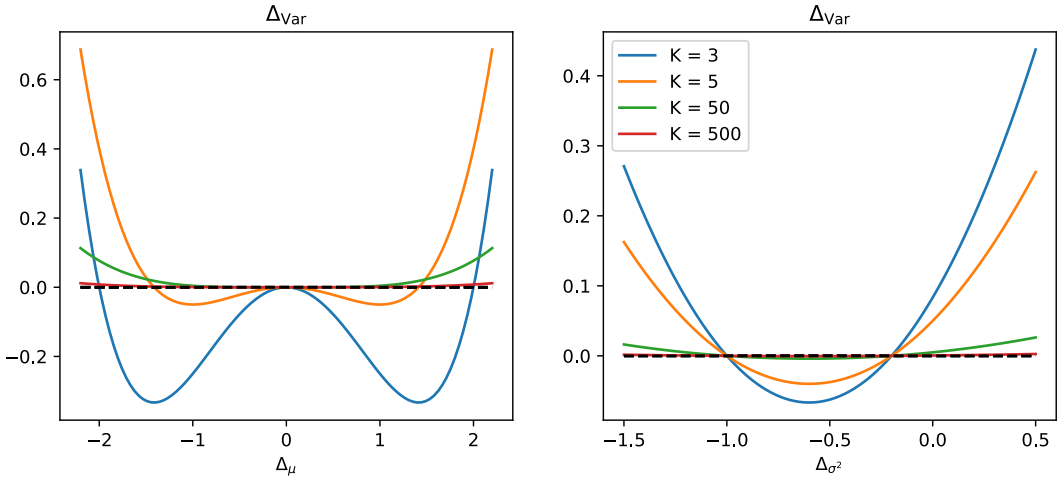


Figure 5.8: Variance differences of the Reinforce estimator and VarGrad with varying Δ_μ and Δ_{σ^2} for different sample sizes K.

For a D-dimensional Gaussian it is hard to compute the condition from (5.35) in full generality, but we can derive the following stronger criterion that can guarantee better performance of VarGrad when assuming that $\text{ELBO}(\theta) \leq 0$ (which for instance holds in the discrete-data setting).

Lemma 5.13. *Assume* $\text{ELBO}(\theta) \leq 0$ *and*

$$\text{Cov}\left(f_\theta, \left(\partial_{\theta_i} \log q_\theta\right)^2\right) > 0. \tag{5.58}$$

Then there exists $K_0 \in \mathbb{N}$ *such that*

$$\text{Var}\left(\widehat{g}_{\text{VarGrad},i}(\theta)\right) \leq \text{Var}\left(\widehat{g}_{\text{Reinforce},i}(\theta)\right), \qquad \text{for all} \quad K \geq K_0. \tag{5.59}$$

Proof. With $\text{ELBO}(\theta) \leq 0$ we have

$$\text{Cov}\left(f_\theta, \left(\partial_{\theta_i} \log q_\theta\right)^2\right) \leq \mathbb{E}_{q_\theta}\left[f_\theta \left(\partial_{\theta_i} \log q_\theta\right)^2\right] - \frac{1}{2}\,\mathbb{E}_{q_\theta}\left[f_\theta\right]\mathbb{E}_{q_\theta}\left[\left(\partial_{\theta_i} \log q_\theta\right)^2\right] \tag{5.60a}$$

$$= \mathbb{E}_{q_\theta}\left[\left(\partial_{\theta_i} \log q_\theta\right)^2\right]\left(\delta_i^{\text{CV}} - \frac{1}{2}\text{ELBO}(\theta)\right). \tag{5.60b}$$

If now

$$\mathrm{Cov}\left(f_\theta, (\partial_{\theta_i} \log q_\theta)^2\right) > 0, \tag{5.61}$$

then also

$$\delta_i^{\mathrm{CV}} - \frac{1}{2}\mathrm{ELBO}(\theta) > 0, \tag{5.62}$$

and the statement follows by Proposition 5.9. □

The condition from (5.58) gives another guarantee for VarGrad having smaller variance than the Reinforce estimator. However, we note that the converse statement is not necessarily true, i.e. if the condition does not hold, VarGrad can still be better. The advantage of (5.58), however, is that it can be verified more easily in certain settings, as for instance done for D-dimensional diagonal Gaussians in the following lemma.

Lemma 5.14 (Covariance term for diagonal Gaussians)**.** *Let $q(z)$ and $p(z|x)$ be diagonal D-dimensional Gaussians with means μ and $\widetilde{\mu}$ and covariance matrices $\Sigma = \mathrm{diag}(\sigma_1^2, \dots, \sigma_D^2)$ and $\widetilde{\Sigma} = \mathrm{diag}(\widetilde{\sigma}_1^2, \dots, \widetilde{\sigma}_D^2)$. Then*

$$\mathrm{Cov}_{q_\theta}\left(f_\theta, (\partial_{\theta_k} \log q_\theta)^2\right) = \frac{1}{\widetilde{\sigma}_k^2} - \frac{1}{\sigma_k^2} \tag{5.63}$$

for $k \in \{1, \dots, D\}$,

$$\mathrm{Cov}_{q_\theta}\left(f_\theta, (\partial_{\theta_k} \log q_\theta)^2\right) = \frac{1}{\sigma_k^2}\left(\frac{1}{\widetilde{\sigma}_k^2} - \frac{1}{\sigma_k^2}\right) \tag{5.64}$$

for $k \in \{D+1, \dots, 2D\}$ with $\theta = (\mu_1, \dots, \mu_D, \sigma_1^2, \dots, \sigma_D^2)^\top$ and

$$\mathrm{Cov}_{q_\theta}\left(f_\theta, (\partial_{\theta_k} \log q_\theta)^2\right) = \frac{1}{\widetilde{\sigma}_k^2} - \frac{1}{\sigma_k^2} \tag{5.65}$$

for $k \in \{D+1, \dots, 2D\}$ with $\theta = (\mu_1, \dots, \mu_D, \log \sigma_1^2, \dots, \log \sigma_D^2)^\top$.

Proof. See Appendix C.4. □

Optimal control variates in the Gaussian case

In the diagonal Gaussian case we can also easily analytically compute the optimal control variate coefficients from (5.8), along the lines of the proof of Lemma 5.14. Our setting is again $q(z) = \mathcal{N}(z; \mu, \Sigma)$, $p(z|x) = \mathcal{N}(z; \widetilde{\mu}, \widetilde{\Sigma})$ with $\Sigma = \mathrm{diag}\left(\sigma_1^2, \dots, \sigma_D^2\right)$, $\widetilde{\Sigma} = \mathrm{diag}\left(\widetilde{\sigma}_1^2, \dots, \widetilde{\sigma}_D^2\right)$. In Figure 5.9 we plot the variances of four different gradient estimators with varying sample size K, namely $\widehat{g}_{\mathrm{Reinforce}}$, $\widehat{g}_{\mathrm{VarGrad}}$, as well as the Reinforce estimator augmented with the optimal control variate, once computed analytically and once sampled using K samples. The variance depends on the mean and the covariance matrix; here we choose $\mu_i = 3$, $\sigma_i^2 = 3$, $\widetilde{\mu}_i = 1$, $\widetilde{\sigma}_i^2 = 1$ for all $i \in \{1, \dots, D\}$.

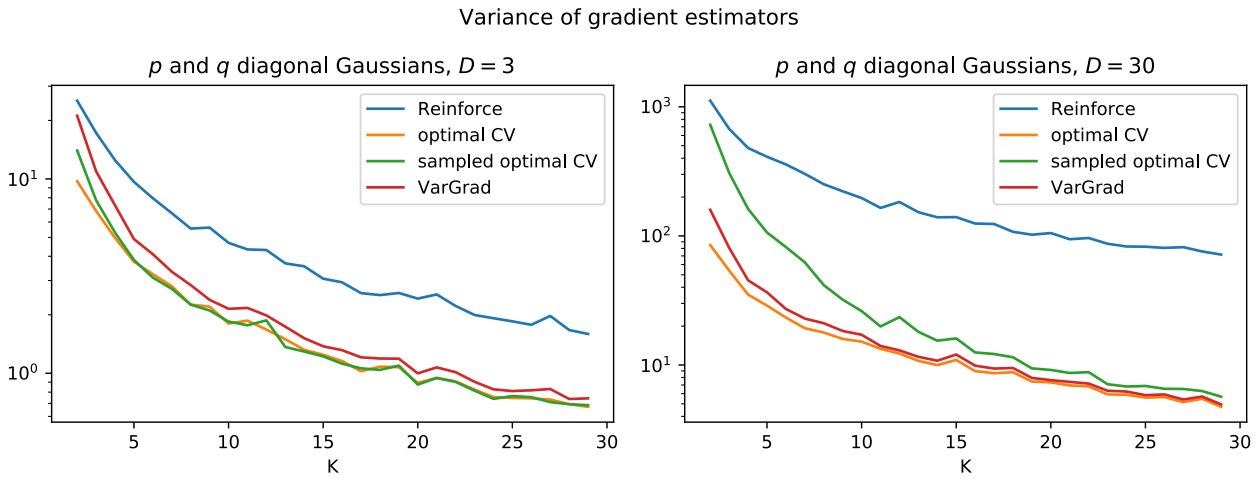


Figure 5.9: Comparion of the variances of the different gradient estimators $\widehat{g}_{\mathrm{Reinforce}}$, $\widehat{g}_{\mathrm{VarGrad}}$, as well as the reinforce estimator with the optimal control variate coefficient, once computed analytically and once sampled with K samples, for dimensions $D = 3$ and $D = 30$.

We observe that the VarGrad estimator is close to the analytical optimal control variate, and that the sampled optimal control variate performs significantly worse in a small sample size regime. These observations get more pronounced in higher dimensions and indicate that the variance of the sampled optimal control variate can itself be high, which shows that using it might not always be beneficial in practice.

Let us additionally investigate the optimal control variate correction term δ_i^{CV} as defined in (5.22) for D-dimensional Gaussians $q(z)$ and $p(z|x)$ as considered above. In Figure 5.10 we display the variances, means and relative errors of δ_i^{CV} and $a^{\mathrm{VarGrad}} = \bar{f}_\theta$ and realize that indeed the ratio of those two converges to zero when D gets larger. Furthermore we notice that the relative error of δ^{CV} increases with the dimension, explaining the difficulties when estimating the optimal control variate coefficients from samples. Finally, we plot a histogram of δ_i^{CV} (varying across i) in Figure 5.11, showing that δ_i^{CV} is small in comparison to $\mathbb{E}\left[a^{\mathrm{VarGrad}}\right]$ and distributed around zero.

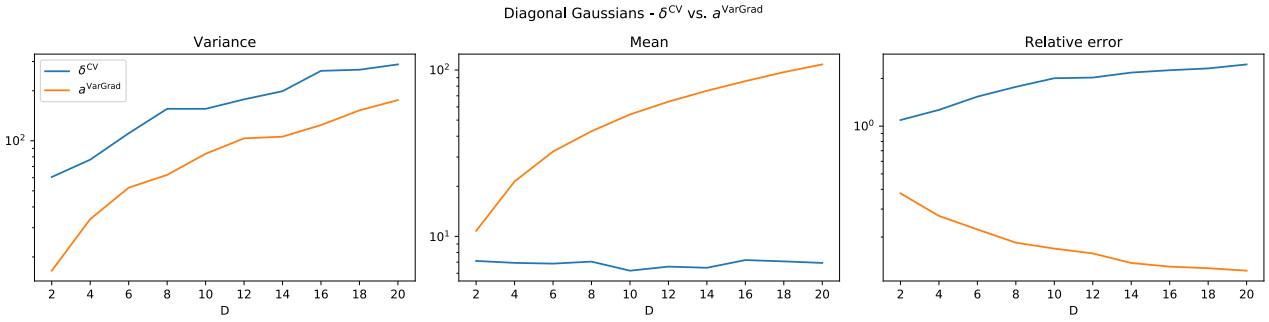


Figure 5.10: Mean, variance and relative errors associated to the two contributions to the optimal control variate coefficient, δ_i^{CV} and $a^{\mathrm{VarGrad}} = \bar{f}_\theta$.

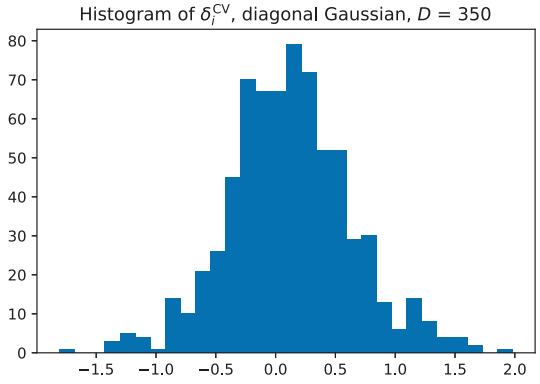


Figure 5.11: The histogram of δ_i^{CV} shows that it is usually rather small in comparison to $\mathbb{E}\left[a^{\mathrm{VarGrad}}\right]$, which is roughly 700 here, and that it fluctuates around zero.

5.4.3 VarGrad's connections to other divergences

Comparing to the other losses from the path measure setting in Chapter 4, the Reinforce gradient estimator (5.5) can as well be derived from the *moment loss*

$$\mathcal{L}_r^{\mathrm{moment}}(q_\theta(z)|p(z|x)) = \frac{1}{2}\,\mathbb{E}_{r(z)}\left[\log^2\left(\frac{q_\theta(z)}{p(z|x)}\right)\right], \tag{5.66}$$

namely

$$\nabla_\theta \mathcal{L}_r^{\mathrm{moment}}(q_\theta(z)|p(z|x))\Big|_{r=q_\theta} = \mathbb{E}_{q_\theta}\left[\log\left(\frac{q_\theta(z)}{p(z|x)}\right)\nabla_\theta \log q_\theta(z)\right]. \tag{5.67}$$

In the log-variance loss, on the other hand, one can omit the logarithm to obtain the *variance loss*

$$\mathcal{L}_r^{\mathrm{Var}}(p(z|x)|q_\theta(z)) = \frac{1}{2}\,\mathrm{Var}_{r(z)}\left(\frac{p(z|x)}{q_\theta(z)}\right) = \frac{1}{2}\,\mathbb{E}_{r(z)}\left[\left(\frac{p(z|x)}{q_\theta(z)}\right)^2 - 1\right], \tag{5.68}$$

which with $r = q_\theta$ coincides with the χ^2-divergence, see Definition 3.2. The potential of using the latter in the context of variational inference was suggested in [70]. Here one is in principle again free of choosing $r(z)$, but unlike the log-variance loss, this loss in not symmetric with respect to $q_\theta(z)$ and $p(z|x)$. Our analysis in Chapter 4 (for distributions on path space) however suggests that the variance loss (unlike the log-variance loss) scales unfavorably in high-dimensional settings in terms of the variance associated to standard Monte Carlo estimators, see also Chapter 3.

Chapter 6

Solving high-dimensional PDEs

In Chapter 4 we have developed and analyzed numerical methods for solving semi-linear PDEs with nonlinearities that depend on the solution only through its gradient (in particular relating to Problems 1.1-1.5), offering the perspective of path space measures as a unifying framework. In this chapter we want to deal with more general PDEs, now allowing for nonlinearities that may also directly depend on the solution. Even though we will not focus on path spaces anymore, stochastic representations of PDEs via diffusion processes will still play a central role. On the one hand we will consider backward iterations that approach the problem by dividing it into multiple subproblems. On the other hand, variational formulations will bring iterative optimization routines in the spirit of machine learning. An essential difference of the latter approach is that it aims to approximate solutions on entire domains rather than along trajectories of the forward process, in particular allowing for the approximation of solutions to elliptic and parabolic boundary value problems.

This chapter is organized as follows. In Section 6.1 we will start with a variational formulation of linear PDEs that relies on L^2 projections. Those projections have been prominent in algorithms for solving certain semi-linear parabolic PDEs on unbounded domains based on backward iterations, which we will cover in Section 6.2. We will particularly suggest to combine these algorithms with the tensor train format for efficient computations in high dimensions. Section 6.3 is devoted to variational formulations of elliptic and parabolic boundary value problems, introducing the novel *diffusion loss*, which can be interpreted as an interpolation of two existing methods. We will provide multiple numerical examples demonstrating advantages and drawbacks of either of the methods and show how some of them can be extended to approximate solutions of elliptic eigenvalue problems.

6.1 Linear PDEs and L^2 projections

Let us start with linear PDEs. In this section we will review and further develop an approach that relies on the Feynman-Kac formula that we have stated in Theorem 2.14. We recall that the solution to the linear PDE[39]

$$(\partial_t + L - f(x,t))V(x,t) + k(x,t) = 0, \qquad (x,t) \in \mathbb{R}^d \times [0,T), \tag{6.1a}$$

$$V(x,T) = g(x), \qquad x \in \mathbb{R}^d, \tag{6.1b}$$

admits the stochastic representation

$$V(x,t) = \mathbb{E}\left[\int_t^T e^{-\int_t^r f(X_s,s)\mathrm{d}s} k(X_r,r)\mathrm{d}r + e^{-\int_t^T f(X_s,s)\mathrm{d}s} g(X_T) \middle| X_t = x\right], \tag{6.2}$$

where the process X follows the dynamics as stated in (1.2),

$$\mathrm{d}X_s = b(X_s,s)\,\mathrm{d}s + \sigma(X_s,s)\,\mathrm{d}W_s, \qquad X_t = x, \tag{6.3}$$

for times $s \in [t,T]$. Further, let us recall that a conditional expectation as in (6.2) can be characterized as the minimizer of a least squares problem.

[39] Note that we have called the solution of a linear PDE ψ in Section 2.2.1 in order to highlight a connection to a nonlinear PDE attained via a logarithmic transformation, see Lemma 2.11. Here we will call all PDE solutions V for consistency.

Lemma 6.1 (Conditional expectation as L^2 projection)**.** *Let $A \in \mathbb{R}^{d+1}$ and $B \in \mathbb{R}$ be two random variables and let $\varphi \in C(\mathbb{R}^{d+1}, \mathbb{R})$. Then the solution to*

$$\varphi^* = \operatorname*{arg\,min}_{\varphi \in C(\mathbb{R}^{d+1},\mathbb{R})} \mathbb{E}\left[(\varphi(A) - B)^2\right] \tag{6.4}$$

is given by

$$\varphi^*(a) = \mathbb{E}[B|A = a]. \tag{6.5}$$

Proof. See Appendix C.5. □

By exploiting Lemma 6.1 we can now design a learning algorithm for the approximation of the solution to PDE (6.1). The idea is to consider the random variables

$$A = (x,t)^\top \qquad \text{and} \qquad B = \int_t^T e^{-\int_t^r f(X_s,s)\mathrm{d}s}\, k(X_r, r)\mathrm{d}r + e^{-\int_t^T f(X_s,s)\mathrm{d}s}\, g(X_T), \tag{6.6}$$

where X follows the diffusion (6.3) with randomly sampled initial times $t \sim \mu([0,T])$ and initial values $x \sim \nu(\mathbb{R}^d)$ from prescribed probability measures ν, μ. Relation (6.4) then suggests to minimize the loss

$$\mathcal{L}_{\text{linear}}(\varphi) = \mathbb{E}\left[(\varphi(x,t) - B))^2\right] \tag{6.7}$$

w.r.t. an approximating function $\varphi \in C(\mathbb{R}^d \times [0,T], \mathbb{R})$ noting that φ fulfills PDE (6.1) if and only if $\mathcal{L}(\varphi) = 0$. This algorithm has been suggested in the special case of a fixed initial time $t = 0$ for the approximation of $V(X_0, 0)$ in [12, 29] and was extended to variable $t \in [0,T]$ in the setting of parameter-dependent Kolmogorov PDEs in [27]. In practice, φ is often represented by a neural network, and we will provide a numerical example in Section 6.3.7.9.

Remark 6.2 (Linear PDEs on bounded domains). An application of the above learning strategy to linear PDEs on bounded domains $\mathcal{D} \subset \mathbb{R}^d$ is straightforward by letting the stochastic process (6.3) run until it exits $\mathcal{D}$ and by replacing the terminal condition (6.1b) with a corresponding boundary condition, cf. Remark 2.17. We note, however, that corresponding random stopping times of the process can be large, e.g. if the domain is large or if the dynamics exhibits metastable behavior. We will provide a numerical experiment in Section 6.3.7.9 and refer to Section 6.3.6.2 for further discussion on the hitting time aspect in the case of nonlinear PDEs.

6.2 Backward iteration schemes for parabolic PDEs

The L^2 projection idea stated in Lemma 6.1 has found a popular application in solving nonlinear PDEs via BSDEs. The idea it to approximate semi-linear PDEs by backward iteration schemes, which are reminiscent of the dynamic programming principle from control theory (cf. Section 2.1.1), where one divides the problem at hand into multiple subproblems and iterates backwards in time. For this attempt we shall focus on parabolic PDEs on unbounded domains, as applications to elliptic and bounded problems turn out to be not straightforward (see also Remark 6.8).

As in Definition 2.19, we consider terminal value problems of the form

$$(\partial_t + L)V(x) + h(x,t,V(x),\sigma^\top \nabla V(x)) = 0, \qquad (x,t) \in \mathbb{R}^d \times [0,T), \tag{6.8a}$$

$$V(x,T) = g(x), \qquad x \in \mathbb{R}^d. \tag{6.8b}$$

Backward iteration schemes rely on a BSDE representation of the PDE solution and we recall from Section 2.2.2 that defining the $\mathcal{F}_s$-adapted processes

$$Y_s = V(X_s, s), \qquad Z_s = \sigma^\top \nabla V(X_s, s), \tag{6.9}$$

leads to the system of forward-backward SDEs

$$\mathrm{d}X_s = b(X_s,s)\,\mathrm{d}s + \sigma(X_s,s)\,\mathrm{d}W_s, \qquad X_0 = x_{\text{init}}, \tag{6.10a}$$

$$\mathrm{d}Y_s = -h(X_s,s,Y_s,Z_s)\mathrm{d}s + Z_s \cdot \mathrm{d}W_s, \qquad Y_T = g(X_T). \tag{6.10b}$$

We aim at numerically solving for the backward processes Y and Z and therefore provide approximations of the PDE (6.8) along the forward trajectories of X via the connection stated in (6.9). To this end, let us consider a discrete version of the forward process (6.10a) on a time grid $0 = t_0 < t_1 < \cdots < t_N = T$ by

$$\widehat{X}_{n+1} = \widehat{X}_n + b(\widehat{X}_n, t_n)\Delta t + \sigma(\widehat{X}_n, t_n)\xi_{n+1}\sqrt{\Delta t}, \tag{6.11}$$

where $n \in \{0, \dots, N-1\}$ enumerates the steps, $\Delta t = t_{n+1} - t_n$ is the step-size, $\xi_{n+1} \sim \mathcal{N}(0, \mathrm{Id}_{d\times d})$ are normally distributed random variables and $\widehat{X}_0 = x_0$ provides the initial condition, as already defined in Section 2.2.3. In order to motivate different numerical discretization schemes for the backward process note that we can write (6.10b) in its integrated form for the times $t_n < t_{n+1}$ as

$$Y_{t_{n+1}} = Y_{t_n} - \int_{t_n}^{t_{n+1}} h(X_s, s, Y_s, Z_s)\mathrm{d}s + \int_{t_n}^{t_{n+1}} Z_s \cdot \mathrm{d}W_s. \tag{6.12}$$

In a discrete version we have to replace the integrals with suitable discretizations, where for the deterministic integral we can decide which end point to consider, leading to either[40]:

$$\widehat{Y}_{n+1} = \widehat{Y}_n - h(\widehat{X}_{n+1}, t_{n+1}, \widehat{Y}_{n+1}, \widehat{Z}_{n+1})\Delta t + \widehat{Z}_n \cdot \xi_{n+1}\sqrt{\Delta t}, \tag{6.13a}$$

or

$$\widehat{Y}_{n+1} = \widehat{Y}_n - h(\widehat{X}_n, t_n, \widehat{Y}_n, \widehat{Z}_n)\Delta t + \widehat{Z}_n \cdot \xi_{n+1}\sqrt{\Delta t}. \tag{6.13b}$$

Finally, we complement (6.13a) and (6.13b) by specifying the terminal conditions $\widehat{Y}_N = g(\widehat{X}_N)$ and $\widehat{Z}_N = \sigma^\top \nabla g(\widehat{X}_N)$.

Both of our schemes solve the discrete processes backwards in time. To wit, we start with the known terminal value $\widehat{Y}_N = g(\widehat{X}_N)$ and move backwards in iterative fashion until reaching $\widehat{Y}_0$. Throughout this procedure, we posit functional approximations

$$\widehat{\varphi}_n(\widehat{X}_n) \approx \widehat{Y}_n \approx V(\widehat{X}_n, t_n) \tag{6.14}$$

to be learnt in the update step $n+1 \to n$ which can either be based on (6.13a) or on (6.13b), leading to either explicit or implicit schemes. In the following we will elaborate on two approaches for their numerical treatment: L^2 regressions and direct backward schemes.

6.2.1 L^2 projections and regression-based schemes

Popular early attempts for the numerical approximation of BSDEs are regression-based schemes, often also termed *least squares Monte Carlo*, which go back to [200], have been more systematically introduced in [39, 110] and later refined in [20, 21, 112]. We can motivate them by writing the backward process Y in a slightly different way. First note that taking the conditional expectation w.r.t. the filtration $\mathcal{F}_{t_n}$ on both sides of (6.12) yields[41]

$$Y_{t_n} = \mathbb{E}\left[Y_{t_{n+1}} + \int_{t_n}^{t_{n+1}} h(X_s, s, Y_s, Z_s)\mathrm{d}s \,\middle|\, X_{t_n} \right], \tag{6.17}$$

which is reminiscent of the dynamic programming equation from Theorem 2.2 and suggests that the value of the backward process at time t_n can be written as depending on its value at time t_{n+1}. Discretizing the forward and backward processes as suggested above, while taking either the left or right end point in discretizing the

[40] It can be shown that both converge to the continuous-time process (6.10b) as $\Delta t \to 0$, see [172].

[41] Alternatively, taking conditional expectation of the integrated version of (6.10b) we can write the backward process as

$$Y_t = V(X_t, t) = \mathbb{E}\left[g(X_T) + \int_t^T h(X_s, s, Y_s, Z_s)\mathrm{d}s \,\middle|\, X_t \right], \tag{6.15}$$

where we note that $\mathbb{E}[\cdot|\mathcal{F}_t] = \mathbb{E}[\cdot|X_t]$ holds due to the Markovianity of the forward process and that by definition $Y_t = V(X_t, t)$ is $\mathcal{F}_t$-measurable (compare also to (2.67)). Using the tower property of the conditional expectation, we can now derive that for any $0 \le t_n \le t_{n+1} \le T$

$$Y_{t_n} = \mathbb{E}\left[g(X_T) + \int_{t_n}^{t_{n+1}} h(X_s, s, Y_s, Z_s)\mathrm{d}s + \int_{t_{n+1}}^{T} h(X_s, s, Y_s, Z_s)\mathrm{d}s \,\middle|\, X_{t_n} \right] \tag{6.16a}$$

$$= \mathbb{E}\left[\mathbb{E}\left[g(X_T) + \int_{t_{n+1}}^{T} h(X_s, s, Y_s, Z_s)\mathrm{d}s \,\middle|\, X_{t_{n+1}} \right] + \int_{t_n}^{t_{n+1}} h(X_s, s, Y_s, Z_s)\mathrm{d}s \,\middle|\, X_{t_n} \right] = \mathbb{E}\left[Y_{t_{n+1}} + \int_{t_n}^{t_{n+1}} h(X_s, s, Y_s, Z_s)\mathrm{d}s \,\middle|\, X_{t_n} \right]. \tag{6.16b}$$

integral analogous to (6.13a) or (6.13b), we can immediately find a discretized version of (6.17) to be[42]

$$\widehat{Y}_n = \mathbb{E}\left[\widehat{Y}_{n+1} + h(\widehat{X}_{n+1}, t_{n+1}, \widehat{Y}_{n+1}, \widehat{Z}_{n+1})\Delta t \middle| \widehat{X}_n\right], \tag{6.18}$$

or

$$\widehat{Y}_n = \mathbb{E}\left[\widehat{Y}_{n+1} + h(\widehat{X}_n, t_n, \widehat{Y}_n, \widehat{Z}_n)\Delta t \middle| \widehat{X}_n\right] \tag{6.19}$$

which hold for any $n \in \{0, \dots, N-1\}$ and where as before the terminal condition is $\widehat{Y}_N = g(\widehat{X}_N)$. Furthermore, we note that the explicit scheme (6.18) can be similarly written as

$$\widehat{Y}_n = \mathbb{E}\left[\widehat{Y}_{n+1} + h(\widehat{X}_n, t_n, \widehat{Y}_{n+1}, \widehat{Z}_{n+1})\Delta t \middle| \widehat{X}_n\right]. \tag{6.20}$$

All three schemes rely on the fact that forward and backward processes are decoupled and we realize that the explicit and implicit schemes are equivalent up to terms of order $(\Delta t)^2$ [173] so that for small enough Δt we expect similar numerical performances. In fact, the numerical properties of the above schemes (and some variants of them) have been extensively studied. One known result is that given suitable assumptions on the stochastic process and the functions g and h in the BSDE (6.10b), the approximations (6.18), (6.19) and (6.20) converge in a strong sense of order $\frac{1}{2}$, i.e. there exists a constant $C > 0$ such that

$$\max_{0 \le n \le N} \mathbb{E}\left[\left(\widehat{Y}_n - Y_{t_n}\right)^2\right] \le C\Delta t, \tag{6.21}$$

see Theorem 2.32 or [109, Theorem 7.3.1].

Remark 6.3 (Backward multistep schemes). In contrast to taking only the left or right end point, we can alternatively approximate the integral in (6.17) by employing multiple time-steps (or, equivalently, by recursively plugging in the formula for $\widehat{Y}_{n+1}$ in (6.20) and using the tower property of conditional expectations) to get the formula

$$\widehat{Y}_n = \mathbb{E}\left[g(\widehat{X}_N) + \sum_{i=n}^{N-1} h(\widehat{X}_i, t_i, \widehat{Y}_i, \widehat{Z}_i)\Delta t \middle| \widehat{X}_n\right]. \tag{6.22}$$

A corresponding scheme will lead to different numerical behaviors and we will come back to it in equation (6.43).

Let us now move towards implementable algorithms. A strategy to actually solve the backward process numerically with formulas (6.18), (6.19), (6.20) at hand can be derived by applying Lemma 6.1 and recalling that a conditional expectation can be characterized as a best approximation in L^2, namely

$$\mathbb{E}[B|\mathcal{F}_n] = \underset{\substack{A \in L^2 \\ \mathcal{F}_n-\text{measurable}}}{\arg\min} \mathbb{E}\left[(A-B)^2\right] \tag{6.23}$$

for any random-variable $B \in L^2$. Let us further recall the relations

$$Y_s = V(X_s, s), \qquad Z_s = \sigma^\top \nabla V(X_s, s) \tag{6.24}$$

for the time-continuous processes and therefore write

$$\widehat{Y}_n \approx \widehat{\varphi}_n(\widehat{X}_n), \qquad \widehat{Z}_n \approx \sigma^\top \nabla \widehat{\varphi}_n(\widehat{X}_n) \tag{6.25}$$

for their discrete counterparts, where the approximating functions $\widehat{\varphi}_n : \mathbb{R}^d \to \mathbb{R}, n \in \{0, \dots N-1\}$ are in some suitable function class $\mathcal{F}$. We can therefore define our explicit scheme (6.20) (and the schemes (6.18) and (6.19) analogously) as

$$\widehat{\varphi}_n = \underset{\varphi_n \in \mathcal{F}}{\arg\min}\, \mathbb{E}\left[\left(\varphi_n(\widehat{X}_n) - \widehat{\varphi}_{n+1}(\widehat{X}_{n+1}) - h(\widehat{X}_n, t_n, \widehat{\varphi}_{n+1}(\widehat{X}_{n+1}), \sigma^\top \nabla \widehat{\varphi}_{n+1}(\widehat{X}_{n+1}))\Delta t\right)^2\right] \tag{6.26}$$

for $n \in \{0, \dots, N-1\}$, initializing $\widehat{\varphi}_N = g$. In a sample version considering K realizations of the discrete forward process (6.11) we get the Monte Carlo approximation

$$\widehat{\varphi}_n \approx \underset{\varphi_n \in \mathcal{F}}{\arg\min} \frac{1}{K} \sum_{k=1}^{K} \left(\varphi_n(\widehat{X}_n^{(k)}) - \widehat{\varphi}_{n+1}(\widehat{X}_{n+1}^{(k)}) - h(\widehat{X}_n^{(k)}, t_n, \widehat{\varphi}_{n+1}(\widehat{X}_{n+1}^{(k)}), \sigma^\top \nabla \widehat{\varphi}_{n+1}(\widehat{X}_{n+1}^{(k)})\Delta t\right)^2. \tag{6.27}$$

[42] This relation can alternatively be derived directly from the Euler schemes (6.13a) and (6.13b) by arranging terms and taking the conditional expectation $\mathbb{E}[\cdot|\widehat{X}_n]$.

We can now state a schematic version of our backward iteration algorithm.

Algorithm 3: Backward iteration scheme for parabolic PDEs

Choose a batch size $K \in \mathbb{N}$ and time grid $0 = t_0 < \cdots < t_N = T$.
Choose a function space $\mathcal{F}$ and parametrizations $\mathbb{R}^p \ni \theta_n \mapsto \varphi_n^{\theta_n} \in \mathcal{F}$ for $n \in \{0, \dots, N-1\}$.
Simulate K trajectories of the discrete forward process (6.11).
Initialize $\widehat{Y}_N = g(\widehat{X}_N)$ and (if needed) $\widehat{Z}_N = \sigma^\top \nabla g(\widehat{X}_N)$.
for $n = N-1$ **to** 0 **do**
  Approximate either of (6.26), (6.35), (6.36), (6.42) with Monte Carlo (all depending on $\widehat{\varphi}_{n+1}$).
  Minimize this quantity (explicitly or by iterative schemes).
  Set $\widehat{\varphi}_n$ to be the minimizer.
end
Result: $\widehat{\varphi}_n \approx V(\cdot, t_n)$ along the trajectories of the forward process for $n \in \{0, \dots, N-1\}$.

Remark 6.4 (Error propagation). It is important to mention that even though dividing the problem into multiple chunks brings the advantage of potentially making each subproblem easier to solve, potential errors due to Monte Carlo approximation, optimization routines or function class properties might propagate from one time-step to the other, entailing the problem of exploding errors. In fact, assuming that we make a certain error in each time step, one can only get a bound of the overall error that grows exponentially in the time horizon T, cf. Theorems 8.3.2 and 8.3.4 in [109]. This issue does not occur in the variational approaches from Chapter 4 and Section 6.3.

It remains to specify the function class $\mathcal{F}$ and to solve the minimization problem in (6.27). We will consider parametric functions, for which two popular choices are discussed in the sequel. The first one turns out to be particularly efficient when considering explicit schemes.

6.2.1.1 Explicit least squares approximation with basis functions

We will start with the class that consists of linear combinations of some prescribed ansatz functions $\{\phi_1(x), \dots, \phi_M(x)\}$, where each $\phi_m \in C(\mathbb{R}^d, \mathbb{R})$, namely

$$\mathcal{F} = \left\{ \sum_{m=1}^{M} \theta_{n,m} \phi_m(x) : \theta_{n,m} \in \mathbb{R} \right\}. \tag{6.28}$$

We already hinted by the notation that the time-dependency of the functions will be encoded in the parameter $\theta_{n,m}$, i.e. at each time-step $n \in \{0, \dots, N-1\}$ we consider the representation

$$\varphi_n(x) = \sum_{m=1}^{M} \theta_{n,m} \phi_m(x). \tag{6.29}$$

For notational convenience, let us introduce the vectors $\theta_n := (\theta_{n,1}, \dots, \theta_{n,M})^\top$ and $b_n = (b_n^1, \dots, b_n^K)^\top$ with

$$b_n^k = \widehat{\varphi}_{n+1}(\widehat{X}_{n+1}^{(k)}) + h(\widehat{X}_n^{(k)}, t_n, \widehat{\varphi}_{n+1}(\widehat{X}_{n+1}^{(k)}), \sigma^\top \nabla \widehat{\varphi}_{n+1}(\widehat{X}_{n+1}^{(k)}))\Delta t \tag{6.30}$$

and let us define the matrices

$$A_n = \left(\phi_m(\widehat{X}_n^{(k)}) \right)_{1 \le m \le M, 1 \le k \le K}. \tag{6.31}$$

Our minimization problem (6.27) then translates to

$$\theta_n = \operatorname*{arg\,min}_{\theta \in \mathbb{R}^M} |A_n \theta - b_n|^2, \tag{6.32}$$

which, assuming that A_n has maximal rank M, can be solved explicitly by

$$\theta_n = \left(A_n^\top A_n \right)^{-1} A_n^\top b_n. \tag{6.33}$$

This easily implementable scheme has been extensively analyzed, and we for instance refer to [110] for proving convergence of order $\frac{1}{2}$ as $\Delta t \to 0$ and $M, K \to \infty$. Even though it looks inherently simple, we should note that it remains challenging to make suitable choices of ansatz functions ϕ_m in practice, where especially high-dimensional problems usually suffer from the curse of dimensionality. We will come back to this in Section 6.2.3.

6.2.1.2 Least squares approximation with neural networks

We have argued before that neural networks bring remarkable properties for function approximation (cf. Section 2.4). In this section we will therefore tackle the minimization problem (6.27) by considering the function class of feed-forward neural networks, i.e.

$$\mathcal{F} = \left\{ A_L \varrho(A_{L-1}\varrho(\cdots \varrho(A_1 x + b_1)\cdots) + b_{L-1}) + b_L : A_l \in \mathbb{R}^{n_l \times n_{l-1}}, b_l \in \mathbb{R}^{n_l}, 1 \leq l \leq L, \varrho : \mathbb{R} \to \mathbb{R} \right\}, \quad (6.34)$$

as already stated in Definition 2.49. The least-squares problem from before can be readily transferred to this setting (and we note that it has appeared under the name *deep splitting* in [11]), yielding the scheme

$$\widehat{\varphi}_n = \operatorname*{arg\,min}_{\varphi_n \in \mathcal{F}} \mathbb{E}\left[\left(\varphi_n(\widehat{X}_n) - \widehat{\varphi}_{n+1}(\widehat{X}_{n+1}) - h(\widehat{X}_{n+1}, t_{n+1}, \widehat{\varphi}_{n+1}(\widehat{X}_{n+1}), \sigma^\top \nabla \widehat{\varphi}_{n+1}(\widehat{X}_{n+1}))\Delta t \right)^2 \right] \quad (6.35)$$

for $n \in \{0, \dots, N-1\}$ with $\widehat{\varphi}_N = g$, where we have now chosen the discretization variant (6.18) instead of (6.20) in order to be more compatible with the existing literature. In contrast to the linear combination of ansatz functions in Section 6.2.1.1, the minimization problem cannot be solved analytically anymore and one instead relies on iterative minimization routines such as gradient descent. For an analysis of the approximation error of this scheme we refer to [102].

Remark 6.5 (Gradient descent initializations). The initialization of the parameters when using gradient decent algorithms (or other iterative (stochastic) optimization routines) matters and it not clear how to make good choices a priori. It is likely, however, that $\widehat{\varphi}_n$ is not all too different from $\widehat{\varphi}_{n+1}$ and therefore it seems reasonable to initialize the parameters of φ_n with the parameters of φ_{n+1} that have just been learnt in the previous time-step. In practice, this then suggests a rather long optimization runtime for the first time step and allows for much faster optimization runs thereafter.

Since we now rely on iterative optimization methods anyway, it is not much more complicated to consider the implicit time scheme instead. Following (6.19) this then leads to the minimization problem

$$\widehat{\varphi}_n = \operatorname*{arg\,min}_{\varphi_n \in \mathcal{F}} \mathbb{E}\left[\left(\varphi_n(\widehat{X}_n) - \widehat{\varphi}_{n+1}(\widehat{X}_{n+1}) - h(\widehat{X}_n, t_n, \varphi_n(\widehat{X}_n), \sigma^\top \nabla \varphi_n(\widehat{X}_n))\Delta t \right)^2 \right], \quad (6.36)$$

for $n \in \{0, \dots, N-1\}$ with $\widehat{\varphi}_N = g$, where now the function with respect to which we optimize additionally appears in the nonlinear term h.

Remark 6.6 (Numerical scheme for Z). Sometimes explicit representations of $\widehat{Z}_n$ relying on $\nabla \widehat{\varphi}_n(\widehat{X}_n)$ might not be available or are hard to compute. We can then alternatively make use of an explicit time stepping scheme for $\widehat{Z}_n$. To this end, let us multiply (6.13b) (elementwise) with ξ_{n+1}, take conditional expectation and use the fact that $\widehat{Y}_n$ is $\mathcal{F}_n$-adapted to get

$$\mathbb{E}\left[\xi_{n+1} \left(\widehat{Y}_{n+1} - \sqrt{\Delta t} \widehat{Z}_n \cdot \xi_{n+1} \right) \middle| \widehat{X}_n \right] = 0, \quad (6.37)$$

or equivalently

$$\widehat{Z}_n = \frac{1}{\sqrt{\Delta t}} \mathbb{E}\left[\xi_{n+1} \widehat{Y}_{n+1} \middle| \widehat{X}_n \right]. \quad (6.38)$$

This can be solved by the least squares approach as before, now introducing additional functions $\psi_n : \mathbb{R}^d \to \mathbb{R}^d$, aiming at

$$\widehat{\psi}_n = \operatorname*{arg\,min}_{\psi_n \in \mathcal{F}} \mathbb{E}\left[\left(\psi_n(\widehat{X}_n) - \xi_{n+1} \widehat{\varphi}_{n+1}(\widehat{X}_{n+1}) \right)^2 \right]. \quad (6.39)$$

We should note, however, that a small Δt might lead to potential numerical instabilities of this scheme.

6.2.2 Direct backward schemes

An alternative to L^2 projections for the numerical approximation of backward SDEs has been proposed in [144] and is based directly on the implicit discrete backward process (6.13b), which we can write as

$$\widehat{Y}_n - \widehat{Y}_{n+1} - h(\widehat{X}_n, t_n, \widehat{Y}_n, \widehat{Z}_n)\Delta t + \widehat{Z}_n \cdot \xi_{n+1} \sqrt{\Delta t} = 0. \quad (6.40)$$

Instead of taking conditional expectations as in Section 6.2.1, we now enforce the Euler step by penalizing deviations from (6.40) with a quadratic loss at every time-step. This results in two schemes that differ in

whether we approximate $\widehat{Z}_n$ with extra functions or whether we exploit the relation $Z_s = \sigma^\top \nabla V(X_s, s)$. For the first case this brings the minimization problem

$$(\widehat{\varphi}_n, \widehat{\psi}_n) = \underset{\varphi_n, \psi_n \in \mathcal{F}}{\arg\min}\, \mathbb{E}\left[\left(\varphi_n(\widehat{X}_n) - \widehat{\varphi}_{n+1}(\widehat{X}_{n+1}) - h(\widehat{X}_n, t_n, \varphi_n(\widehat{X}_n), \psi_n(\widehat{X}_n))\Delta t + \psi_n(\widehat{X}_n)\cdot \xi_{n+1}\sqrt{\Delta t}\right)^2\right] \tag{6.41}$$

for $n \in \{0, \dots, N-1\}$, with the initializations $\widehat{\varphi}_N = g, \widehat{\psi}_N = \sigma^\top \nabla g$. We note that in contrast to the L^2-projections from Section 6.2.1, now the Brownian increment appears in the quantity to minimize. An advantage of this approach is that its accuracy can be tested at each time-step when minimizing the loss function. Up to the time discretization, the loss should now be equal to zero for the exact solution. In contrast, the minimum of the loss functions in the previous regression-based schemes is not known for the exact solution as it corresponds to the residual of the L^2 projection, and thus the accuracy of the scheme cannot be tested directly with the available samples. Since this scheme is implicit, no closed-form minimization formulas are available and therefore often neural networks are chosen as approximating functions, relying on gradient descent for the minimizations. In our numerical experiments in Section 6.2.3 we will further rely on the tensor train format for which we will handle implicit regressions by fixed-point iterations (see Appendix B.8.2).

A variant of (6.41) that makes use of the relation $\widehat{Z}_n = \sigma^\top \nabla V(\widehat{X}_n, t_n)$, is given by

$$\widehat{\varphi}_n = \underset{\varphi_n \in \mathcal{F}}{\arg\min}\, \mathbb{E}\left[\left(\varphi_n(\widehat{X}_n) - \widehat{\varphi}_{n+1}(\widehat{X}_{n+1}) - h(\widehat{X}_n, t_n, \varphi_n(\widehat{X}_n), \sigma^\top \nabla \varphi_n(\widehat{X}_n))\Delta t + \sigma^\top \nabla \varphi_n(\widehat{X}_n)\cdot \xi_{n+1}\sqrt{\Delta t}\right)^2\right] \tag{6.42}$$

for $n \in \{0, \dots, N-1\}$ with initialization $\widehat{\varphi}_N = g$.

Finally, let us introduce one last version of direct backward iterations that relies on a multistep scheme. Here the essential idea is to incorporate multiple time-steps at once (see e.g. [102]), a procedure that we have already motivated in Remark 6.3. This results in the minimization problem

$$\begin{aligned}(\widehat{\varphi}_n, \widehat{\psi}_n) = \underset{\varphi_n, \psi_n \in \mathcal{F}}{\arg\min}\, \mathbb{E}\Bigg[\Bigg(&\varphi_n(\widehat{X}_n) - h(\widehat{X}_n, t_n, \varphi_n(\widehat{X}_n), \psi_n(\widehat{X}_n))\Delta t + \psi_n(\widehat{X}_n)\cdot \xi_{n+1}\sqrt{\Delta t} \\ &- \sum_{i=n+1}^{N-1}\left(h(\widehat{X}_i, t_i, \widehat{\varphi}_i(\widehat{X}_i), \widehat{\psi}_i(\widehat{X}_i))\Delta t - \widehat{\psi}_i(\widehat{X}_i)\cdot \xi_{i+1}\sqrt{\Delta t}\right) - g(\widehat{X}_N)\Bigg)^2\Bigg].\end{aligned} \tag{6.43}$$

for $n \in \{0, \dots, N-1\}$. While this scheme seems to lead to better convergence rates [102], it is at the same time computationally more expensive as the sum in the quantity to minimize gets large with successive time iterations.

Remark 6.7 (Log-variance loss). We have already elaborated in Remark 4.14 that we can usually consider the log-variance loss as defined in Definition 4.6 instead of a quadratic loss as long as the nonlinearity in the corresponding PDE, h, only depends on the solution V through its gradient ∇V and if one is interested in approximating ∇V instead of V (which is for instance of common interest in optimal control problems). This is also true for the backward iteration schemes that we have introduced in this section, translating for instance (6.42) into the objective

$$\widehat{\varphi}_n = \underset{\varphi_n \in \mathcal{F}}{\arg\min}\, \mathrm{Var}\left(\varphi_n(\widehat{X}_n) - \widehat{\varphi}_{n+1}(\widehat{X}_{n+1}) - h(\widehat{X}_n, t_n, \varphi_n(\widehat{X}_n), \sigma^\top \nabla \varphi_n(\widehat{X}_n))\Delta t + \sigma^\top \nabla \varphi_n(\widehat{X}_n)\cdot \xi_{n+1}\sqrt{\Delta t}\right). \tag{6.44}$$

Whether this adjustment of loss functions can lead to numerical advantages as we have demonstrated for the variational BSDE algorithms in Chapter 4 will be an interesting question of future work.

Remark 6.8 (Backward iterations on bounded domains). It is not completely obvious how to apply backward iteration schemes to elliptic and parabolic boundary value problems, where different trajectories have different lengths due to random boundary hitting times. One idea is to only consider "active" trajectories in the regression steps, however, corresponding Monte Carlo estimators might suffer from variance issues especially at the end of the trajectories due to potentially small sample sizes. Besides some attempts in [38] we are not aware of any rigorous error and convergence analysis for this case and have neither seen a systematic investigation of numerical performances. Still, in a numerical example in Section 6.2.3.6 we can see that in certain scenarios the above backward iteration schemes can produce reasonable results even in bounded domains.

Remark 6.9 (Backward iterations with forward control). Naturally, the backward iteration schemes can also be applied to generalized FBSDE systems

$$\mathrm{d}X_s^v = (b(X_s^v, s) + \sigma(X_s^v, s)v(X_s^v, s))\,\mathrm{d}s + \sigma(X_s^v, s)\,\mathrm{d}W_s, \qquad X_{t_0}^v = x_{\text{init}}, \tag{6.45a}$$

$$\mathrm{d}Y_s^v = -h(X_s^v, s, Y_s^v, Z_s^v)\,\mathrm{d}s + v(X_s^v, s)\cdot Z_s^v\,\mathrm{d}s + Z_s^v \cdot \mathrm{d}W_s, \qquad Y_T^v = g(X_T^v), \tag{6.45b}$$

that admit a control $v \in C(\mathbb{R}^d \times [0,T], \mathbb{R}^d)$ in the forward process as already defined in (4.25) for a special case. This forward control can be understood as pushing the trajectories into desired regions of the state space, noting that the relations (6.9) hold true independent of the choice of v (see Corollary 2.28). The algorithms from this section readily transfer to this change of sampling the forward process by adapting the backward process and the corresponding schemes accordingly. We will demonstrate potential numerical advantages for this adjustment in Section 6.2.3.6. Additionally, this perspective can lead to a demonstration of the relation between the backward iteration schemes and the dynamic programming principle appearing in optimal control problems (as for instance specified in Theorem 2.2), as we will show in the following lemma.

Lemma 6.10. *Consider the generalized FBSDE system as in* (6.45) *with a forward control given by* $v = -\sigma^\top \nabla V$*, where* V *is the solution to* (6.8) *with a nonlinear term*

$$h(x, s, y, z) = f(x, s) - \frac{1}{2}|z|^2. \tag{6.46}$$

Then the backward recursion as defined in (6.17) *is equivalent to the dynamic programming equation from* (2.8)*.*

Proof. See Appendix C.5. □

The above lemma is of course only of heuristic value, as V is not known in practice and therefore no numerical consequences follow.

6.2.3 Numerical examples for backward iteration schemes

In this section we provide some numerical examples for solving (high-dimensional) PDEs with the backward iteration schemes that we have introduced in the previous sections. The examples have appeared in the papers [127] and [251] and have been conducted in collaboration with Leon Sallandt, Nikolas Nüsken and Carsten Hartmann. Corresponding code can be found at `https://github.com/lorenzrichter/PDE-backward-solver` and `https://github.com/lorenzrichter/BSDE`.

For the experiments we will often rely on neural networks for the approximating functions, and, instead of using a linear combination of ansatz functions as explained in Section 6.2.1.1, we will further consider the tensor train (TT) format [220]. In fact, the salient features of tensor trains make them an ideal match for the stochastic methods alluded to in the previous section: First, tensor trains have been designed to tackle high-dimensional problems while still being computationally cheap by exploiting inherent low-rank structures [73, 163, 164] typically encountered in physically inspired PDE models. Second, built-in orthogonality relations allow fast and robust optimization in regression type problems arising naturally in stochastic backward formulations of parabolic PDEs. Third, the function spaces corresponding to tensor trains can be conveniently extended to incorporate additional information such as initial or final conditions imposed on the PDE to be solved. Last but not least, tensor trains allow for extremely efficient and explicit computation of first and higher order derivatives. All those potential benefits are expected to become particularly relevant for explicit schemes that lead to closed-form formulas for the minimization as stated in (6.33). We will provide some background on tensor trains in Appendix B.8.

Remark 6.11 (Terminal function in function class). It seems likely that at least close to the terminal time the solution to the PDE looks similar to its terminal condition g. This suggests to add g to the function class $\mathcal{F}$ and consider instead

$$\mathcal{F}_g = \{\varphi + \alpha g : \varphi \in \mathcal{F}, \alpha \in \mathbb{R}\}, \tag{6.47}$$

where α is an additional parameter that can be fixed or learnt during optimization. Note that this modification is compatible with both the linear combination of ansatz functions and nonlinear neural networks.

Remark 6.12 (Amount of samples). In the backward iteration schemes we usually consider a fixed amount of samples for the forward trajectories, which we use to compute either closed-form formulas or full gradients in iterative minimization schemes. For gradient descent, randomly choosing mini-batches in each gradient step is feasible and brings the advantage of potential computational speedups, while incorporating more different

data points over the entire optimization. However, this approach has led to worse convergence in our numerical experiments. This is different from the variational approaches in Chapter 4 and Section 6.3, where new samples are generated on the fly, which might potentially lead to a better generalizability and state space exploration in high dimensional problems.

6.2.3.1 High-dimensional Hamilton-Jacobi-Bellman equation

As stated in Section 2.1, the Hamilton-Jacobi-Bellman equation (HJB) is a PDE for the value function that represents the minimal cost-to-go in stochastic optimal control problems from which the optimal control policy can be deduced. As suggested in [86], we consider the HJB equation

$$(\partial_t + \Delta)\, V(x,t) - |\nabla V(x,t)|^2 = 0, \qquad (x,t) \in \mathbb{R}^d \times [0,T), \tag{6.48a}$$

$$V(x,T) = g(x), \qquad x \in \mathbb{R}^d, \tag{6.48b}$$

with $g(x) = \log\left(\frac{1}{2} + \frac{1}{2}|x|^2\right)$, leading to

$$b = \mathbf{0}, \quad \sigma = \sqrt{2}\,\mathrm{Id}_{d\times d}, \quad h(x,s,y,z) = -\frac{1}{2}|z|^2 \tag{6.49}$$

in terms of the notation established in (6.8). One appealing property of this equation is that (up to Monte Carlo approximation) a reference solution is available:

$$V(x,t) = -\log \mathbb{E}\left[e^{-g(x+\sqrt{T-t}\sigma\xi)}\right], \tag{6.50}$$

where $\xi \sim \mathcal{N}(\mathbf{0}, \mathrm{Id}_{d\times d})$ is a normally distributed random variable (see Lemma 2.11).

In our experiments we consider $d = 100, T = 1, \Delta t = 0.01, x_0 = (0,\dots,0)^\top$ and $K = 2000$ samples. In Table 6.1 we compare the explicit scheme stated in (6.27) with the implicit scheme from (6.42), once with tensor trains and once with neural networks. For the tensor trains we try different polynomial degrees, and it turns out that choosing constant ansatz functions is the best choice, while fixing the rank to be 1. For the neural networks we use a DenseNet like architecture with 4 hidden layers (all the details can be found in Appendices B.9).

We display the approximated solutions at $(x_0, 0)$, the corresponding relative errors $\left|\frac{\widehat{\varphi}_n(x_0) - V_{\mathrm{ref}}(x_0,0)}{V_{\mathrm{ref}}(x_0,0)}\right|$ with $V_{\mathrm{ref}}(x_0,0) = 4.589992$ being provided in [86], their computation times, as well as PDE and reference losses, which are specified in Appendix B.9. We can see that the TT approximation is both more accurate and much faster than the NN-based approaches, improving also on the results in [11, 86]. As it turns out that the explicit scheme for neural networks is worse in terms of accuracy than its implicit counterpart in all our experiments, but takes a very similar amount of computation time we will omit reporting it for the other experiments. In Figures 6.1 and 6.2 we plot the reference solutions computed by (6.50) along two trajectories of the discrete forward process (6.11) in dimensions $d = 10$ and $d = 100$ and compare to the implicit TT and NN-based approximations correspondingly. We can see that the TT approximations perform particularly well in higher dimensions.

	$\mathrm{TT}_{\mathrm{impl}}$	$\mathrm{TT}_{\mathrm{expl}}$	$\mathrm{NN}_{\mathrm{impl}}$	$\mathrm{NN}_{\mathrm{expl}}$
$\widehat{\varphi}_0(x_0)$	4.5903	4.5909	4.5822	4.4961
relative error	$5.90\mathrm{e}^{-5}$	$3.17\mathrm{e}^{-4}$	$1.71\mathrm{e}^{-3}$	$2.05\mathrm{e}^{-2}$
reference loss	$3.55\mathrm{e}^{-4}$	$5.74\mathrm{e}^{-4}$	$4.23\mathrm{e}^{-3}$	$1.91\mathrm{e}^{-2}$
PDE loss	$1.99\mathrm{e}^{-3}$	$3.61\mathrm{e}^{-3}$	90.89	91.12
comp. time	41	25	44712	25178

Table 6.1: Comparison of approximation results for the HJB equation in $d = 100$.

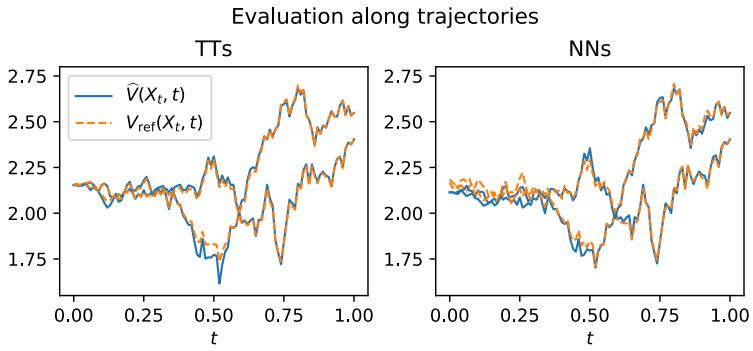


Figure 6.1: Reference solutions compared with implicit TT and NN approximations along two trajectories in $d = 10$.

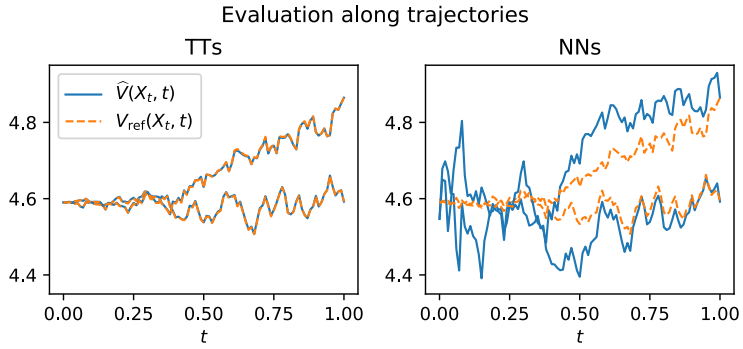


Figure 6.2: Reference solutions compared with implicit TT and NN approximations along two trajectories in $d = 100$.

In Figure 6.3 we plot the mean relative error over time, as defined in Appendix B.9, indicating that both schemes are stable and where again the implicit TT scheme yields better results than the NN scheme.

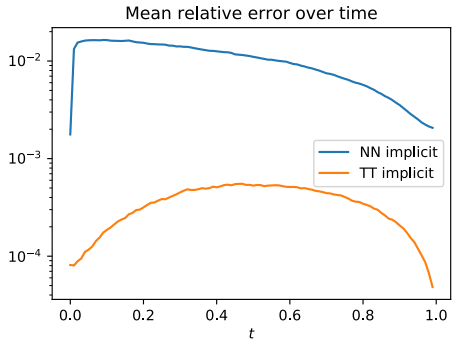


Figure 6.3: Mean relative error for TT and NN attempts.

The accuracy of the TT approximations is surprising given that the ansatz functions are constant in space. We further investigate this behavior in Table 6.2 and observe that the required polynomial degree decreases with increasing dimension. While similar "blessings of dimensionality" have been reported and discussed (see, for instance, Figure 3 in [10] and Section 1.3 in [167]), a thorough theoretical understanding is still lacking. To guide intuition, we would like to point out that the phenomenon that high-dimensional systems become in some sense simpler is well known from the theory of interacting particle systems ("propagation of chaos", see [277]):

In various scenarios, the joint distribution of a large number of particles tends to approximately factorize as the number of particles increases (that is, as the dimensionality of the joint state space grows large). It is plausible that similar approximate factorizations are relevant for high-dimensional PDEs and that tensor methods are useful (i) to detect this effect and (ii) to exploit it. In this experiment, the black-box nature of neural networks does not appear to reveal such properties.

d	Polynomial degree				
	0	1	2	3	4
1	3.62e^{-1}	3.60e^{-1}	2.47e^{-3}	3.86e^{-4}	4.27e^{-2}
2	1.03e^{-1}	1.02e^{-1}	1.87e^{-2}	1.79e^{-2}	1.79e^{-2}
5	1.55e^{-2}	1.54e^{-2}	1.03e^{-3}	9.52e^{-4}	1.96e^{-2}
10	2.84e^{-3}	2.86e^{-3}	1.37e^{-3}	1.34e^{-3}	1.10e^{-1}
50	1.17e^{-4}	1.29e^{-4}	2.79e^{-4}	3.35e^{-4}	6.96e^{-5}
100	5.90e^{-5}	4.99e^{-5}	8.65e^{-5}	1.23e^{-4}	3.62e^{-5}

Table 6.2: Relative errors of the TT approximations $\widehat{\varphi}_n(x_0)$ for different dimensions and polynomial degrees.

6.2.3.2 HJB with double-well dynamics

In another example we consider again an HJB equation, however this time making the drift in the dynamics nonlinear, as suggested in [217]. The PDE becomes

$$(\partial_t + L)\, V(x,t) - \frac{1}{2}|\sigma^\top \nabla V(x,t)|^2 = 0, \qquad (x,t) \in \mathbb{R}^d \times [0,T), \tag{6.51a}$$

$$V(x,T) = g(x), \qquad x \in \mathbb{R}^d, \tag{6.51b}$$

with L as in (1.19), where now the drift is given as the gradient of the double-well potential

$$b = -\nabla\Psi, \qquad \Psi(x) = \sum_{i,j=1}^{d} C_{ij}(x_i^2 - 1)(x_j^2 - 1) \tag{6.52}$$

and the terminal condition is $g(x) = \sum_{i=1}^d \nu_i (x_i - 1)^2$ for $\nu_i > 0$. Similarly as before a reference solution is available,

$$V(x,t) = -\log \mathbb{E}\left[e^{-g(X_T)} \middle| X_t = x \right], \tag{6.53}$$

where X_t is the forward diffusion as specified in (6.10a).

First, we consider diagonal matrices $C = 0.1\,\mathrm{Id}_{d\times d}, \sigma = \sqrt{2}\,\mathrm{Id}_{d\times d}$, implying that the dimensions do not interact, and take $T = 0.5, d = 50, \Delta t = 0.01, K = 2000, \nu_i = 0.05$. We set the TT-rank to 2, use polynomial degree 3 and refer to Appendix B.9 for further details on the TT and NN configurations. Since in the solution of the PDE the dimensions do not interact either, we can compute a reference solution with finite differences. In Table 6.3 we see that the TT and NN approximations are compatible with tensor trains having an advantage in computational time. We assume that the TT result could possibly be improved by choosing a better fit of ansatz functions, as due to the local behavior of the double well potential non-global ansatz functions might be a better choice.

	$\mathrm{TT}_{\mathrm{impl}}$	$\mathrm{NN}_{\mathrm{impl}}$
$\widehat{\varphi}_0(x_0)$	9.6876	9.6942
relative error	1.41e^{-3}	7.27e^{-4}
reference loss	1.36e^{-3}	4.25e^{-3}
PDE loss	3.62e^{-2}	2.66e^{-1}
computation time	95	1987

Table 6.3: Approximation results for the HJB equation with non-interacting double well potential in $d = 50$.

Let us now consider a non-diagonal matrix $C = \mathrm{Id}_{d\times d} + (\xi_{ij})$, where $\xi_{ij} \sim \mathcal{N}(0, 0.01)$ are sampled once at the beginning of the experiment and further choose $\sigma = \sqrt{2}\,\mathrm{Id}_{d\times d}, \nu_i = 0.5, T = 0.3$. We aim at the solution at $x_0 = (-1, \dots, -1)^\top$ and compute a reference solution with (6.53) using 10^7 samples. We see in Table 6.4 that tensor trains are much faster than neural networks, while yielding a similar performance. Note that due to the non-diagonality of C it is expected that the TTs are of rank larger than 2. For the explicit case we do not cap the ranks of the TT and the rank-adaptive solver finds ranks of mostly 4 and never larger than 6. Motivated

by these results we cap the ranks at $r_i \leq 6$ in the implicit case and indeed they are obtained for nearly every dimension, as seen from the ranks below,

$$[5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 5].$$

The results were obtained with polynomial degree 7.

	$\mathrm{TT}_{\mathrm{impl}}$	$\mathrm{TT}_{\mathrm{expl}}$	$\mathrm{NN}_{\mathrm{impl}}$
$\widehat{V}_0(x_0)$	35.015	34.756	34.917
relative error	$1.52\mathrm{e}^{-3}$	$2.82\mathrm{e}^{-3}$	$4.24\mathrm{e}^{-3}$
reference loss	$1.30\mathrm{e}^{-2}$	$1.59\mathrm{e}^{-2}$	$6.38\mathrm{e}^{-2}$
PDE loss	79.9	341	170.64
computation time	460	15	16991

Table 6.4: Approximation results for the HJB equation with interacting double well potential in $d = 20$.

6.2.3.3 Cox–Ingersoll–Ross model

Our last example is taken from financial mathematics. As suggested in [158] we consider a bond price in a multidimensional Cox–Ingersoll–Ross (CIR) model, see also [5, 151]. The underlying PDE is specified as

$$\partial_t V(x,t) + \frac{1}{2}\sum_{i,j=1}^{d} \sqrt{x_i x_j}\gamma_i\gamma_j \partial_{x_i}\partial_{x_j} V(x,t) + \sum_{i=1}^{d} a_i(b_i - x_i)\partial_{x_i} V(x,t) - \left(\max_{1\leq i\leq d} x_i\right) V(x,t) = 0. \tag{6.54}$$

Here, $a_i, b_i, \gamma_i \in [0,1]$ are uniformly sampled at the beginning of the experiment and $V(T,x) = 1$. We set $d = 100$ and aim to estimate the bond price at the initial condition $x_0 = (1,\dots,1)^\top$. As there is no reference solution known, we rely on the PDE loss to compare our results. Table 6.5 shows that all three approaches yield similar results, while having a rather small PDE loss. The TT approximations seem to be slightly better and we note that the explicit TT scheme is again much faster.

	$\mathrm{TT}_{\mathrm{impl}}$	$\mathrm{TT}_{\mathrm{expl}}$	$\mathrm{NN}_{\mathrm{impl}}$
$\widehat{\varphi}_0(x_0)$	0.312	0.306	0.31087
PDE loss	$5.06\mathrm{e}^{-4}$	$5.04\mathrm{e}^{-4}$	$7.57\mathrm{e}^{-3}$
computation time	5281	197	9573

Table 6.5: $K = 1000$, $d = 100$, $x_0 = [1, 1, \dots, 1]$

In Table 6.6 we compare the PDE loss using different polynomial degrees for the TT ansatz function and see that we do not get any improvements with polynomials of degree larger than 1.

	Polynom. degree			
	0	1	2	3
$\widehat{\varphi}_0(x_0)$	0.294	0.312	0.312	0.312
PDE loss	$9.04\mathrm{e}^{-2}$	$7.80\mathrm{e}^{-4}$	$1.05\mathrm{e}^{-3}$	$5.06\mathrm{e}^{-4}$
computation time	110	3609	4219	5281

Table 6.6: PDE loss and computation time for tensor trains with different polynomial degrees

Noticing the similarity between the results for polynomial degrees 1, 2, and 3, we further investigate by computing the value function along a sample trajectory in Figure 6.4, where we see that indeed the approximations with those polynomial degrees are indistinguishable.

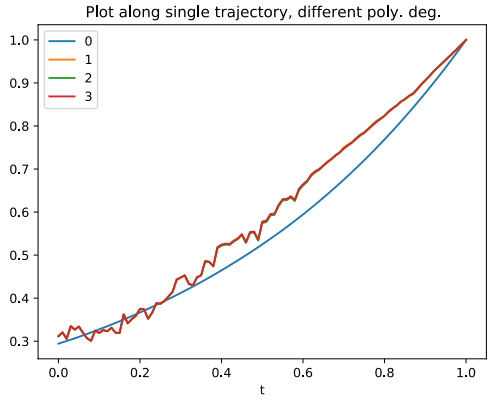


Figure 6.4: Reference trajectory for different polynomial degrees.

6.2.3.4 PDE with unbounded solution

As an additional problem, we choose an example from [144] which offers an analytical reference solution. For the PDE as defined in (2.70) we consider the coefficients

$$b(x,t) = \mathbf{0}, \qquad \sigma(x,t) = \frac{\mathrm{Id}_{d\times d}}{\sqrt{d}}, \qquad g(x) = \cos\left(\sum_{i=1}^{d} i x_i\right), \tag{6.55}$$

$$h(x,t,y,z) = k(x) + \frac{y}{2\sqrt{d}}\sum_{i=1}^{d} z_i + \frac{y^2}{2}, \tag{6.56}$$

where, with an appropriately chosen k, a solution can shown to be

$$V(x,t) = \frac{T-t}{d}\sum_{i=1}^{d}\left(\sin(x_i)\mathbb{1}_{x_i<0} + x_i\mathbb{1}_{x_i\geq 0}\right) + \cos\left(\sum_{i=1}^{d} i x_i\right). \tag{6.57}$$

In Table 6.7 we compare the results for $d = 10, K = 1000, T = 1, \Delta t = 0.001, x_0 = (0.5, \dots, 0.5)^\top$. For the TT case it was sufficient to set the ranks to 1 and we see that the results are improved significantly if we increase the sample size K from 1000 to 20000. Note that even when increasing the sample size by a factor 20, the computational time is still lower than the NN implementation. It should be highlighted that adding the function g to the neural network (as explained in Appendix B.9) is essential for its convergence in higher dimensions and thereby mitigates the observed difficulties in [144]).

	$\mathrm{TT}_{\mathrm{impl}}$	$\mathrm{TT}^*_{\mathrm{impl}}$	$\mathrm{NN}_{\mathrm{impl}}$
$\widehat{\varphi}_0(x_0)$	-0.1887	-0.2136	-0.2137
relative error	$1.22\mathrm{e}^{-1}$	$6.11\mathrm{e}^{-3}$	$5.50\mathrm{e}^{-3}$
ref loss	$2.47\mathrm{e}^{-1}$	$7.57\mathrm{e}^{-2}$	$3.05\mathrm{e}^{-1}$
abs. ref loss	$2.52\mathrm{e}^{-2}$	$9.29\mathrm{e}^{-3}$	$1.69\mathrm{e}^{-2}$
PDE loss	2.42	0.60	1.38
computation time	360	1778	4520

Table 6.7: Approximation results for the PDE with an unbounded analytic solution. For $\mathrm{TT}^*_{\mathrm{impl}}$ we choose $K = 20000$, for the others we choose $K = 1000$.

6.2.3.5 Allen-Cahn like equation

Let us consider the following Allen-Cahn like PDE with a cubic nonlinearity in $d = 100$:

$$(\partial_t + \Delta)V(x,t) + V(x,t) - V^3(x,t) = 0, \qquad x \in \mathbb{R}^d \times [0,T), \tag{6.58a}$$

$$V(x,T) = g(x), \qquad x \in \mathbb{R}^d, \tag{6.58b}$$

where we choose $g(x) = \left(2 + \frac{2}{5}|x|^2\right)^{-1}$, $T = \frac{3}{10}$ and are interested in an evaluation at $x_0 = (0, \dots, 0)^\top$. This problem has been considered in [86], where a reference solution of $V(x_0, 0) = 0.052802$ calculated by means of

the branching diffusion method is provided. We consider a sample size of $K = 1000$ and a step-size $\Delta t = 0.01$ and provide our approximation results in Table 6.8.

	$\mathrm{TT}_{\mathrm{impl}}$	$\mathrm{TT}_{\mathrm{expl}}$	$\mathrm{NN}_{\mathrm{impl}}$	$\mathrm{NN}^*_{\mathrm{impl}}$
$\widehat{\varphi}_0(x_0)$	0.05280	0.05256	0.04678	0.05176
relative error	$4.75\mathrm{e}^{-5}$	$4.65\mathrm{e}^{-3}$	$1.14\mathrm{e}^{-1}$	$1.97\mathrm{e}^{-2}$
PDE loss	$2.40\mathrm{e}^{-4}$	$2.57\mathrm{e}^{-4}$	$9.08\mathrm{e}^{-1}$	$6.92\mathrm{e}^{-1}$
comp. time	24	10	23010	95278

Table 6.8: Approximations for Allen-Cahn PDE, where $\mathrm{NN}^*_{\mathrm{impl}}$ uses $K = 8000$ and the others $K = 1000$ samples.

6.2.3.6 Double well potential with controlled forward trajectories

As an example for rare events in $d = 1$ we consider computing the probability of leaving a metastable set before a prescribed time T, namely

$$\psi(x,t) = P(\tau < T | X_t = x),$$

where the dynamics is given by the Langevin equation

$$\mathrm{d}X_s = -\nabla\Psi(X_s)\mathrm{d}s + \sigma\,\mathrm{d}W_s \tag{6.59}$$

with a potential $\Psi(x) = (x^2-1)^2$, a diffusion coefficient $\sigma > 0$ and a random stopping time $\tau = \inf\{t > 0 : X_t \notin \mathcal{D}\}, \mathcal{D} = (\infty, 0)$. We recall that leaving a metastable set scales exponentially with the energy barrier $\Delta\Psi$ and the inverse of the diffusion coefficient σ by Kramers law, namely

$$\lim_{\sigma\to 0} \sigma^2 \log(\mathbb{E}[\tau]) = 2\Delta\Psi. \tag{6.60}$$

The overall stopping time is $\min\{\tau, T\}$. Referring to the notation in (1.8) this corresponds to choosing $f(x) = 0$ and $g(x) = -\log(\mathbb{1}_{\partial\mathcal{D}}(x))$ and since the latter expression is difficult to handle numerically we consider the regularized problem by taking $g^\varepsilon(x) = -\log(\mathbb{1}_{\partial\mathcal{D}}(x) + \varepsilon)$ for a small $\varepsilon > 0$ and note that $\psi(x,t) = \psi^\varepsilon(x,t) - \varepsilon$ and $V(x,t) = -\log\left(\exp\left(-V^\varepsilon(x,t)\right) - \varepsilon\right)$. We also note that the choice of ε can have a significant effect on the corresponding optimal control as illustrated in Figure 6.5 for the choice of $\sigma = 0.2$.

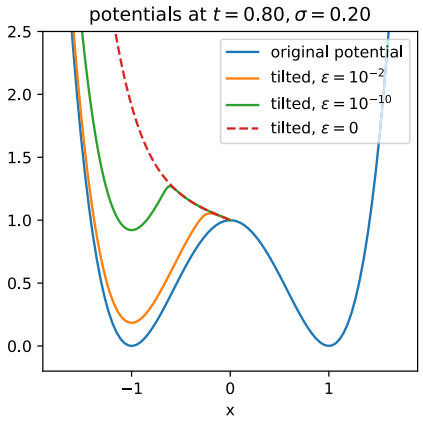


Figure 6.5: The original potential and its optimally tilted versions for different regularization values ε.

Via the Feynman-Kac formula $\psi(x,t)$ fulfills the parabolic PDE

$$(\partial_t + L)\psi(x,t) = 0, \qquad (x,t) \in \mathcal{D} \times [0,T), \tag{6.61}$$

with the boundary conditions

$$\psi(0,t) = 1, \qquad t \in [0,T), \tag{6.62a}$$
$$\psi(x,T) = 0, \qquad x \in \mathcal{D}. \tag{6.62b}$$

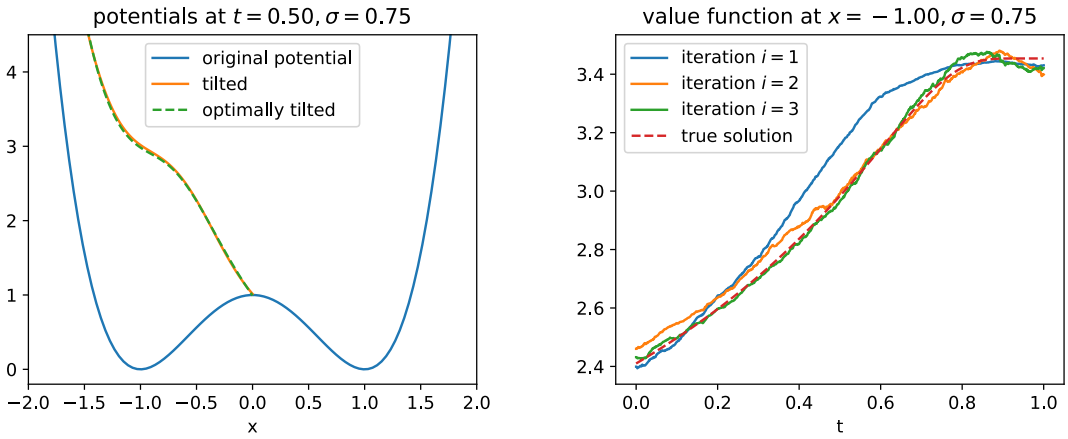


Figure 6.6: Left: The original potential and its two tilted versions – once the optimal tilting that leads to a zero-variance estimator and once its approximation. Right: The approximations of the value functions with the iterated regression based algorithm.

We numerically approach this problem by using the regression-based algorithm explained in Section 6.2.1, to be specific (6.27), which we additionally iterate by using a previously found approximation as an additional forward control as explained in Remark 6.9. More precisely, after the first iteration, the algorithm provides approximations for $\hat{Y}_n$, $\hat{Z}_n$ for $0 \leq n \leq N$, and we can use $-\hat{Z}_n$, corresponding to the optimal control, as an additional drift in the forward process to run the algorithm once again and iterate. As a small modification to the above described algorithm we choose random initial points $\widehat{X}_0 \sim \mathrm{Unif}([-1.5, 0])$, which make the algorithm more stable since in particular the matrix inversion in (6.33) is easier if trajectories are more spread out.

In our simulation, we choose $M = 5$ equidistant Gaussian functions $\phi_m(x)$ in the linear ansatz (6.29) and let $\varepsilon = 0.01, T = 1, \Delta t = 0.001, \sigma = 0.75, K = 1000$. A reference solution is computed by a numerical discretization of (6.61). In the bottom panel of Figure 6.6 we see that after the second iteration we get quite close to the true value function, however, we have no guarantee for such a behavior and depending on σ we have observed stability issues of the algorithm, which are related to a clever choice of ansatz functions and a possible clever first initial guess of a drift in the forward process. Convergence analysis of the iteration procedure is a question for further research.

As an alternative strategy for computing the rare event probabilities that we are after, which is also suitable in the case where the value function approximation seems to not converge, one can do importance sampling as an additional step. Our algorithm provides an approximation of the control by once again noting $u^* = -\sigma^\top \nabla V$ and we can use this – even if potentially suboptimal – in a Girsanov reweighting such as explained in Section 2.3.2. We illustrate this for the choice of $\sigma = 0.5$, for which the value function approximation itself did not yield satisfactory results. We compare the importance sampling approach to naive Monte Carlo, where one does not add any drift to the forward trajectories. Here the true value is

$$\psi(-1, 0) = \mathbb{P}(\tau < T) = 2.62 \times 10^{-4} \tag{6.63}$$

and we realize that the importance sampling approach brings a significant reduction of the relative error by, roughly, a factor of 20, as a consequence of which the amount of samples needed in order to reach a given accuracy is reduced by a factor of 400.

	estimated prob.	relative error	trajectories hit
naive MC	2.42×10^{-4}	61.08	0.02 %
importance sampling	2.54×10^{-4}	2.76	68.15 %

6.3 Variational formulations of elliptic and parabolic boundary value problems

In this section we will treat elliptic and parabolic PDEs on bounded and unbounded domains. Similar to Chapter 4 we will aim to approximate solutions via variational minimizations in the spirit of machine learning,

however now allowing for more general PDEs. To be precise, we consider boundary value problems of the form

$$(\partial_t + L)V(x,t) + h(x,t,V(x,t),\sigma^\top \nabla V(x,t)) = 0, \qquad (x,t) \in \mathcal{D} \times [0,T), \tag{6.64a}$$

$$V(x,T) = f(x), \qquad x \in \mathcal{D}, \tag{6.64b}$$

$$V(x,t) = g(x,t), \qquad (x,t) \in \partial\mathcal{D} \times [0,T], \tag{6.64c}$$

on a domain $\mathcal{D} \subset \mathbb{R}^d$, where $h \in C(\mathbb{R}^d \times [0,T] \times \mathbb{R} \times \mathbb{R}^d, \mathbb{R}), f \in C(\mathbb{R}^d, \mathbb{R}), g \in C(\mathbb{R}^d \times [0,T], \mathbb{R})$ are given functions and

$$L = \frac{1}{2} \sum_{i,j=1}^{d} (\sigma\sigma^\top)_{ij}(x,t)\partial_{x_i}\partial_{x_j} + \sum_{i=1}^{d} b_i(x,t)\partial_{x_i} \tag{6.65}$$

is a differential operator including the functions $b \in C(\mathbb{R}^d \times [0,T], \mathbb{R}^d)$ and $\sigma \in C(\mathbb{R}^d \times [0,T], \mathbb{R}^{d\times d})$ with σ being assumed to be non-degenerate (as defined e.g. in (1.19)). We will later make use of the fact that L is the infinitesimal generator of the diffusion process defined by the SDE

$$\mathrm{d}X_s = b(X_s, s)\,\mathrm{d}s + \sigma(X_s, s)\,\mathrm{d}W_s, \tag{6.66}$$

where W_s is a standard d-dimensional Brownian motion. Let us note that PDEs of the form (6.64) include certain prominent cases of parabolic and elliptic problems to be specified in Section 6.3.4 and that they can readily be transferred to eigenvalue problems, on which we shall elaborate in Section 6.3.5.

We have stated before that a notorious challenge which appears in the numerical treatment of PDEs is the curse of dimensionality, suggesting that the computational complexity increases exponentially in the dimension of the state space. In recent years, however, multiple numerical [86, 144, 242] and some theoretical works [119, 157] have suggested that combining certain Monte Carlo methods with neural networks offers a promising way to overcome this problem. Two strategies that allow for solving quite general nonlinear PDEs are based either on direct residual minimizations (e.g. physics informed neural networks (PINNs) [242] and the deep Galerkin method (DGM) [271]) or on backward stochastic differential equations (BSDEs) [86]. In Sections 6.3.1 and 6.3.2 we will review and generalize those attempts, while highlighting that both can be understood as variational formulations of the boundary value problem at hand. Motivated by the existing methods and building on the inherent connection between the PDE (6.64) and the SDE (6.66), we will introduce an alternative variational approach that builds on the novel *diffusion loss* in Section 6.3.3. It turns out that this new method contains the residual and BSDE methods as edge cases in some appropriate sense, noting that to the best of our knowledge it is in fact the first time that these two seemingly different approaches get connected. Besides this theoretical insight, we realize that the diffusion loss brings some algorithmic advantages, such as fast computations in high dimensions, especially when full Hessians are present in the PDE, as well as accurate approximations near the boundary of a domain. We will discuss some modification of the losses in Section 6.3.6 and provide numerous numerical examples, especially in high dimensional settings in Section 6.3.7.

This section is based on joint work with Nikolas Nüsken and will be published soon in [216].

Remark 6.13 (Compact notation). Note that for the sake of a more compact notation, we can write problem (6.64) in the following equivalent way. Consider the operator $\mathcal{A} = \partial_t + L$, the domain $\mathcal{D}_T = \mathcal{D} \times [0,T)$, the boundary $\partial\mathcal{D}_T = \mathcal{D} \times \{T\} \cup \partial\mathcal{D} \times [0,T]$, and the augmented variable $z = (x,t)^\top \in \mathbb{R}^{d+1}$. Then problem (6.64) becomes

$$\mathcal{A}V(z) + h(z, V(z), \sigma^\top \nabla_x V(z)) = 0, \qquad z \in \mathcal{D}_T, \tag{6.67a}$$

$$V(z) = k(z), \qquad z \in \partial\mathcal{D}_T, \tag{6.67b}$$

with

$$k(z) = \begin{cases} f(x), & t = T, x \in \mathcal{D}, \\ g(x,t), & t \le T, x \in \partial\mathcal{D}. \end{cases} \tag{6.68}$$

Remark 6.14 (Neumann boundary conditions). In (6.64) we have posed a PDE with Dirichlet boundary conditions. Similarly we can include Neumann boundary terms, such as

$$\frac{\partial}{\partial\nu} V(x,T) = f^N(x), \qquad x \in \partial\mathcal{D}, \tag{6.69a}$$

$$\frac{\partial}{\partial\nu} V(x,t) = g^N(x,t), \qquad (x,t) \in \partial\mathcal{D} \times [0,T], \tag{6.69b}$$

where $\frac{\partial}{\partial\nu}$ is the derivative in the direction normal to the boundary. All methods that we will discuss can be applied in this scenario too.

We now consider boundary value problems such as (6.64) in a variational formulation. We make the following assumption for the PDE solution.

Assumption 5. *The boundary value problem* (6.64) *admits a unique classical solution* $V \in C^{2,1}(\mathbb{R}^d \times [0,T], \mathbb{R})$. *Moreover, the gradient of* V *satisfies a polynomial growth condition in* x, *that is,*

$$|\nabla V(x,t)| \leq C(1+|x|^q) \tag{6.70}$$

for some $C, q > 0$.

In the spirit of machine learning, we aim to approximate the solution V with some function $\varphi \in \mathcal{F}$ by minimizing suitable loss functionals

$$\mathcal{L} : \mathcal{F} \to \mathbb{R}^+, \tag{6.71}$$

which are zero if and only if the boundary value problem is fulfilled, i.e.

$$\mathcal{L}(\varphi) = 0 \iff \varphi = V. \tag{6.72}$$

Here $\mathcal{F} \subset C^{2,1}(\mathbb{R}^d \times [0,T], \mathbb{R})$ is some appropriate function class, usually consisting of deep neural networks. With a loss function at hand we can apply gradient-descent like algorithms to minimize estimator versions of $\mathcal{L}$, having in mind that different choices of losses lead to different statistical and computational properties and therefore potentially to different convergence speeds and robustness qualities (see Chapter 4). Let us start by introducing two prominent losses that we have already referred to before.

6.3.1 PINN loss

A loss based on PDE residuals goes back to [184, 185] and has gained recent popularity under the name *physics informed neural network* (PINN) in [242] and under the name *deep Galerkin method* (DGM) in [271]. The idea is quite simple: one minimizes the L^2 norm of the residuals of both the PDE and its boundary terms respectively that one gets when using the approximating function φ instead of V, where the derivatives of φ are computed analytically or via automatic differentiation and the data on which φ is evaluated is distributed according to some prescribed probability measure (often a uniform distribution). A precise definition is as follows:

Definition 6.15 (PINN loss)**.** Let $\varphi \in \mathcal{F}$. The *PINN loss* consists of three terms,

$$\mathcal{L}_{\text{PINN}}(\varphi) = \alpha_1 \mathcal{L}_{\text{PINN,int}}(\varphi) + \alpha_2 \mathcal{L}_{\text{PINN,T}}(\varphi) + \alpha_3 \mathcal{L}_{\text{PINN,b}}(\varphi), \tag{6.73}$$

where

$$\mathcal{L}_{\text{PINN,int}}(\varphi) = \mathbb{E}\left[\left((\partial_t + L)\varphi(X,t) + h(X,t,\varphi(X,t),\sigma^\top \nabla\varphi(X,t))\right)^2\right], \tag{6.74a}$$

$$\mathcal{L}_{\text{PINN,T}}(\varphi) = \mathbb{E}\left[\left(\varphi(X^T,T) - f(X^T)\right)^2\right], \tag{6.74b}$$

$$\mathcal{L}_{\text{PINN,b}}(\varphi) = \mathbb{E}\left[\left(\varphi(X^b,t^b) - g(X^b,t^b)\right)^2\right], \tag{6.74c}$$

$\alpha_1, \alpha_2, \alpha_3 > 0$ are suitable weights and $X, X^T \sim \nu(\mathcal{D}), X^b \sim \mu(\partial\mathcal{D}), t, t^b \sim \lambda([0,T])$ are sampled randomly from probability measures with full supports on the respective domains.

Remark 6.16 (PINN loss). The PINN loss does in fact not rely on the specific form of the differential operator L as defined in (6.65) and can easily be applied to more general PDEs. One can for instance include differential operators containing $\partial_t\partial_t$ or $\partial_{x_i}\partial_{x_j}\partial_{x_k}$. Let us further already mention that making appropriate choices of the weights $\alpha_1, \alpha_2, \alpha_3 > 0$ is important, but not trivial. We will elaborate on this aspect in Section 6.3.6.1.

6.3.2 BSDE loss

The second loss makes use of a stochastic representation of boundary value problem (6.64) given by a backward stochastic differential equation (BSDE) rooted in the correspondence between the differential operator L defined in (6.65) and the stochastic process defined in (6.66) (cf. Section 2.2.2, Chapter 4 and Section 6.2). To wit, the PDE (6.64) is equivalent to the system of forward and backward SDEs [228]

$$\mathrm{d}X_s = b(X_s,s)\,\mathrm{d}s + \sigma(X_s,s)\,\mathrm{d}W_s, \qquad X_{t_0} = x_{\text{init}}, \tag{6.75a}$$

$$\mathrm{d}Y_s = -h(X_s,s,Y_s,Z_s)\mathrm{d}s + Z_s \cdot \mathrm{d}W_s, \qquad Y_T = k(X_{T\wedge\tau}, T\wedge\tau), \tag{6.75b}$$

where $\tau = \inf\{t > 0 : X_t \notin \mathcal{D}\}$ is a stopping time and k is defined as in (6.68), in the sense that, given some regularity conditions, the backward processes are equal to

$$Y_s = V(X_s,s), \qquad Z_s = \sigma^\top \nabla V(X_s,s), \tag{6.76}$$

i.e. they provide the solution and its derivative along trajectories of the forward process. Aiming for the approximation $\varphi \approx V$, the idea is to penalize deviations from the terminal condition via the loss

$$\mathcal{L}(\varphi) = \mathbb{E}\left[\left(k(X_{T\wedge\tau}, T\wedge\tau) - \widetilde{Y}_T(\varphi)\right)^2\right], \tag{6.77}$$

where $\widetilde{Y}(\varphi)$ is the backward process Y as in (6.75b) and (6.76) with V replaced by φ. This results in the following loss.

Definition 6.17 (BSDE loss)**.** Let $\varphi \in \mathcal{F}$. The *BSDE loss* is defined as

$$\begin{aligned}\mathcal{L}_{\mathrm{BSDE}}(\varphi) = \mathbb{E}\Bigg[\Bigg(&f(X_{\tau\wedge T})\mathbb{1}_{\tau\wedge T=T} + g(X_{\tau\wedge T}, \tau\wedge T)\mathbb{1}_{\tau\wedge T=\tau} - \varphi(X_{t_0}, t_0) - \int_{t_0}^{\tau\wedge T} \sigma^\top \nabla\varphi(X_s, s)\cdot \mathrm{d}W_s \\ &+ \int_{t_0}^{\tau\wedge T} h(X_s, s, \varphi(X_s, s), \sigma^\top\nabla\varphi(X_s, s))\mathrm{d}s\Bigg)^2\Bigg],\end{aligned} \tag{6.78}$$

where $(X_t)_{0\le t\le \tau\wedge T}$ is a solution to (6.66), $\tau = \inf\{t > 0 : X_t \notin \mathcal{D}\}$ is the first exit time from $\mathcal{D}$ and $X_0 \sim \nu(\mathcal{D}), t_0 \sim \lambda([0,T])$ are sampled from prescribed probability measures.

Remark 6.18 (BSDE loss). In contrast to the PINN loss, the BSDE loss consists of only one term and does not rely on additional data for learning the boundary conditions. This has the advantage of not needing to tune weights $\alpha_1, \alpha_2, \alpha_3$, but brings the additional challenge of simulating hitting times τ efficiently and accurately. We shall elaborate on this aspect in Section 6.3.6.2.

Remark 6.19 (Related work). The idea of approximating PDEs by solving BSDEs has been studied extensively [39, 110, 228], where first approaches were regression based, relying on iterations backwards in time (cf. Section 6.2). A variational attempt using neural networks has first been introduced in [86], where however in contrast to Definition 6.17, $t_0 = 0$ is fixed, only parabolic problems are considered and slightly different choices for the approximations are chosen, namely V is only approximated at the fixed initial condition X_0 at $t_0 = 0$ and ∇V instead of V is learnt by multiple instead of only one neural network. We note that the correspondence between BSDEs and PDEs on bounded domains is e.g. justified by [228, Section 4].

Remark 6.20 (Compact version of BSDE loss). Relying on the more compact PDE notation from Remark 6.13, we can equivalently define the BSDE loss as

$$\mathcal{L}_{\mathrm{BSDE}}(\varphi) = \mathbb{E}\left[\left(k(X_{\tau_{\mathcal{D}_T}}, \tau_{\mathcal{D}_T}) - \varphi(X_{t_0}, t_0) - \int_{t_0}^{\tau_{\mathcal{D}_T}} \sigma^\top\nabla\varphi(X_s, s)\cdot \mathrm{d}W_s + \int_{t_0}^{\tau_{\mathcal{D}_T}} h(X_s, s, \varphi(X_s, s), \sigma^\top\nabla\varphi(X_s, s))\mathrm{d}s\right)^2\right], \tag{6.79}$$

where $(X_t)_{0\le t\le \tau_{\mathcal{D}_T}}$ is a solution to (6.66), $\tau_{\mathcal{D}_T} = \inf\{t > 0 : X_t \notin \mathcal{D}_T\}$ is an exit time and $X_0 \sim \nu(\mathcal{D}), t_0 \sim \mu([0,T])$ are sampled.

Remark 6.21 (BSDE loss for linear PDEs). When approaching linear PDEs with the BSDE loss from Definition 6.17, the resulting algorithm can be directly compared to an approach that relies on L^2 projections as suggested in Section 6.1. For simplicity let us consider the linear PDE

$$(\partial_t + L)V(x,t) = 0, \qquad (x,t) \in \mathbb{R}^d \times [0,T), \tag{6.80a}$$

$$V(x,T) = g(x), \qquad x \in \mathbb{R}^d, \tag{6.80b}$$

which amounts to choosing $f = k = 0$ in (6.1). The variational formula (6.7) then translates to

$$\mathcal{L}_{\mathrm{linear}}(\varphi) = \mathbb{E}\left[\left(g(X_T) - \varphi(x,t)\right)^2\right], \tag{6.81}$$

recalling that the initial times $t \sim \mu([0,T])$ and initial values $x \sim \nu(\mathbb{R}^d)$ are sampled from some prescribed probability measures ν, μ. The BSDE loss as defined in Definition 6.17 on the other hand is

$$\mathcal{L}_{\mathrm{BSDE}}(\varphi) = \mathbb{E}\left[\left(g(X_T) - \varphi(x,t) - \int_t^T \sigma^\top\nabla\varphi(X_s, s)\cdot \mathrm{d}W_s\right)^2\right]. \tag{6.82}$$

Comparing expressions (6.81) and (6.82) we realize that they only differ in the extra Itô integral involving gradients of the approximating function. We refer to Section 6.3.7.9 for a numerical comparison of the two losses.

6.3.3 Diffusion loss

We will now introduce a novel loss that combines ideas of the previously defined PINN and BSDE losses. Similar to the BSDE loss it is rooted in the connection between SDE (6.66) and its infinitesimal generator (6.65), to be precise, it relies on Itô's formula

$$V(X_T, T) - V(X_0, 0) = \int_0^T (\partial_s + L)\, V(X_s, t)\mathrm{d}s + \int_0^T \sigma^\top \nabla V(X_s, s) \cdot \mathrm{d}W_s, \tag{6.83}$$

which motivates the following variational formulation of the boundary value problem (6.64).

Definition 6.22 (Diffusion loss)**.** Let $\varphi \in \mathcal{F}$. The *diffusion loss* consists of three terms,

$$\mathcal{L}^{\mathfrak{t}}_{\text{diffusion}}(\varphi) = \alpha_1 \mathcal{L}^{\mathfrak{t}}_{\text{diffusion,int}}(\varphi) + \alpha_2 \mathcal{L}^{\mathfrak{t}}_{\text{diffusion,T}}(\varphi) + \alpha_3 \mathcal{L}^{\mathfrak{t}}_{\text{diffusion,b}}(\varphi), \tag{6.84}$$

where

$$\mathcal{L}^{\mathfrak{t}}_{\text{diffusion,int}}(\varphi) = \mathbb{E}\left[\left(\varphi(X_{\mathcal{T}}, \mathcal{T}) - \varphi(X_{t_0}, t_0) - \int_{t_0}^{\mathcal{T}} \sigma^\top \nabla\varphi(X_s, s) \cdot \mathrm{d}W_s \right.\right. \tag{6.85a}$$

$$\left.\left. + \int_{t_0}^{\mathcal{T}} h(X_s, s, \varphi(X_s, s), \sigma^\top \nabla\varphi(X_s, s))\mathrm{d}s\right)^2\right],$$

$$\mathcal{L}^{\mathfrak{t}}_{\text{diffusion,T}}(\varphi) = \mathbb{E}\left[\left(\varphi(X^T, T) - f(X^T)\right)^2\right], \tag{6.85b}$$

$$\mathcal{L}^{\mathfrak{t}}_{\text{diffusion,b}}(\varphi) = \mathbb{E}\left[\left(\varphi(X^b, t^b) - g(X^b, t^b)\right)^2\right], \tag{6.85c}$$

$\alpha_1, \alpha_2, \alpha_3 > 0$ are suitable weights, $(X_t)_{t_0 \le t \le \mathcal{T}}$ is a solution to (6.66) with maximal trajectory length $\mathfrak{t} > 0$, $\mathcal{T} := (t_0 + \mathfrak{t}) \wedge \tau \wedge T$ is a shorthand notation, where $\tau = \inf\{t > 0 : X_t \notin \mathcal{D}\}$ is an exit time, and $X_{t_0}, X^T \sim \nu(\mathcal{D}), X^b \sim \lambda(\partial\mathcal{D}), t_0, t^b \sim \mu([0, T])$ are sampled randomly from prescribed probability measures with full supports on the respective domains.

Remark 6.23 (Differences to other losses). Let us comment on the essential differences of the diffusion loss to the other two losses that we have defined previously. In contrast to the PINN loss, the data inside the domain is not sampled according to a prescribed probability measure, but along trajectories of the diffusion (6.66). Consequently, second derivatives do not have to be computed explicitly, but are approximated via the driving Brownian motion. A main difference to the BSDE loss is that the simulated trajectories have a maximal length, which might be beneficial if exit times are large. Additionally, the sampling of extra boundary data circumvents the problem of accurately simulating those exit times. Both aspects will be further discussed in Section 6.3.6. We refer to Figure 6.7 for a graphical illustration of the three losses.

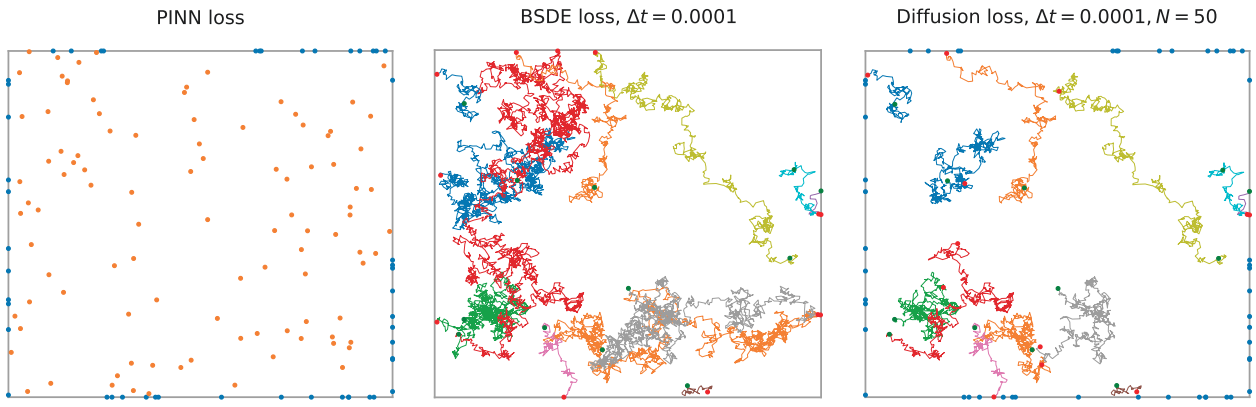


Figure 6.7: We illustrate the training data used for the three losses inside the unit square $\mathcal{D} = (0, 1)^2$. The PINN loss in the left panel takes i.i.d. data points that are sampled from prescribed probability distributions in the domain and on the boundary respectively (in this case a uniform distribution). The BSDE loss in the middle consists of trajectories that are started at random points (green points) and run until they hit the boundary (red points). The trajectories of the diffusion loss on the other hand have a maximal length and can therefore start and end inside the domain, as displayed in the right panel. The blue points for the PINN and diffusion losses indicate the additionally sampled boundary data.

Let us now show that the novel loss $\mathcal{L}^{\mathfrak{t}}_{\text{diffusion}}$ is indeed suitable for the boundary value problem (6.64).

Proposition 6.24. *Consider the diffusion loss as defined in Definition 6.22 and assume that b and σ are globally Lipschitz continuous in x, uniformly in t. Furthermore, assume the following Lipschitz and boundedness conditions on f, g and h,*

$$\begin{aligned}
|f(x)| &\leq C(1+|x|^p),\\
|g(x,t)| &\leq C(1+|x|^p),\\
|h(t,x,y,z)| &\leq C(1+|x|^p+|y|+|z|),\\
|h(t,x,y,z)-h(t,x,y',z)| &\leq C|y-y'|,\\
|h(t,x,y,z)-h(t,x,y,z')| &\leq C|z-z'|,
\end{aligned}$$

for appropriate constants $C, p \geq 0$. Finally, assume that Assumption 5 is satisfied. Then for $\varphi \in \mathcal{F}$ the following are equivalent:

1. *φ minimizes the diffusion loss,*
$$\mathcal{L}^{\mathfrak{t}}_{\text{diffusion}}(\varphi) = 0. \tag{6.86}$$
2. *φ fulfills the boundary value problem* (6.64)*.*

Proof. We first note that $\mathcal{L}^{\mathfrak{t}}_{\text{diffusion}}(\varphi) \geq 0$ for all $\varphi \in \mathcal{F}$. Denoting by X_s the unique strong solution to (6.66), an application of Itô's lemma to $\varphi(X_s, s)$ yields

$$\varphi(X_{\mathcal{T}}, \mathcal{T}) = \varphi(X_{t_0}, t_0) + \int_{t_0}^{\mathcal{T}} (\partial_s + L)\varphi(X_s, s)\,\mathrm{d}s + \int_{t_0}^{\mathcal{T}} \sigma^\top \nabla\varphi(X_s, s)\cdot \mathrm{d}W_s, \tag{6.87}$$

almost surely. Assuming that φ fulfills the PDE (6.64a), it follows from the definition in (6.85a) that $\mathcal{L}^{\mathfrak{t}}_{\text{diffusion,int}}(\varphi) = 0$. Similarly, the boundary conditions (6.64b) and (6.64c) imply that $\mathcal{L}^{\mathfrak{t}}_{\text{diffusion,T}}(\varphi) = \mathcal{L}^{\mathfrak{t}}_{\text{diffusion,b}}(\varphi) = 0$. Consequently, we see that $\mathcal{L}^{\mathfrak{t}}_{\text{diffusion}}(\varphi) = 0$.
For the converse direction, observe that $\mathcal{L}^{\mathfrak{t}}_{\text{diffusion}}(\varphi) = 0$ implies that

$$\varphi(X_{\mathcal{T}}, \mathcal{T}) = \varphi(X_{t_0}, t_0) + \int_{t_0}^{\mathcal{T}} \sigma^\top \nabla\varphi(X_s, s)\cdot \mathrm{d}W_s - \int_{t_0}^{\mathcal{T}} h(X_s, s, \varphi(X_s, s), \sigma^\top\nabla\varphi(X_s, s))\,\mathrm{d}s, \tag{6.88}$$

almost surely, and that the same holds with φ replaced by V. We proceed by defining the processes $\widetilde{Y}_s := \varphi(X_s, s)$ and $\widetilde{Z}_s := \sigma^\top\nabla\varphi(X_s, s)$, as well as $Y_s := V(X_s, s)$ and $Z_s := \sigma^\top\nabla V(X_s, s)$. By the assumptions on φ, b and σ, the processes Y, Z, $\widetilde{Y}$ and $\widetilde{Z}$ are progressively measurable with respect to the filtration generated by $(W_t)_{t\geq 0}$ and moreover square-integrable. Furthermore, the relation (6.88) shows that the pairs (Y, Z) and $(\widetilde{Y}, \widetilde{Z})$ satisfy a BSDE with terminal condition $\xi := \varphi(X_{\mathcal{T}}, \mathcal{T})$ on the random time interval $[t_0, \mathcal{T}]$. Well-posedness of the BSDE (see [228, Theorems 1.2 and 3.2]) implies that $Y = \widetilde{Y}$ and $Z = \widetilde{Z}$, almost surely. Conditional on t_0 and X_{t_0}, we also have $V(X_{t_0}, t_0) = Y^{X_{t_0}, t_0} = \widetilde{Y}^{X_{t_0}, t_0} = \varphi(X_{t_0}, t_0)$, where the superscripts denote conditioning on the initial time t_0 and corresponding initial condition X_{t_0}, see [228, Theorems 2.4 and 4.3]. Hence, we conclude that $\varphi = V$, $\nu \otimes \mu$-almost surely, and the result follows from the continuity of φ and V and the assumption that ν and μ have full support.

□

We have noted before that the diffusion loss combines ideas from the PINN and BSDE losses. In fact, it turns out that it can be interpreted as some kind of interpolation between the two. The following proposition makes this observation precise.

Proposition 6.25 (Relation of diffusion loss to PINN and BSDE losses)**.** *Let $\varphi \in \mathcal{F}$. We have*

$$\frac{\mathcal{L}^{\mathfrak{t}}_{\text{diffusion,int}}(\varphi)}{\mathfrak{t}^2} \to \mathcal{L}_{\text{PINN,int}}(\varphi), \tag{6.89}$$

as $\mathfrak{t} \to 0$. Moreover, for $t_0 = 0$ we have

$$\mathcal{L}^{\mathfrak{t}}_{\text{diffusion,int}}(\varphi) \to \mathcal{L}_{\text{BSDE}}(\varphi), \tag{6.90}$$

as $\mathfrak{t} \to \infty$.

Proof. Itô's formula shows that $\mathcal{L}^{\mathfrak{t}}_{\text{diffusion,int}}$ can be expressed as

$$\mathcal{L}^{\mathfrak{t}}_{\text{diffusion,int}}(\varphi) = \mathbb{E}\left[\left(\int_{t_0}^{\mathcal{T}} (\partial_s + L)\varphi(X_s, s)\,\mathrm{d}s + \int_{t_0}^{\mathcal{T}} h(X_s, s, \varphi(X_s, s), \sigma^\top \nabla\varphi(X_s, s))\,\mathrm{d}s\right)^2\right], \tag{6.91}$$

which implies the limit (6.89) by noting that $\mathcal{T} \to t_0$ as $\mathfrak{t} \to 0$. Relation (6.90) follows immediately from the definition of $\mathcal{L}_{\text{BSDE}}$ by noting that $\mathcal{T} \to \tau \wedge T$ as $\mathfrak{t} \to \infty$. □

6.3.4 Special PDE cases

We have formulated the boundary value problem (6.64) in a rather general form. In this section we shall mention two special cases.

Parabolic PDE on an unbounded domain

The general problem formulation in (6.64) allows for considering unbounded domains $\mathcal{D} = \mathbb{R}^d$, which makes the boundary condition (6.64c) obsolete and yields the parabolic terminal value problem

$$(\partial_t + L)V(x,t) + h(x,t,V(x,t),\sigma^\top \nabla V(x,t)) = 0, \qquad (x,t) \in \mathbb{R}^d \times [0,T), \tag{6.92a}$$

$$V(x,T) = f(x), \qquad x \in \mathbb{R}^d. \tag{6.92b}$$

Of course, in practice, we cannot sample data from the entire unbounded domain, but rather define some region of interest from which we sample and on which we aim to approximate the PDE.

Solving parabolic PDEs on unbounded domains via stochastic representations has been considered many times. Motivated by original attempts for the numerical approximation of BSDEs by backward-in-time iterations [39, 110, 228], those methods have been considered and further developed with neural networks in [11, 144] and endowed with tensor trains in [251], see also Section 6.2. As we have already mentioned in Remark 6.19, the original variational BSDE algorithm in [86] as well as a variant of it in [241] has been formulated for problems of type (6.92). Furthermore, linear parabolic PDEs on unbounded domains have for instance been approached via combining the Feynman-Kac formula with neural networks in [12, 29].

Elliptic boundary value problems

If we remove the time dependency from the solution we get the elliptic boundary value problem

$$LV(x) + h(x, V(x), \sigma^\top \nabla V(x)) = 0, \qquad x \in \mathcal{D}, \tag{6.93a}$$

$$V(x) = g(x), \qquad x \in \partial\mathcal{D}, \tag{6.93b}$$

where now $h \in C(\mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d, \mathbb{R})$. In analogy to (6.75), the corresponding backward equation can then be defined as

$$\mathrm{d}Y_s = -h(X_s, Y_s, Z_s)\mathrm{d}s + Z_s \cdot \mathrm{d}W_s, \qquad Y_\tau = g(X_\tau), \tag{6.94}$$

where $\tau = \{t > 0 : X_t \notin \mathcal{D}\}$ is the first exit time from the domain. Given suitable assumptions on h and assuming that τ is almost surely finite, one can show existence and uniqueness of solutions Y and Z, which, as before, represent the solution V and its gradient along trajectories of the forward process [228, Theorem 4.6]. Furthermore, Proposition 6.24 and its proof can straightforwardly be generalized to the elliptic setting, assuming that the stopping time τ is finite, almost surely. An algorithm for solving elliptic PDEs as in (6.93) in the spirit of the BSDE loss has been suggested in [179], using the same approximation ideas as in [86] (cf. Remark 6.19). We note that linear elliptic PDEs often admit alternative variational formulas via some sort of energy minimization [296]. An approach via the Feynman-Kac formula has been considered in [117].

6.3.5 Elliptic eigenvalue problems

We can extend our algorithmic approaches to eigenvalue problems of the form

$$LV(x) = \lambda V(x), \qquad x \in \mathcal{D}, \tag{6.95a}$$

$$V(x) = g(x), \qquad x \in \partial\mathcal{D}, \tag{6.95b}$$

which corresponds to choosing $h(x, y, z) = -\lambda y$ in the elliptic PDE (6.93). Note, however, that h now depends on the unknown eigenvalue $\lambda \in \mathbb{R}$. Furthermore, we can consider nonlinear eigenvalue problems of the form

$$LV(x) + h(x, V(x), \sigma^\top \nabla V(x)) = \lambda V(x), \qquad x \in \mathcal{D}, \tag{6.96a}$$
$$V(x) = g(x), \qquad x \in \partial\mathcal{D}, \tag{6.96b}$$

with $h \in C(\mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d, \mathbb{R})$.

For the linear problem (6.95) it is known that, given suitable assumptions, there exists a principal eigenvalue and that the corresponding eigenfunction is the only one that is positive on the entire domain $\mathcal{D}$ [25, Theorem 2.3]. This motivates us to consider the above losses now depending also on λ as well as enhanced with an additional term, and we define

$$\mathcal{L}^{\text{eigen}}(\varphi, \lambda) = \mathcal{L}_\lambda(\varphi) + \alpha_c \mathcal{L}_c(\varphi), \tag{6.97}$$

where $\mathcal{L}_\lambda(\varphi)$ is any of the losses from above, where $\mathcal{L}_c(\varphi) = (\varphi(x_c) - 1)^2$ with $x_c \in \mathcal{D}$ being somewhere in the center of the domain and where $\alpha_c > 0$ is an additional weight. The second term shall avoid finding the trivial solution that is zero everywhere, noting that V is often only defined up to a scalar factor (unless there is a fixed non-zero boundary condition). Often, one has periodic boundary conditions instead of an explicitly given boundary function g. In this case we can replace the loss term $\mathcal{L}_\text{b}$ for the boundary (e.g. in Definition 6.15 or Definition 6.22) by

$$\mathcal{L}_\text{b}(\varphi) = \mathbb{E}\left[\left(\varphi(X^b) - \varphi(\overline{X}^b)\right)^2\right] + \mathbb{E}\left[\left|\nabla\varphi(X^b) - \nabla\varphi(\overline{X}^b)\right|^2\right], \tag{6.98}$$

where $X^b \sim \mu(\partial\mathcal{D})$ is sampled randomly and $\overline{X}^b$ is its reflected/periodic counterpart. We note that in this scenario we cannot rely on the BSDE loss.

The idea is to constrain the function φ to be non-negative and to minimize $\mathcal{L}^{\text{eigen}}(\varphi, \lambda)$ w.r.t. $\varphi \in \mathcal{F}$ and $\lambda \in \mathbb{R}$ simultaneously. The following proposition shows that this is indeed a good idea for the approximation of the first eigenpair.

Proposition 6.26. *Let $\varphi \in \mathcal{F}$ with $\varphi \geq 0$ and assume that $\mathcal{L}_\lambda(\varphi) = 0$ if and only if (6.95) is satisfied. Then the following are equivalent:*

1. *φ is the principal eigenfunction for (6.95) with principal eigenvalue λ and normalization $\varphi(x_c) = 1$.*
2. *The pair (φ, λ) minimizes the loss (6.97), that is*

$$\mathcal{L}^{\text{eigen}}(\varphi, \lambda) = 0. \tag{6.99}$$

Remark 6.27. The assumption that $\mathcal{L}_\lambda(\varphi)$ is equivalent to (6.95) is satisfied for any 'reasonable' loss function. For the diffusion loss, Proposition 6.24 establishes this condition whenever the coefficients in (6.95) are regular enough.

Proof. It is clear that 1.) implies 2.) by the construction of (6.97). For the converse direction, notice that (6.99) implies $\varphi(x_c) = 1$ as well as (6.95), that is, φ is an eigenfunction with eigenvalue λ. In conjunction with the constraint $\varphi \geq 0$, it follows by [25, Theorem 2.3] that φ is the principal eigenfunction. □

A similar approach can be found in [124], where, however, the eigenvalue problem is connected to a parabolic PDE and is formulated as a fixed point problem.

6.3.6 From losses to algorithms

In this section we will discuss some details regarding the losses that we have introduced in Sections 6.3.1-6.3.3. We will show how they can be applied in practice and elaborate on certain implementational aspects. For convenience let us start by stating a prototypical algorithm that relies on the variational formulations that we have discussed before.

Algorithm 4: Approximation of solution V to boundary value problem (6.64)

Choose a parametrization $\mathbb{R}^p \ni \theta \mapsto \varphi_\theta$.
Initialize φ_θ (with a parameter vector $\theta \in \mathbb{R}^p$).
Choose an optimization method *descent*, a batch size $K \in \mathbb{N}$ and a learning rate $\eta > 0$. For PINN and diffusion losses choose weights $\alpha_1, \alpha_2, \alpha_3 > 0$ and batch sizes $K_b, K_T \in \mathbb{N}$ for the boundary terms. For BSDE and diffusion losses choose a step-size $\Delta t > 0$, for the diffusion loss choose a trajectory length $\mathrm{t} > 0$.
repeat
  Choose a loss function $\mathcal{L}$ of either (6.73), (6.78) or (6.84).
  Simulate data according to the chosen loss.
  Compute $\widehat{\mathcal{L}}(\varphi_\theta)$ as a Monte Carlo version of $\mathcal{L}$.
  Compute $\nabla_\theta \widehat{\mathcal{L}}(\varphi_\theta)$ using automatic differentiation.
  Update parameters: $\theta \leftarrow \theta - \eta\, descent(\nabla_\theta \widehat{\mathcal{L}}(\varphi_\theta))$.
until *convergence*;
Result: $\varphi_\theta \approx V$.

6.3.6.1 Training data and weights

The three losses that we have introduced differ in how training data is created. Our boundary problem (6.64) consists of three parts: the PDE (6.64a) that is defined inside the domain $\mathcal{D}$ as well as the two boundary conditions, in time (6.64b) and in space (6.64c). The general idea is to create sufficient artificial training data such that all corresponding parts of problem (6.64) can be learnt by minimizing the empirical versions of the losses evaluated on this data. In the PINN loss, domain data is sampled i.i.d. from a prescribed probability measure ν that has full support on the domain. While for ν taking the uniform distribution seems like an intuitive choice, alternative attempts seem promising and further research might focus on some sort of importance sampling with the goal to highlight certain more relevant regions of the domain, thereby potentially speeding up convergence or improving approximation accuracy in regions of interest. We note that it is mathematically not well understood why (uniformly) sampling from very high dimensional spaces seems to not make the algorithms suffer from the curse of dimensionality.
The BSDE and diffusion losses on the other hand rely on data that is generated by the forward SDE (6.66), implying that second derivatives as well as time derivatives do not have to be computed explicitly since they are approximated by the underlying Brownian motion. We will further elaborate on the simulation of diffusions in Section 6.3.6.2.

For the PINN and diffusion losses training data for the boundary terms is sampled explicitly from prescribed probability measures μ and λ, for which one often chooses uniform distributions in practice. The boundary data are incorporated into the losses as additional terms and one has to choose weights $\alpha_1, \alpha_2, \alpha_3 > 0$ for balancing the three loss parts. It is important to note that these choices are crucial, as in practice only certain weight configurations bring convergence to the right solution. Unfortunately it is generally not clear how to make appropriate choices a priori. For the PINN loss we refer to [288, 293, 294, 295] for some systematic insights and strategies, which might also be applied to the diffusion loss. For the BSDE loss on the other hand, boundary data is sampled implicitly by hitting the boundary with the underlying diffusion process. We will elaborate on this aspect in the upcoming section.

6.3.6.2 Simulation of diffusions and their exit times

The BSDE and diffusion losses rely on data from the stochastic process (6.66). In practice the SDE has to be approximated on a time grid $t_0 \le t_1 \le \cdots \le t_N$, for instance with the Euler-Maruyama scheme

$$\widetilde{X}_{n+1} = \widetilde{X}_n + b(\widetilde{X}_n, t_n)\Delta t + \sigma(\widetilde{X}_n, t_n)\xi_{n+1}\sqrt{\Delta t}, \tag{6.100}$$

or, to be precise, by its stopped version

$$\widehat{X}_{n+1} = \widehat{X}_n + \left(b(\widehat{X}_n, t_n)\Delta t + \sigma(\widehat{X}_n, t_n)\xi_{n+1}\sqrt{\Delta t}\right)\mathbb{1}_{\mathcal{C}_{n+1}} \tag{6.101}$$

with step condition $\mathcal{C}_n := \left(\widetilde{X}_n \in \mathcal{D}\right) \vee (t_n \le T)$ and time-increment $t_{n+1} = t_n + \Delta t\, \mathbb{1}_{\mathcal{C}_{n+1}}$, where $\Delta t > 0$ is the step-size and $\xi_{n+1} \sim \mathcal{N}(0, \mathrm{Id}_{d\times d})$ is a standard normally distributed random variable. We can then easily construct Monte Carlo versions of either the BSDE or the diffusion loss. For an example, the discrete version of the domain part of the diffusion loss (6.85a) reads

$$\widehat{\mathcal{L}}^{(K,N)}_{\text{diffusion,int}}(\varphi) = \frac{1}{K}\sum_{k=1}^{K}\Bigg(\varphi(\widehat{X}_N^{(k)}, t_N^{(k)}) - \varphi(\widehat{X}_0^{(k)}, t_0^{(k)}) - \sum_{n=0}^{N-1}\sigma^\top\nabla\varphi(\widehat{X}_n^{(k)}, t_n^{(k)})\cdot\xi_{n+1}^{(k)}\sqrt{\Delta t}\,\mathbb{1}_{\mathcal{C}_n^{(k)}} + \sum_{n=0}^{N-1} h\left(\widehat{X}_n^{(k)}, t_n^{(k)}, \varphi(\widehat{X}_n^{(k)}, t_n^{(k)}), \sigma^\top\nabla\varphi(\widehat{X}_n^{(k)}, t_n^{(k)})\right)\Delta t\,\mathbb{1}_{\mathcal{C}_n^{(k)}}\Bigg)^2, \tag{6.102}$$

where K is the sample size and $N = \frac{\mathrm{t}}{\Delta T}$ the maximal discrete trajectory length. The Monte Carlo version of the BSDE loss can be formed analogously.

It is known that, given suitable assumptions, the strong discretization errors of the forward and backward processes are of order $\sqrt{\Delta t}$ [172, 307], cf. also [123] for a numerical analysis on the original version of the BSDE loss. An additional challenge when considering bounded domains, however, is the approximation of exit times. In the BSDE loss boundary data is implicitly generated by the diffusion hitting the boundary at a random time τ. There are two problems that might occur here: On the one hand, exit times can be very large (e.g. if the domain is large or if the diffusion exhibits some metastable characteristics [26]), leading to very long runtimes of corresponding algorithms. One can try to counteract this phenomenon by adding an additional control to the forward process, however the choice of an adequate control seems to be non-trivial in practice (see Section 6.3.6.4 for further details). On the other hand, we note that any numerical scheme of SDE (6.66) leads to discretization errors not only of the process itself, but also of the exit times, leading to nontrivial effects and additional challenges at the boundary. In the two left panels of Figure 6.8 we illustrate this problem by displaying multiple "last positions" of an Euler-Maruyama discretization of Brownian motion as defined in (6.101) just before leaving the unit square using two different step-sizes. For our algorithms, all these points should in principle lie on the boundary, which in practice can only be achieved by choosing very small step-sizes, leading to additional computational challenges.

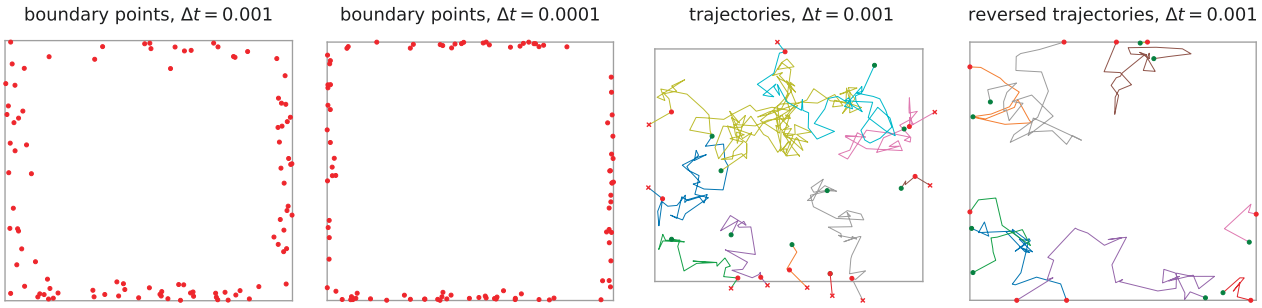


Figure 6.8: Illustration of the boundary data in the BSDE method.

The problem of discretizing exit times has been addressed multiple times and some improved sampling strategies have been suggested, see e.g. [42, 135]. For our problem, however, we should emphasize that it is not the principal goal to estimate exit times themselves, but it is rather of interest that trajectories are stopped accurately. We can therefore suggest the following two attempts that aim to improve the sampling of boundary data:

1. Rescaling: Start X_0 randomly in $\mathcal{D}$, simulate the trajectory and stop once the boundary has been crossed, however scale the last time step in such a way that the trajectory exactly ends on $\partial\mathcal{D}$.

2. Time-reversal: Start X_0 on the boundary $\partial\mathcal{D}$ and simulate the trajectory for a given time T (unless it hits the boundary again before time T, in this case stop the trajectory accordingly). Then reverse the process such that the reversed process ends on the boundary exactly.

An illustration of the two different strategies can be found in the two right panels of Figure 6.8.

Remark 6.28 (Backward iterations on bounded domains). We have mentioned in Remark 6.19 that an original attempt to solve BSDEs relies on backward iterations, however mostly formulated for unbounded domains. One can also try to solve BSDEs on bounded domains, incorporating random stopping times, by backward iteration algorithms. Of course the issue of approximating stopping times accurately remains the same and there is the additional challenge that trajectories have different lengths and therefore regression techniques might suffer from variance issues at the end of the trajectories, cf. [127] and Remark 6.8. Some numerical analysis in the context of parabolic PDEs on bounded domains has been done in [38], however numerical simulations are lacking and we are not aware of any rigorous error analysis or systematic numerical study related to this attempt.

6.3.6.3 Further modifications of the losses

The three losses that we have introduced in Sections 6.3.1-6.3.3 are natural candidates for solving the boundary value problem (6.64), differing however in how training data is generated. In Table 6.9a we contrast those different data generation attempts. We have discussed before that each of the losses has advantages and potential drawbacks. In Table 6.9b we summarize some of them.

Table 6.9: Comparison of the different losses.

	PINN	BSDE	Diffusion
SDE simulation		✗	✗
boundary data	✗		✗

(a) The three losses can be characterized by how training data is generated.

	PINN	BSDE	Diffusion
Hessian computations	✗		
boundary issues		✗	
weight tuning	✗		✗
long runtimes		✗	
discretization		✗	✗

(b) In this table we list potential challenges and drawbacks for the corresponding losses.

In the following we shall discuss certain modifications for some of the losses, relating also to versions that have appeared in the literature before.

6.3.6.4 Forward control

We can modify the SDE-based losses by adding control functions $v \in C(\mathbb{R}^d \times [0,T], \mathbb{R}^d)$ to the forward process (6.66), yielding the controlled diffusion

$$\mathrm{d}X_s^v = (b(X_s^v, s) + \sigma(X_s^v, s)v(X_s^v, s))\,\mathrm{d}s + \sigma(X_s^v, s)\,\mathrm{d}W_s. \tag{6.103}$$

By applying Itô's formula we can get appropriate losses similar to the ones from before. For instance, the diffusion loss can then be written as

$$\begin{aligned}\mathcal{L}^{\mathrm{t},v}_{\mathrm{diffusion,int}}(\varphi) = \mathbb{E}\Bigg[\Bigg(&\varphi(X_{\mathcal{T}}^v, \mathcal{T}) - \varphi(X_{t_0}^v, t_0) - \int_{t_0}^{\mathcal{T}} \sigma^\top \nabla\varphi(X_s, s)\cdot \mathrm{d}W_s \\ &+ \int_{t_0}^{\mathcal{T}} \left[h(X_s^v, s, \varphi(X_s^v, s), \sigma^\top\nabla\varphi(X_s^v, s)) - v(X_s^v, s)\cdot\sigma^\top\nabla\varphi(X_s^v, s)\right]\mathrm{d}s\Bigg)^2\Bigg]\end{aligned} \tag{6.104}$$

instead of (6.85a), where we note that we actually have a family of losses parametrized by v since Proposition 6.24 holds true for $\mathcal{L}^{\mathrm{t},v}_{\mathrm{diffusion}}$ with any v.

The same considerations are true for the BSDE loss, noting that with the generalized BSDE system

$$\mathrm{d}X_s^v = (b(X_s^v, s) + \sigma(X_s^v, s)v(X_s^v, s))\,\mathrm{d}s + \sigma(X_s^v, s)\,\mathrm{d}W_s, \qquad X_{t_0}^v = x_{\mathrm{init}}, \tag{6.105a}$$

$$\mathrm{d}Y_s^v = -h(X_s^v, s, Y_s^v, Z_s^v)\,\mathrm{d}s + v(X_s^v, s)\cdot Z_s^v\,\mathrm{d}s + Z_s^v\cdot\mathrm{d}W_s, \qquad Y_T^v = k(X_{T\wedge\tau}^v, T\wedge\tau), \tag{6.105b}$$

we still have the relations

$$Y_s^v = V(X_s^v, s), \qquad Z_s^v = \sigma^\top\nabla V(X_s^v, s) \tag{6.106}$$

for any suitable $v \in C(\mathbb{R}^d \times [0,T], \mathbb{R}^d)$, analog to (6.76), see also Corollary 2.28. This immediately brings the family of losses

$$\begin{aligned}\mathcal{L}^v_{\mathrm{BSDE}}(\varphi) = \mathbb{E}\Bigg[\Bigg(&f(X_{\tau\wedge T}^v)\mathbb{1}_{\tau\wedge T = T} + g(X_{\tau\wedge T}^v, \tau\wedge T)\mathbb{1}_{\tau\wedge T=\tau} - \varphi(X_{t_0}^v, t_0) - \int_{t_0}^{\tau\wedge T}\sigma^\top\nabla\varphi(X_s^v, s)\cdot\mathrm{d}W_s \\ &+ \int_{t_0}^{\tau\wedge T}\left(h(X_s^v, s, \varphi(X_s, s), \sigma^\top\nabla\varphi(X_s, s)) - v(X_s^v, s)\cdot\sigma^\top\nabla\varphi(X_s^v, s)\right)\mathrm{d}s\Bigg)^2\Bigg],\end{aligned} \tag{6.107}$$

again parametrized by $v \in C(\mathbb{R}^d \times [0,T], \mathbb{R}^d)$.

Adding a control to the forward process can be understood as driving the data generating process into regions of interest, which can be seen as importance sampling of diffusions, see also Section 2.3. We have noted before that a challenge in the BSDE loss is that exit times might be large. One idea is therefore to aim for controls v that decrease such exit times, while still allowing for low-variance estimators of the losses and their gradients. How to identify such suitable forward controls might be an interesting topic for future research (we recall Chapter 4 for some systematic approaches in this respect relating to Hamilton-Jacobi-Bellman PDEs).

6.3.6.5 Approximating the gradient of the solution

Inspecting the BSDE system (6.75) we realize that we can as well consider losses that are slightly different from the BSDE loss as stated in Definition 6.17. Going back to [86], we can for instance use the fact that the backward process Y can be written in a forward way, yielding the discrete process

$$\widehat{Y}_{n+1} = \widehat{Y}_n - h(\widehat{X}_n, t_n, \widehat{Y}_n, \widehat{Z}_n)\Delta t + \widehat{Z}_n \cdot \xi_{n+1}\sqrt{\Delta t}. \tag{6.108}$$

We realize that this scheme is explicit and the only unknowns are $\widehat{Y}_0$ and $\widehat{Z}_n$ for $n \in \{0, \dots, N-1\}$. This motivates to learn the single parameter $y_0 \approx \widehat{Y}_0 \in \mathbb{R}$ and the functions $\phi \approx \sigma^\top \nabla V \in C(\mathbb{R}^d \times [0,T], \mathbb{R}^d)$ (rather than V directly). This approach can be summarized in the loss

$$\begin{aligned}\mathcal{L}_{\mathrm{BSDE-2}}(\phi, y_0) = \mathbb{E}\Bigg[\Bigg(f(X_{\tau\wedge T})\mathbb{1}_{\tau\wedge T=T} + g(X_{\tau\wedge T}, \tau\wedge T)\mathbb{1}_{\tau\wedge T=\tau} - y_0 - \int_0^{\tau\wedge T} \phi(X_s, s)\cdot \mathrm{d}W_s \\ + \int_0^{\tau\wedge T} h(X_s, s, Y_s, \phi(X_s, s))\mathrm{d}s\Bigg)^2\Bigg].\end{aligned} \tag{6.109}$$

In this setting X_0 has to be chosen deterministically and we note that we can only hope to approximate $y_0 \approx V(X_0, 0)$ as well as the gradient of V along the trajectories of the forward process. We have shown in Chapter 4 that one can consider alternative losses (like the log-variance loss) whenever the nonlinearity h only depends on the solution through its gradient, in which case the extra parameter y_0 can be omitted.

6.3.6.6 Penalizing deviations from the discrete scheme

Another approach that is grounded in the discrete backward process has been suggested in [241] for problems on unbounded domains. It relies on the idea that for each $n \in \{0, \dots, N-1\}$ we can penalize deviations from (6.108) (cf. also [144] and Section 6.2, where however an implicit scheme and backward iterations are used). Noting that we aim for $\widehat{Y}_n \approx \varphi(\widehat{X}_n, t_n), \widehat{Z}_n \approx \sigma^\top \nabla\varphi(\widehat{X}_n, t_n)$, this motivates the loss

$$\widehat{\mathcal{L}}^{(K,N)}_{\mathrm{BSDE-3}}(\varphi) = \alpha_1 \widehat{\mathcal{L}}^{(K,N)}_{\mathrm{BSDE-3,int}}(\varphi) + \alpha_2 \widehat{\mathcal{L}}^{(K,N)}_{\mathrm{BSDE-3,b}}(\varphi) \tag{6.110}$$

with the interior part

$$\begin{aligned}\widehat{\mathcal{L}}^{(K,N)}_{\mathrm{BSDE-3,int}}(\varphi) = \frac{1}{K}\sum_{k=1}^{K}\sum_{n=0}^{N-1}\Bigg(\varphi(\widehat{X}^{(k)}_{n+1}) - \varphi(\widehat{X}^{(k)}_n) + h\Big(\widehat{X}^{(k)}_n, \varphi(\widehat{X}^{(k)}_n), \sigma^\top\nabla\varphi(\widehat{X}^{(k)}_n)\Big)\Delta t \\ - \sigma^\top\nabla\varphi(\widehat{X}^{(k)}_n)\xi_{n+1}\sqrt{\Delta t}\Bigg)^2\end{aligned} \tag{6.111}$$

and the boundary term

$$\widehat{\mathcal{L}}^{(K,N)}_{\mathrm{BSDE-3,b}}(\varphi) = \frac{1}{K}\sum_{k=1}^{K}\Big(\varphi(\widehat{X}^{(k)}_N) - g(\widehat{X}^{(k)}_N)\Big)^2. \tag{6.112}$$

A generalization to bounded problems is straightforward and we note that it is not possible to write down a continuous version of (6.110). Interesting, however, is that we can relate this loss to the diffusion loss via Jensen's inequality, yielding

$$\widehat{\mathcal{L}}^{(K,N)}_{\mathrm{diffusion,int}}(\varphi) \le N\widehat{\mathcal{L}}^{(K,N)}_{\mathrm{BSDE-3,int}}(\varphi). \tag{6.113}$$

Yet another approach that is based on a discrete backward scheme is the following. Let us initialize $\widehat{Y}_0 = \varphi(\widehat{X}_0, 0)$ and simulate

$$\widehat{Y}_{n+1} = \widehat{Y}_n - h(\widehat{X}_n, \widehat{Y}_n, \sigma^\top\nabla\varphi(\widehat{X}_n, t_n))\Delta t + \sigma^\top\nabla\varphi(\widehat{X}_n, t_n)\cdot\xi_{n+1}\sqrt{\Delta t} \tag{6.114}$$

for $n \in \{0, \dots, N-1\}$, where in contrast to the previous attempt we have now only replaced $\widehat{Z}_n$ by its approximation $\sigma^\top \nabla\varphi(\widehat{X}_n, t_n)$ and take $\widehat{Y}_n$ from previous iteration steps, which is in spirit similar to the attempt in $\mathcal{L}_{\mathrm{BSDE}-2}$. Following the motivation of $\mathcal{L}_{\mathrm{BSDE}-3}$, i.e. penalizing deviations from the discrete scheme, we can now introduce the loss

$$\widehat{\mathcal{L}}^{(K,N)}_{\mathrm{BSDE}-4}(\varphi) = \frac{\alpha_1}{K}\sum_{k=1}^{K}\sum_{n=0}^{N}\left(\varphi(\widehat{X}_n^{(k)}, t_n) - \widehat{Y}_n^{(k)}\right)^2 + \frac{\alpha_2}{K}\sum_{k=1}^{K}\left(\varphi(X_b^{(k)}) - g(X_b^{(k)})\right)^2. \tag{6.115}$$

Let us note that in both $\mathcal{L}_{\mathrm{BSDE}-3}$ and $\mathcal{L}_{\mathrm{BSDE}-4}$ we can replace the deterministic initial point $\widehat{X}_0$ at $t = 0$ by random choices $\widehat{X}_{t_0} \sim \nu$ at random times $t_0 \sim \mu$, where ν and μ are prescribed probability measures, adjusting the sums in (6.111) and (6.115) accordingly.

6.3.7 Numerical experiments

In this section we will provide several numerical examples of high-dimensional parabolic and elliptic PDEs that shall demonstrate the performances of Algorithm 4 using the different loss functions that we have discussed before. We will focus on the three losses from Sections 6.3.1-6.3.3 since their modified versions from Section 6.3.6.3 did not yield better performances consistently.

6.3.7.1 Computational aspects and function approximations

We usually rely on neural networks for our approximating function φ, where we note that its derivatives can be computed by autodifferentiation tools. We should note however that computing second derivatives with tools such as PyTorch or Tensorflow can be expensive, especially if the state space dimension d is large. In particular, if for the differential operator as defined in (6.65) we need to compute the full Hessian matrix of φ, which is the case if σ is non-diagonal, d^2 partial derivatives need to be computed, which can be costly even for state-of-the art software packages. Here the BSDE and diffusion losses have a potential advantage since they do not rely on an explicit computation of the Hessian, but rather approximate second derivatives by the underlying Brownian motion.

In our experiments we will rely on standard feed-forward neural networks and on the DenseNet as defined in Section 2.4. For the optimization of the losses we rely on the Adam optimizer [169]. If not specified otherwise, we take a DenseNet with ReLU activation function and four hidden layers with $d + 20, d, d, d$ hidden units respectively for the approximating function φ and a learning rate $\eta = 0.001$ for the optimization. We take $K = 200$ samples inside the domain and $K = 50$ on the boundary. For the SDE discretization we usually choose a step-size of $\Delta t = 0.001$. For the PINN and diffusion losses we try out different weight configurations beforehand and decide for best.

6.3.7.2 Elliptic problem with Dirichlet boundary data

Let us start with a nonlinear toy problem for which we can test different PDE settings by comparing our approximations against available analytical reference solutions. Here we define the domain to be the unit ball $\mathcal{D} = \{x \in \mathbb{R}^d : |x| < 1\}$. Let us first consider an elliptic boundary value problem as defined in (6.93). Let $\gamma \in \mathbb{R}$ and choose

$$b(x,t) = \mathbf{0}, \qquad \sigma(x,t) = \sqrt{2}\,\mathrm{Id}_{d\times d}, \qquad g(x) = e^{\gamma}, \tag{6.116a}$$

$$h(x,y,z) = -2\gamma y(\gamma|x|^2 + d) + \sin\left(e^{2\gamma|x|^2} - y^2\right). \tag{6.116b}$$

One can readily check that

$$V(x) = e^{\gamma|x|^2} \tag{6.117}$$

is the solution to (6.93).

We consider $d = 50$ and choose $\gamma = 1$. For the PINN and diffusion losses we decide for the weights $\alpha = (10^{-5}, 1)$ and $\alpha = (0.1, 1)$ respectively. We sample the data uniformly and take a maximal trajectory length of $N = 20$ for the diffusion loss. In Figure 6.9 we display the average relative errors $\frac{|\varphi(x) - V(x)|}{V(x)}$ as a function of $r = |x|$ in the left panel, noting that most samples are placed close to the boundary of the ball. In the right panel we display the L^2 error during the training iterations evaluated on uniformly sampled test data. We can see that PINN and diffusion losses yield similar results and that the BSDE loss performs worse in particular close to the boundary, which might be due to the hitting time discretizations as described in Section 6.3.6.2.

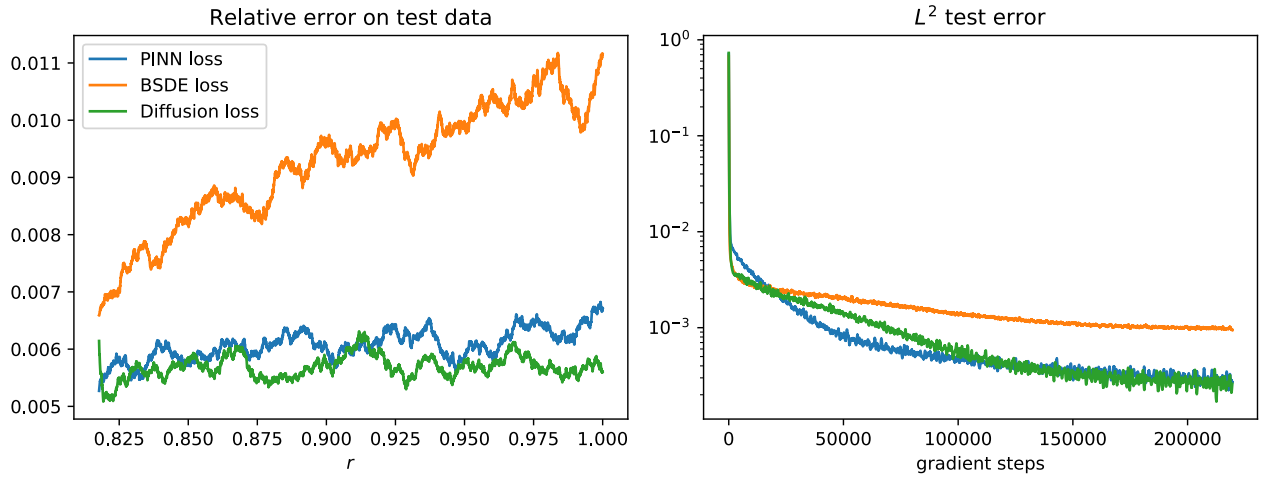


Figure 6.9: Left: Average relative errors as a function of $r = |x|$ evaluated on uniformly sampled data for the three losses smoothed with a moving average over 500 data points. Right: L^2 error during the training iterations evaluated on uniformly sampled test data.

6.3.7.3 Elliptic problem requiring full Hessian matrix

Let us consider the same problem as before, but now replace the diffusion coefficient and the nonlinearity by

$$\sigma(x,t) = \sqrt{\frac{2}{d}} \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix}, \qquad h(x,y,z) = -2\gamma y \left(\gamma \sum_{i,j=1}^{d} x_i x_j + d \right) + \sin\left(e^{2\gamma|x|^2} - y^2 \right), \tag{6.118}$$

respectively. We can check that $V(x) = e^{\gamma|x|^2}$ is still the solution to the corresponding boundary problem. Since σ is not diagonal anymore, inspecting the differential operator (6.65) shows that now the full Hessian matrix of the solution is present in the PDE. We have discussed already in Section 6.3.7.1 that this particularly impacts the runtime of the PINN method since all derivatives need to be computed explicitly. For the BSDE and diffusion losses, on the other hand, second derivatives are approximated by the underlying Brownian motion and we therefore do not expect longer runtimes in those cases.

Let us consider $d = 20$ and $\gamma = 1$. In Figure 6.10 we display the L^2 error during the training process, once plotted against the gradient steps and once plotted against the runtime. We can see that the PINN loss takes significantly longer, as expected. This effect should become even more severe with growing state space dimension.

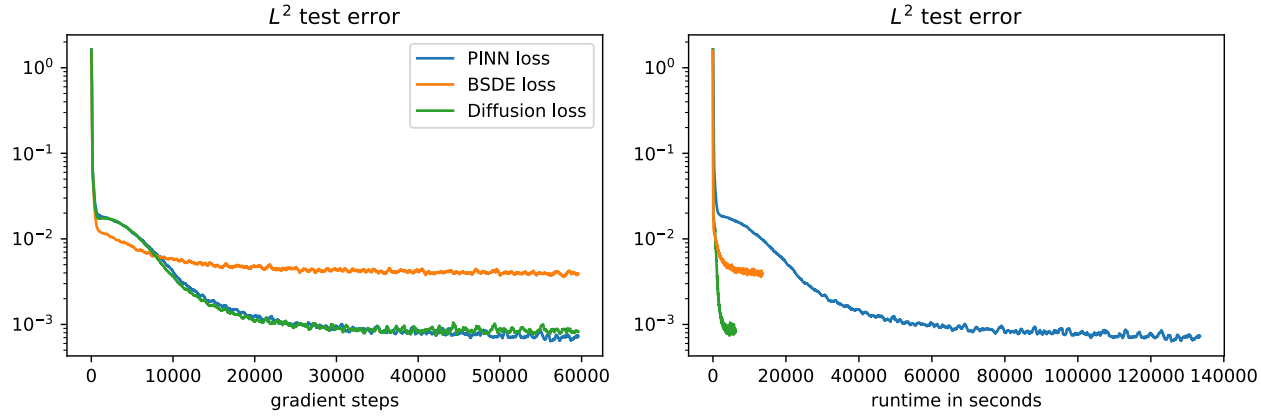


Figure 6.10: L^2 error during the training process evaluated on test data for the three losses, once plotted against the gradient steps and once against the runtime.

6.3.7.4 Parabolic problem with Neumann boundary data

Let us now consider a parabolic problem with Neumann instead of Dirichlet data for the spatial boundary. To this end, we refer to problem (6.64) with the boundary term (6.64c) replaced by a Neumann condition (as described in Remark 6.14) and take

$$b(x,t) = \mathbf{0}, \qquad \sigma(x,t) = \sqrt{2}\,\mathrm{Id}_{d\times d}, \qquad f(x) = e^{\gamma|x|^2+T}, \qquad g^N(x,t) = 2\gamma e^{\gamma+t}, \tag{6.119a}$$

$$h(x,t,y,z) = -y(2\gamma(2\gamma|x|^2+d)+1) + \sin\left(e^{2\gamma|x|^2+2t} - y^2\right). \tag{6.119b}$$

We can check that now

$$V(x,t) = e^{\gamma|x|^2+t} \tag{6.120}$$

solves our problem.

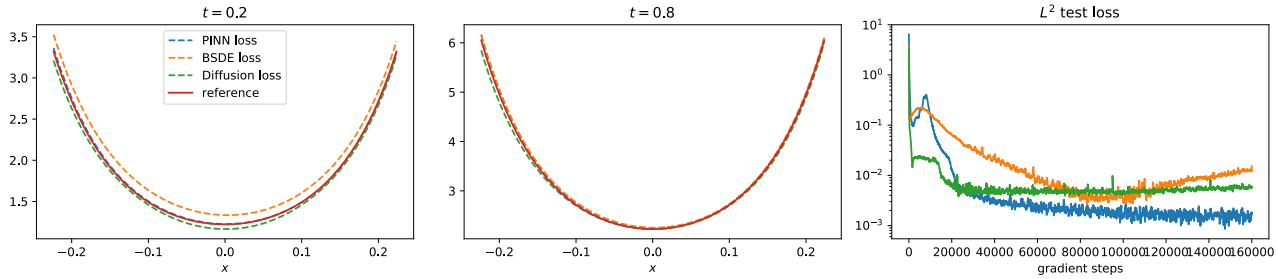


Figure 6.11: Left and central panel: Approximations along a curve for two different times using the three losses. Right: L^2 test error along the training iterations.

We choose $d = 20$ and $\gamma = 1$. In the left and central panels of Figure 6.11 we display the approximated solutions along the curve $\left\{(x,\dots,x)^\top : x \in [-1/\sqrt{d}, 1/\sqrt{d}]\right\}$ for two different times. We can see that the BSDE loss seems to be slightly worse than the other two losses, with small advantages for the PINN loss. The right panel displays the L^2 test error over the iterations and confirms this observation. We note that in fact the BSDE loss holds the disadvantage that due to a potential exit before time T only few data points might be available for fitting the function f, which might also explain the instability of this loss that can be observed in the right panel.

6.3.7.5 Dependence on the trajectory length

In the diffusion loss as stated in Definition 6.22 we are free to choose the length $\mathfrak{t}$ of the forward trajectories, which affects the generated training data. Let us therefore investigate how different choices of $\mathfrak{t}$ influence the performance of Algorithm 4. To this end, we consider again the elliptic problem from Section 6.3.7.2, now in $d = 10$, and vary $\mathfrak{t}$. To be precise, let us fix different step-sizes Δt and vary the Euler steps N (recalling that $\mathfrak{t} = N\Delta t$). As displayed in Figure 6.12, it turns out that there seems to be an optimal choice.

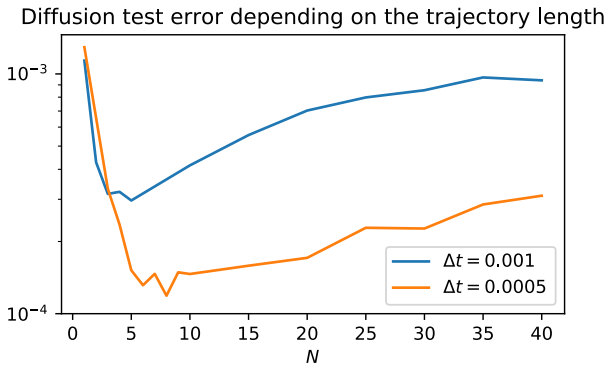


Figure 6.12: We display the L^2 error that one attains when using different choices of the maximal Euler steps N in the diffusion loss for different discretization step-sizes Δt.

6.3.7.6 Committor functions

Committor functions are important quantities in molecular dynamics as they specify transition pathways and transition rates between (potentially metastable) regions of interest [84, 201]. Since for most interesting applications those functions are high-dimensional and hard to compute, there have been recent attempts to approach this problem with neural networks [166, 193, 254]. Since committor functions fulfill elliptic boundary value problems, we can rely on the methods that we have discussed in this section.

The committor function is defined as

$$V(x) = \mathbb{P}(\tau_B < \tau_A | X_0 = x) = \mathbb{E}[\mathbb{1}_B(X_\tau)|X_0 = x], \tag{6.121}$$

where X is a stochastic process as defined in (6.66), $A, B \subset \mathbb{R}^d$ are given sets and $\tau_A = \inf\{t > 0 : X_t \in A\}$ and $\tau_B = \inf\{t > 0 : X_t \in B\}$ are corresponding hitting times, denoting $\tau = \min\{\tau_A, \tau_B\}$. V is therefore the probability of hitting set A before B and via the Kolmogorov backward PDE it fulfills the elliptic boundary value problem

$$LV = 0, \qquad V_{|\partial A} = 0, \qquad V_{|\partial B} = 1. \tag{6.122}$$

In the notation of (6.93) we have $\mathcal{D} = \mathbb{R}^d \setminus (A \cap B), h = 0$ and $g(x) = \mathbb{1}_B(x)$.

Let us construct the following example for which a reference solution can be computed analytically [127]. We take standard Brownian motion $X_t = x + W_t$, i.e. $b = \mathbf{0}$ in (6.65), and are interested in leaving a domain through either of two surrounding spheres. To be precise, we define the two sets

$$A = \{x \in \mathbb{R}^d : |x| < a\}, \qquad B = \{x \in \mathbb{R}^d : |x| > b\} \tag{6.123}$$

with $b > a > 0$. An analytic solution can be computed as

$$V(x) = \frac{a^2 - |x|^{2-d}a^2}{a^2 - b^{2-d}a^2} \tag{6.124}$$

for any $d \geq 3$. Let us consider $d = 10$ and choose $a = 1, b = 2$. We take a DenseNet with tanh as an activation function and compare our three losses against each other. In Figure 6.13 we display the approximated solutions along a curve $\left\{(x, \dots, x)^\top : x \in [a/\sqrt{d}, b/\sqrt{d}]\right\}$ in the left panel, realizing that in particular the PINN and diffusion losses lead to good approximations. This can also be observed in the right panel, where we plot a moving average of the L^2 error on test data with moving window of 200. It seems that the diffusion loss is more stable than the PINN and BSDE losses.

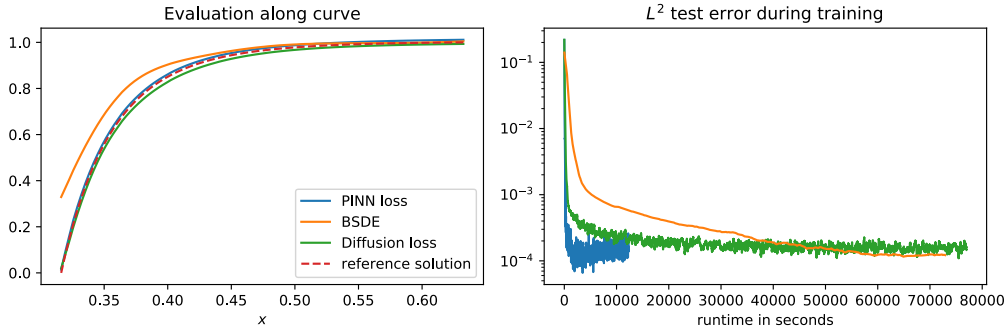


Figure 6.13: Left: approximations of the 10-dimensional committor function evaluated along a curve. Right: moving average of the test L^2 error along the training iterations.

We should note that an evaluation along a curve as specified above is slightly unfair since uniformly sampled data barely lies close to the left endpoint of this curve. Realizing that the solution (6.124) is radial symmetric, i.e. depends on x only through its distance to the origin denoted by $r = |x|$, we sample 10000 data points uniformly on $\mathcal{D}$ and plot the evaluation of the approximating function φ along these points in Figure 6.14. We can again see the superiority of the PINN and diffusion loss approximations, noticing in particular that the BSDE loss approximation gets worse closer to the boundary, which might be due to the numerical challenges that we described in Section 6.3.6.2.

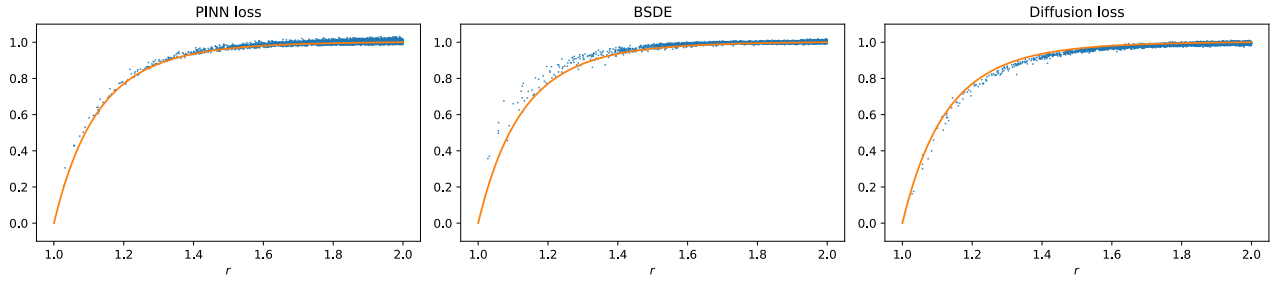


Figure 6.14: We plot the approximated committor functions evaluated at 10000 points uniformly sampled from the domain $\mathcal{D}$ (blue dots) and compare those to the reference solution (orange line) as a function of $r = |x|$.

6.3.7.7 Parabolic Allen-Cahn equation on an unbounded domain

The Allen-Cahn equation in $d = 100$ has been suggested as a benchmark problem in [86]. It is an example for a parabolic PDE on an unbounded domain, compatible with the setting that we have specified in (6.92):

$$(\partial_t + L)V(x,t) + V(x,t) - V^3(x,t) = 0, \qquad (x,t) \in \mathbb{R}^d \times [0,T), \tag{6.125a}$$

$$V(x,T) = f(x), \qquad x \in \mathbb{R}^d, \tag{6.125b}$$

with $f(x) = \left(2 + \frac{2}{5}|x|^2\right)^{-1}, T = \frac{3}{10}$. In order to use Algorithm 4, we nevertheless need to define a domain on which we aim to approximate the solution. Let us choose a ball $\mathcal{D} = \{x \in \mathbb{R}^d : |x| < R\}$, where we take the radius $R = 7$. Instead of sampling uniformly on the domain, we consider sampling uniformly on a box around the origin with side length 2 and multiply each data point x by $\frac{R}{|x|}$. In contrast to the uniform sampling this approach generates more samples close to the origin. We compare our approximations with a reference solution for $x_0 = (0, \dots, 0)^\top$ for different times $t \in [0, T]$ that is provided by a branching diffusion method specified in [86]. In Figure 6.15 we see that all three attempts match this reference solution, with minor advantages of the PINN and diffusion losses. In Table 6.10 we display the computation times needed until reaching convergence of the corresponding algorithms, realizing that the BSDE loss needs significantly longer. We note that the computation times are larger in comparison to e.g. [86] since we aim for a solution on a given domain, whereas other attempts only strive to approximate the solution at a single point.

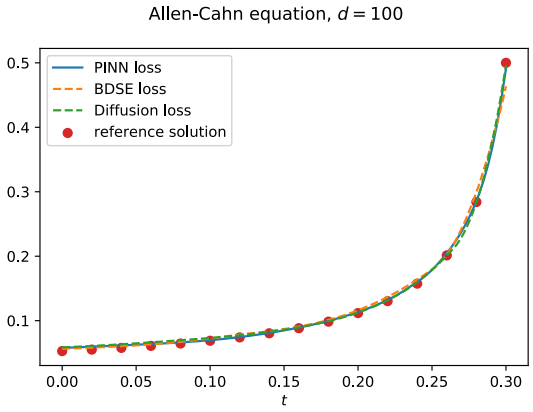


Figure 6.15: Approximation of the solution to an Allen-Cahn equation in $d = 100$ using different losses compared to a referecene solution at $x_0 = (0, \dots, 0)^\top$ for different times $t \in [0, T]$.

Computation time	
PINN loss	325.46 min
BSDE loss	4280.68 min
Diffusion loss	194.38 min

Table 6.10: Computation times needed until the algorithms converged.

6.3.7.8 Elliptic eigenvalue problems

In this section we provide two examples for the approximation of principal eigenvalues and corresponding eigenfunctions. The first one is a linear problem and therefore Proposition 6.26 assures that the minimization of a corresponding loss as in (6.97) is feasible. The second example is a nonlinear eigenvalue problem, for which we can numerically show that our algorithm still provides the correct solution.

Fokker-Planck equation

As suggested in [124] let us aim at computing the principal eigenpair of the linear Fokker-Planck operator, which is defined through acting on a function $V : \mathcal{D} \to \mathbb{R}$ via

$$-\Delta V - \nabla \cdot (V \nabla \Psi) \tag{6.126}$$

on the domain $\mathcal{D} = (0, 2\pi)^d$, where $\Psi(x) = \sin\left(\sum_{i=1}^d c_i \cos(x_i)\right)$ is a potential with $c_i \in [0.1, 1]$, assuming periodic boundary conditions. This results in solving the eigenvalue problem

$$\Delta V(x) + \nabla \Psi(x) \cdot \nabla V(x) + \Delta \Psi(x) V(x) = -\lambda V(x) \tag{6.127}$$

and one can readily check that

$$V(x) = e^{-\Psi(x)} \tag{6.128}$$

is an eigenfunction to the principal eigenvalue $\lambda = 0$. We choose $c_i = 0.1$ and approach this problem in dimension $d = 5$ following Section 6.3.5, i.e. by minimzing the loss (6.97), where for $\mathcal{L}$ we choose the diffusion loss and the periodic boundary condition is encoded via the term (6.98). Here and in the following eigenvalue problems the positivity of the approximating function is achieved by adding a ReLU function after the last layer of the DenseNet.

In Figure 6.16 we display the approximated eigenfunction evaluated along the curve $\{(x, \dots, x)^\top : x \in [0, 2\pi]\}$ in the left panel and compare it to the reference solution. In the central panel we show the L^2 error w.r.t. the reference solution evaluated on uniformly sampled test data along the training iterations. The right panel displays the moving average of the absolute value of the eigenvalue with a moving window of 100 gradient steps (since the true value is $\lambda = 0$ it is not feasible to compute a relative error here). We see that both the eigenfunction and the eigenvalue are approximated sufficiently well.

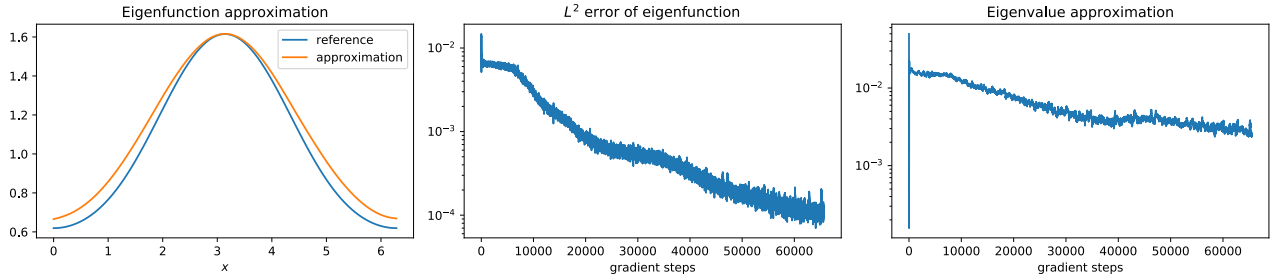


Figure 6.16: Left: Approximation and reference of the eigenfunction corresponding to the principal eigenvalue of the Fokker-Planck operator. Middle: L^2 error w.r.t. test data along the training iterations. Right: Moving average of the absolute value of the approximated eigenvalue along the gradient steps.

Nonlinear Schrödinger equation

Let us now consider a nonlinear eigenvalue problem. Again following [124], we take the nonlinear Schrödinger operator including a cubic term that arises from the Gross-Pitaevskii equation for the single-particle wave function in a Bose-Einstein condensate. To be precise, we consider

$$\Delta V(x) - V^3(x) - \Psi(x) V(x) = -\lambda V(x) \tag{6.129}$$

with

$$\Psi(x) = -\frac{1}{c^2} \exp\left(\frac{2}{d} \sum_{i=1}^d \cos x_i\right) + \sum_{i=1}^d \left(\frac{\sin^2(x_i)}{d^2} - \frac{\cos x_i}{d}\right) - 3, \tag{6.130}$$

again on the domain $\mathcal{D} = (0, 2\pi)^d$. One can show that

$$V(x) = \frac{1}{c} \exp\left(\frac{1}{d} \sum_{i=1}^d \cos x_i\right) \tag{6.131}$$

is the eigenfunction corresponding to the principal eigenvalue $\lambda = -3$, where c is chosen such that $\int_{\mathcal{D}} V^2(x) \mathrm{d}x = |\mathcal{D}|$. We add the latter constraint into the loss function by replacing the term $\mathcal{L}_c(\varphi)$ in (6.97) with $\mathcal{L}_n(\varphi) = \left(\mathbb{E}\left[\varphi(X)^2\right] - 1\right)^2$, where X is sampled uniformly on $\mathcal{D}$. Figure 6.17 shows the approximate solution of the

eigenfunction in $d = 5$ evaluated along the curve $\{(x, \dots, x)^\top : x \in [0, 2\pi]\}$ as well as its L^2 error and the relative error of the approximated eigenvalue along the training iterations. We see that we can approximate both the eigenfunction and the eigenvalue quite well.

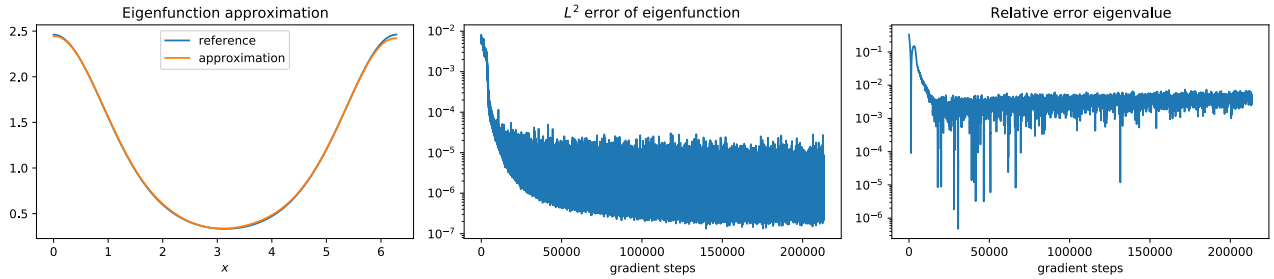


Figure 6.17: Left: Approximation and reference of the eigenfunction corresponding to the principal eigenvalue of the nonlinear Schrödinger operator in $d = 5$. Middle: L^2 error w.r.t. test data along the training iterations. Right: Relative error of the approximated eigenvalue along the gradient steps.

We repeat the experiment in dimension $d = 10$ and display the results in Figure 6.18. The optimization task gets slightly more difficult, but the resulting eigenfunction and eigenvalue still fit the reference solution sufficiently well. Note that in both experiments no explicit boundary conditions, but only the norm constraint and the periodicity are provided.

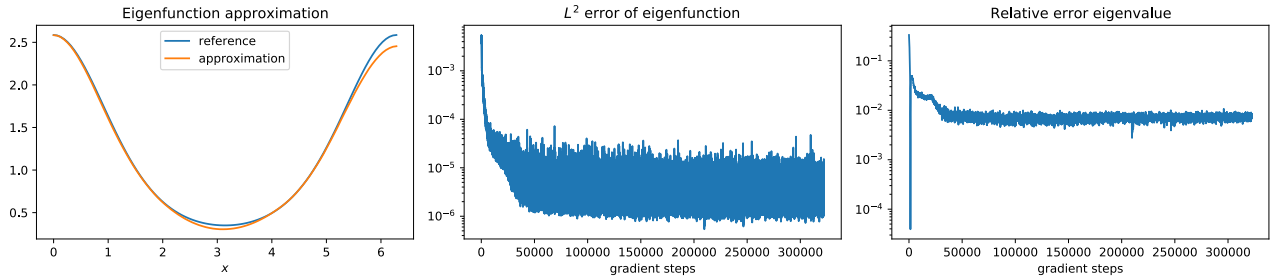


Figure 6.18: Same experiment as in Figure 6.17 in dimension $d = 10$.

6.3.7.9 Heat equation as an example for a linear PDE

As proposed in [12], let us consider the linear PDE

$$(\partial_t + \Delta)V(x,t) = 0, \qquad (x,t) \in \mathbb{R}^d \times [0,T), \tag{6.132a}$$

$$V(x,T) = g(x), \qquad x \in \mathbb{R}^d, \tag{6.132b}$$

with terminal condition $g(x) = |x|^2$ at $T = 1$. A solution can be readily computed to be

$$V(x,t) = |x|^2 + 2d(T-t). \tag{6.133}$$

We shall take this problem in order to evaluate the difference between the L^2 projection loss from Section 6.1 and the BSDE loss from Definition 6.17, cf. Remark 6.21. For the training algorithm we sample data uniformly from the domain $\mathcal{D} = \{x \in \mathbb{R}^d : |x| < 1\}$ with $d = 10$ and consider $K = 200$ data points for each gradient computation. We start with a step-size of $\Delta t = 0.01$ for the discretization of the stochastic processes and switch to $\Delta t = 0.001$ when the loss does not decrease anymore. In Figure 6.19 we display the smoothed L^2 test error as a function of the runtime as well as the approximated functions compared with the reference solution for two different points in time along the curve $\left\{(x, \dots, x)^\top : x \in [-1/\sqrt{d}, 1/\sqrt{d}]\right\}$. The gradient computations of the BSDE loss are expected to take longer due to the additional appearance of the approximating function. Still, we can see that the BSDE loss significantly outperforms the L^2 projection approach, indicating that including the Itô integral seems to bring numerical advantages, which can also be compared to the results in Section 6.2.3.

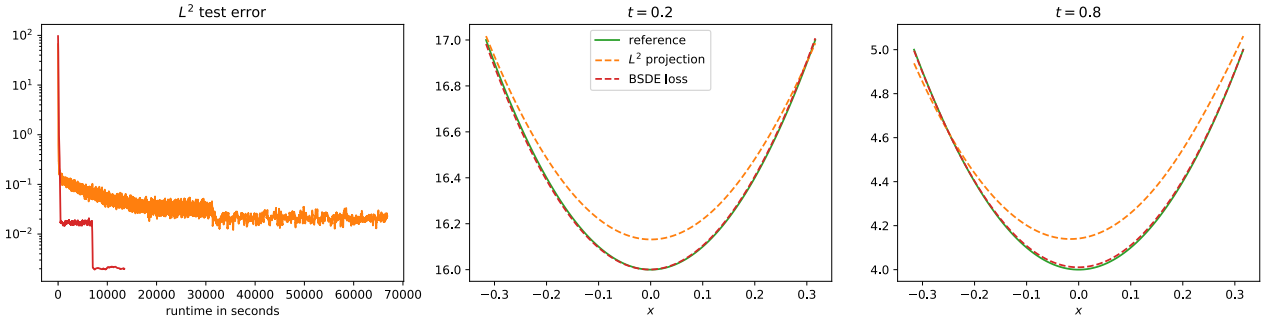


Figure 6.19: Comparison of the BSDE loss and the L^2 projection attempt for approximating the solution of a linear heat equation. Left: Smoothed L^2 test error computed on uniformly sampled data points along the training iterations. Middle and right panels: Approximations compared to the reference solution for two different points in time.

6.3.7.10 Computing expected exit times

In the remaining numerical examples we shall illustrate how sampling problems similar to the ones introduced in Chapter 1 can be approached by approximating solutions to PDEs. We will employ the variational method from Section 6.3 and mostly rely on the PINN loss, noting however that the diffusion loss would be equally valid. For the BSDE loss we note once more that in particular large hitting times might cause numerical challenges.

Let us start by computing expected exit times $\mathbb{E}[\tau]$, where $\tau = \inf\{t > 0 : X_t \notin \mathcal{D}\}$ is the first exit from the domain $\mathcal{D} \subset \mathbb{R}^d$. This expectation value can be approximated by Monte Carlo estimation, which however is difficult if for instance the hitting times are large or the estimators exhibit high variances. Taking the PDE perspective, we note that with Theorem 2.14 and Remark 2.17 the function $V(x) = \mathbb{E}[\tau | X_0 = x]$ fulfills the elliptic boundary value problem

$$LV(x) + 1 = 0, \qquad x \in \mathcal{D}, \tag{6.134a}$$

$$V(x) = 0, \qquad x \in \partial\mathcal{D}. \tag{6.134b}$$

For an example, let us consider the one-dimensional Langevin dynamics

$$\mathrm{d}X_s = -\nabla\Psi(X_s)\mathrm{d}s + \sqrt{\eta}\,\mathrm{d}W_s, \qquad X_0 = x_{\text{init}}, \tag{6.135}$$

with a potential $\Psi(x) = \kappa(x^2 - 1)^2$ and recall that $\mathbb{E}[\tau] \simeq \exp(2\Delta\Psi/\eta)$ as $\eta \to 0$, implying that for large values of κ and small values of η the hitting times become large and naive Monte Carlo sampling gets more and more challenging, cf. e.g. Example 3.13. We choose $\mathcal{D} = (-\infty, 1], \kappa = 10, \eta = 4$ and aim to approximate the solution to (6.134) with the PINN loss. In Figure 6.20 we display the L^2 error compared to a reference solution computed with finite differences. In the right panel we compare the PDE approximation to the reference solution as well as to sampled expected hitting times based on $K = 1000$ trajectories with different step-sizes in the Euler discretization. We can see that the PDE solution does not suffer from variance issues that much, indicating that, even though it brings additional challenges, approximating the PDE might be constructive for solving the estimation problem. We further note that for the Monte Carlo approximation small step-sizes are necessary, whereas the PDE approximation with the PINN loss does not rely on a time discretization.

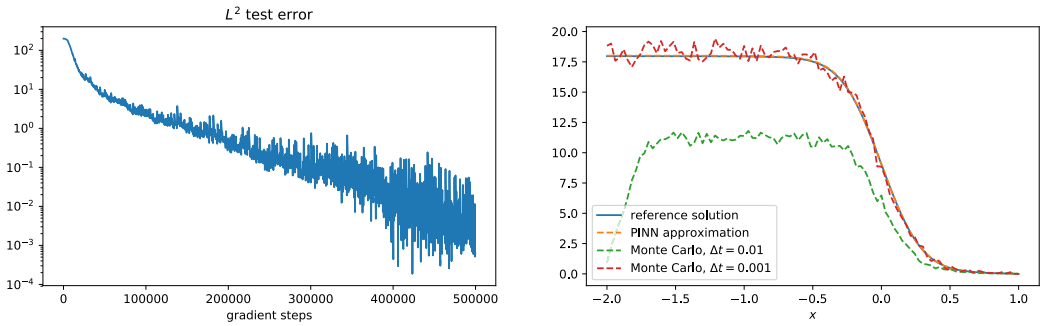


Figure 6.20: Computation of the expected exit time by solving a linear PDE. Left: L^2 error on test data along the training iterations. Right: Reference PDE solution compared with its approximation by the PINN method as well as by Monte Carlo approximation using $K = 1000$ samples.

6.3.7.11 Exit times in a multidimensional double well potential

Let us further aim to compute the quantity $\mathbb{E}[e^{-\tau}]$, which amounts to choosing $f = 1, g = 0$ in (1.8), again with $\tau = \inf\{t > 0 : X_t \notin \mathcal{D}\}$ being the first exit time, where we now choose the domain to be $\mathcal{D} = \{x \in \mathbb{R}^d : x_i < 1\}$. Let us consider the Langevin dynamics as in (6.135), now in $d = 2$, where we choose the double well potential $V(x) = \sum_{i=1}^{d}(x_i^2 - 1)^2$. According to Theorem 2.14 the expectation $V(x) = \mathbb{E}\left[e^{-\tau}|X_0 = x\right]$ fulfills the PDE

$$(\partial_t + L - 1)V(x) = 0, \qquad x \in \mathcal{D}, \tag{6.136a}$$

$$V(x) = 1, \qquad x \in \partial\mathcal{D}. \tag{6.136b}$$

In the left panel of Figure 6.21 we display a neural network approximation that we have gained using the PINN loss and in the left we show a reference solution computed with finite elements. The good agreement of our approximation can be confirmed when looking at the functions evaluated along the curve $\left\{(x, x)^\top : x \in [-2, 1]\right\}$, as illustrated in the right panel.

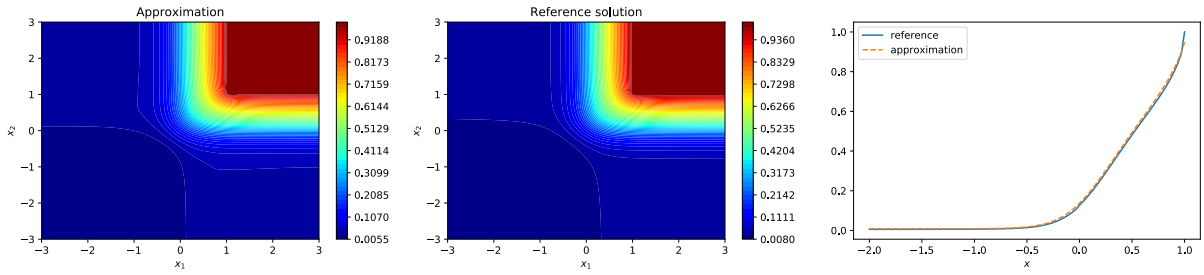


Figure 6.21: Approximation and reference solution for the PDE related to exiting a $2d$ double well potential via the top right well.

6.3.7.12 Leaving metastable sets before given time

Next, let us repeat an example from Section 6.2.3.6 and consider computing the probability of leaving a metastable set before a prescribed time T in $d = 1$, namely

$$V(x, t) = P(\tau < T|X_t = x), \tag{6.137}$$

for the dynamics given by the Langevin equation as specified in (6.135). We take $d = 1$, the potential $\Psi(x) = (x^2 - 1)^2$, the noise coefficient $\sqrt{\eta} = 0.5$ and a random stopping time $\tau = \inf\{t > 0 : X_t \notin \mathcal{D}\}$ for leaving the interval $\mathcal{D} = (\infty, 0)$. One approach is to consider the transformation

$$\widetilde{V} = -\log V, \tag{6.138}$$

which brings a HJB equation with terminal costs $g(x) = -\log(\mathbb{1}_{\partial\mathcal{D}}(x))$. In Section 6.2.3.6 we have taken this path, introducing a regularization in order to cope with the singularity at the boundary. Alternatively, we know that V fulfills the parabolic boundary value problem

$$(\partial_t + L)V(x, t) = 0, \qquad (x, t) \in \mathcal{D} \times [0, T), \tag{6.139a}$$

$$V(0, t) = 1, \qquad t \in [0, T), \tag{6.139b}$$

$$V(x, T) = 0, \qquad x \in \mathcal{D}, \tag{6.139c}$$

which can be solved directly by the methods introduced in Section 6.3. We approach this problem with the PINN loss and compare the approximated solution with a reference solution computed by numerical discretization of the PDE in Figure 6.22. We see good agreement for most of the domain, but note that whenever the committor probability is close to zero, which relates to rare event probabilities, we might still encounter a large relative error $\left|\frac{V(x)-\varphi(x)}{V(x)}\right|$, as displayed in the right panel of Figure 6.22. We conclude that for computing rare event probabilities it seems to be not sufficient to solely rely on the PDE approximation. As demonstrated in Section 6.2.3.6, however, one can still improve rare event estimates by using importance sampling schemes incorporating controls based on the PDE solution.

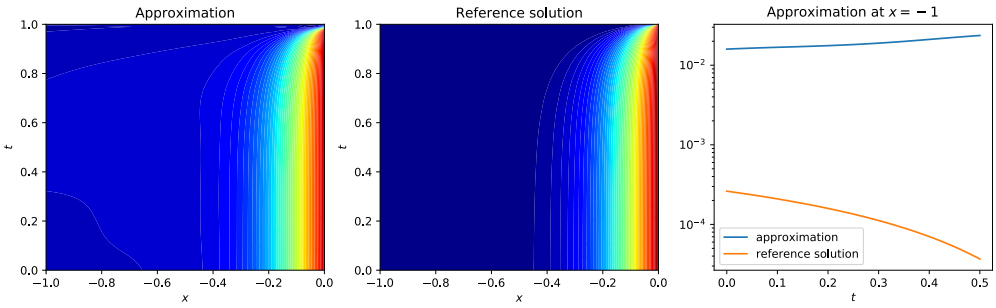


Figure 6.22: Left and middle: Approximation and reference solution of the committor probability as a function of the starting point x and the starting time t. Right: Comparison of approximation and reference solution for small values of the committor function, which relate to rare event probabilities.

Chapter 7

Conclusion and outlook

Motivated by the estimation of functionals related to diffusion processes, this thesis studied the robust computation of expectation values and proposed advanced algorithms for solving high-dimensional semi-linear PDEs. Let us take the chance to conclude this endeavor and provide an outlook towards future research questions. Along these lines, we will state proofs of concepts for novel algorithms related to the Schrödinger problem and the learning of zero-variance importance sampling proposal densities.

Suboptimal importance sampling

In order to explain potential non-robustness issues of importance sampling, Chapter 3 was devoted to quantitative bounds on the relative error of the corresponding estimator that depend on the divergence between the actual proposal measure and the theoretically optimal one. These bounds indicate that importance sampling is very sensitive with respect to suboptimal choices of the proposals, which has been observed frequently in numerical experiments before and is in line with recent theoretical analysis [3, 51, 260]. In particular, we showed that the relative error scales exponentially in the KL divergence between the optimal and the actual proposal measure and argued that this renders importance sampling especially challenging in high dimensional settings.

We have focused on importance sampling of stochastic processes and derived some novel formulas for the relative error depending on the suboptimality of the function u that controls the drift of the process. These formulas can be used to get practically useful bounds and they also indicate two potential issues for importance sampling in path space: for high-dimensional systems and for problems on a long time horizon the relative error becomes exponentially large in the state space dimension d and the time horizon T. We have briefly discussed how this observation can be transferred to random stopping times, such as first hitting times, and have applied our formulas to importance sampling in a small noise regime, offering new perspectives and revealing some potential drawbacks of existing methods.

Even though the key message regarding the use of importance sampling in high dimensions seems to be rather discouraging, the following chapter provided a remedy at least for numerical methods that approach optimal importance sampling schemes.

Robust variational minimization via the log-variance divergence

Motivated by taking different perspectives on the sampling problem, as presented in Problems 1.1-1.5, some of which bearing variational formulations, Chapter 4 provided a unifying framework based on divergences between path space measures, encompassing various existing numerical methods in the class of algorithms that we termed *iterative diffusion optimization*. In particular, we have introduced the novel log-variance divergence and showed its close connection to forward-backward SDEs. We have furthermore shown a fundamental equivalence between approaches based on the KL-divergence and the log-variance divergence. Turning to the variance of Monte Carlo gradient estimators, we have defined and studied two notions of stability – robustness under tensorization and robustness at the solution. Of the losses and estimators under consideration, only the log-variance loss is stable in both senses, which often results in superior numerical performance. The consequences of robustness and non-robustness have been exemplified by extensive numerical experiments.

The results presented in Chapter 4 can be extended in various directions. First, it would be interesting to consider other divergences on path space and construct and study the ensuing algorithms. Deeper understanding of the design of IDO algorithms could be achieved by extending our stability analysis beyond the product case and for controls that differ greatly from the optimal one. In particular, advances in this direction might help

to develop more sophisticated variance reduction techniques. Furthermore, it would be interesting to design algorithms that can treat more general HJB equations and address optimal control problems involving additional minimization tasks posed by general Hamiltonians, for which it might be feasible to include appropriate penalty terms in the loss functional.

Extensions to more general PDEs

We may attempt to generalize the framework and numerical approaches from Chapter 4 to more general PDEs. We have argued in Remark 4.14 that an application of the log-variance loss to any PDE of the form (1.33) is straightforward as long as the nonlinearity only depends on the solution through its gradient. Likewise, an application to elliptic PDEs and boundary value problems follows readily (see also Chapter 6). More challenging will be an application of the path space considerations and the log-variance loss to more general semi-linear PDEs where the nonlinearity may also depend on the solution directly. As already discussed in Remark 4.14 one idea is to add additional appropriate boundary terms that mitigate the shift-invariance of this loss. Another attempt is to combine the log-variance with the moment loss. To this end, similar to (4.29), we can consider

$$Y_T^{\widetilde{u},v} = y_0 + \widetilde{Y}_T^{\widetilde{u},v}, \qquad \widetilde{Y}_T^{\widetilde{u},v} = -\int_0^T h(X_s^v, s, Y_s^v, -\widetilde{u}(X_s^v, s))\,\mathrm{d}s - \int_0^T (v \cdot \widetilde{u})(X_s^v, s)\,\mathrm{d}s - \int_0^T \widetilde{u}(X_s^v, s) \cdot \mathrm{d}W_s. \quad (7.1)$$

An idea is to split the moment loss into two parts, namely

$$\begin{aligned}
\mathcal{L}_{\mathrm{moment}_v}(u, y_0) &= \mathrm{Var}\left(Y_T^{u,v}(y_0) - g(X_T^v)\right) + \mathbb{E}\left[Y_T^{u,v}(y_0) - g(X_T^v)\right]^2 && (7.2a)\\
&= \mathrm{Var}\left(\widetilde{Y}_T^{u,v}(y_0) - g(X_T^v)\right) + \mathbb{E}\left[Y_T^{u,v}(y_0) - g(X_T^v)\right]^2 && (7.2b)\\
&=: \mathcal{L}^{\log}_{\mathrm{Var}_v}(u, y_0) + \mathcal{L}^2_{\mathrm{mean}_v}(u, y_0). && (7.2c)
\end{aligned}$$

Even though both terms depend on u and y_0, one can aim at only differentiating the first term w.r.t. u and the second term w.r.t. y_0, hoping for reduced variances of corresponding gradient estimators. We leave it to future research whether such a method or similar attempts bring any computational advantages.

Efficient forward controls

Recalling that many of the losses that we have considered in Chapter 4 are valid for arbitrary forward controls v, a notorious question is which choice of v to make in practice. It might be desirable to choose controls that improve statistical properties of the gradient estimator or make the exploration of the state space more efficient. We have argued in Section 4.2 that in the case of the nonlinearity $h(x,t,y,z) = -\frac{1}{2}|z|^2$ (relating to Problems 1.1-1.5) the choice $v = u$ seems beneficial, see in particular Proposition 4.19 and Remark 4.23. For the general case, similar to the computation in (C.40), a variation of the moment loss brings

$$\begin{aligned}
\frac{\delta}{\delta u}\mathcal{L}_{\mathrm{moment}_v}(u, y_0) = 2\,\mathbb{E}\Bigg[\left(Y_T^{u,v}(y_0) - g(X_T^v)\right) \Bigg(&\int_0^T \nabla_z h(X_s^v, s, Y_s^v, -u_s) \cdot \phi(X_s^v, s)\,\mathrm{d}s \\
&- \int_0^T (v \cdot \phi)(X_s^v, s)\,\mathrm{d}s - \int_0^T \phi(X_s^v, s)\ \cdot \mathrm{d}W_s \Bigg)\Bigg],
\end{aligned} \quad (7.3)$$

which might suggest to use $v = \nabla_z h$ in order to cancel two of the integrals in (7.3). We expect that further theoretical and numerical research in this direction might be fruitful in particular for more general PDEs.

Multiple scales

Even though the log-variance loss has favorable scaling properties when going to high dimensions, the performance of the variational approach generally degrades when the dimension of the underlying dynamics is high or when the equations are very stiff (e.g. due to the presence of multiple time or length scales). A strategy to cope with high dimensionality or stiffness is to reduce the dimension of the dynamics prior to solving the variational problem by eliminating (e.g. fast or stiff) degrees of freedom.

Assuming that the reduced system captures all features of the quantity of interest (here: f or g) for any given control (or no control), an obvious question then is whether the reduced dynamics can also be used to compute an approximation to the solution of the full dynamics. In terms of the associated path space measure $\mathbb{P}^u$ and $\mathbb{P}^{u^*} = \mathbb{Q}$, the question is whether minimizers stay minimizers if $\mathbb{P}^u$ and $\mathbb{Q}$ are replaced by suitable approximations, and whether both the minimizers and the corresponding functionals are (in some sense) close to each other.

Conditioning of stochastic processes and the Schrödinger problem

In Chapter 1 we have stated five problems that are more or loss equivalent to each other. We can in fact add yet another perspective which will turn out to be highly related, originally going back to Erwin Schrödinger [263]. It considers controlled stochastic processes that are conditioned on starting from an initial density and ending at a prescribed target density, while minimizing a given control cost.

Problem 7.1 (Schrödinger)**.** *Given two densities μ_0 and μ_T and the controlled diffusion as in* (1.4)*,*

$$\mathrm{d}X_s^u = (b(X_s^u, s) + \sigma(X_s^u, s)u(X_s^u, s))\,\mathrm{d}s + \sigma(X_s^u, s)\,\mathrm{d}W_s, \qquad X_0^u \sim \mu_0, \tag{7.4}$$

find a control u^* such that

$$X_T^{u^*} \sim \mu_T \tag{7.5}$$

and such that u^* has minimal control energy, i.e.

$$J(u) = \mathbb{E}\left[\frac{1}{2}\int_0^T |u(X_s^u, s)|^2 \mathrm{d}s\right] \tag{7.6}$$

is minimized.

We refer to Appendix B.10 for a discussion on how Problem 7.1 is connected to Problems 1.1-1.5 and how one can attempt to solve it. Combining ideas from stochastic optimal control and iterative diffusion optimizations from Chapter 4, we can propose a numerical algorithm for the case of μ_0 being a Dirac measure. In particular, this algorithm enables us to sample from any arbitrary prescribed target density, similar to an approach in [284], however relying on the log-variance loss as illustrated in the following example.

Example 7.1 (Sampling from prescribed density)**.** *Let us consider a one-dimensional toy example that shall demonstrate how we can use Theorem B.16 to sample from some prescribed density. Let us say we want to sample from the Gaussian mixture model*[43]

$$\mu_T(x) = \frac{1}{2}\mathcal{N}(x; -2, 0.1) + \frac{1}{2}\mathcal{N}(x; 2, 0.1). \tag{7.7}$$

We can define our data generating process X as we like, as long as we know its distribution at terminal time $T = 1$. Let us consider an Ornstein-Uhlenbeck dynamics

$$\mathrm{d}X_s = -X_s\,\mathrm{d}s + \sqrt{2}\,\mathrm{d}W_s, \qquad X_0 = 0, \tag{7.8}$$

whose density at time $t \in [0, T]$ is given by

$$p_t(x) = \mathcal{N}(x; 0, 1 - e^{-2t}), \tag{7.9}$$

as specified in (B.29)*. As argued in Remark B.17, finding the drift that lets the process end up at the prescribed density μ_T in some optimal way is equivalent to solving an optimal control problem. Since the structure of this problem is the same compared to the problems we have considered in Chapter 4 we shall rely on the log-variance loss to find appropriate approximations of the optimal control. We parametrize the control u with a neural network and recall from Proposition 4.12 that the log-variance loss for this problem is given by*

$$\mathcal{L}^{\log}_{\mathrm{Var}_v}(u) = \mathrm{Var}\left(\widetilde{Y}_T^{u,v} + \log\frac{\mu_T}{p_T}(X_T^v)\right) = \mathrm{Var}\left(\widetilde{Y}_T^{u,v} + \log\frac{\widetilde{\mu}_T}{\widetilde{p}_T}(X_T^v)\right), \tag{7.10}$$

where we added the additional forward control v, with X^v defined as in (7.4) *with u replaced by v, and refer to* (4.19) *for a definition of $\widetilde{Y}_T^{u,v}$. In our simulation we choose for v the current approximation of u^* with the purpose to move our trajectories to the target already in the training runs, aiming for higher numerical stability when computing the likelihood ratio. With $\widetilde{\mu}_T \propto \mu_T, \widetilde{p}_T \propto p_T$ we denote the corresponding unnormalized densities and realize that the log-variance loss offers the convenient property that the normalization constants can be omitted (this can also be compared to Section 5.2). In Figure 7.1 we display the learnt control in the left panel and some controlled trajectories in the center. The right panel demonstrates that indeed the target density μ_T is reached when applying the approximated optimal control.*

[43] Of course, sampling from Gaussians can be done much more efficiently and we choose this toy density merely for demonstrational purposes, noting that it consists of two modes. We stress that our algorithm is black box in nature and can be applied to any density that can be written down even without knowing its normalizing constant.

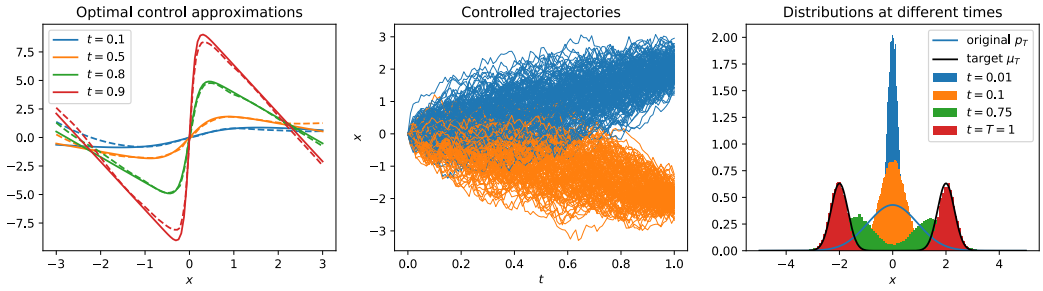


Figure 7.1: Left panel: Optimal control computed by the neural network approximation (dashed lines) compared to a reference solution computed by numerical integration (solid lines) for different times. Center: Half of the controlled sample trajectories end up in either of the two modes of the target density. Right: We display the histograms of controlled trajectories at certain times and see that they indeed agree with the target density at time $t = T$.

Example 7.1 serves as a proof of concept and future research might study how these considerations can be applied to more challenging high-dimensional sampling problems.

Extensions to data assimilation

Another aspect is an extension to data assimilation in a Bayesian framework [246]. Let us consider a situation in which the initial condition in (1.2) is random rather than deterministic, i.e. $X_0 \sim \rho_0$, for a given probability distribution ρ_0. In the context of data assimilation, ρ_0 may be interpreted as encoding prior knowledge or belief about the system under consideration at time $t = 0$. The uncontrolled SDE (1.2) now induces a probability measure $\mathbb{P}$ on $\mathcal{C}$, with time marginal at $t = 0$ given by ρ_0. Assuming a Bayesian perspective, $\mathbb{P}$ represents our a priori belief, that is, our belief in the absence of data, concerning the probability that a certain path is realized by the system described by (1.2).

For the remaining discussion, we will adopt a notation that is common in the statistics literature and denote $\mathbb{P}$ by $p((X_t)_{0\le t\le T})$. Suppose that a noisy measurement y_{obs} of X_t becomes available at time $t = T$. A typical observation model might for instance be given by

$$y_{\text{obs}} = X_T + \xi, \tag{7.11}$$

where ξ is a mean-zero Gaussian random variable with covariance matrix $\Sigma \in \mathbb{R}^{d\times d}$. In other words, the *likelihood* of observing y_{obs} given a (deterministic) path $(X_s)_{0\le s\le T} \in \mathcal{C}$ can be written as

$$p(y_{\text{obs}}|(X_s)_{0\le s\le T}) = p(y_{\text{obs}}|X_T) \propto \exp\left(-\frac{1}{2}(X_T - y_{\text{obs}})\Sigma^{-1}(X_T - y_{\text{obs}})\right). \tag{7.12}$$

In the current setting the likelihood depends on $(X_s)_{0\le s\le T}$ only through X_T; we made this fact explicit by slight abuse of notation. (More general measurement processes can be modelled by alternative specifications of $p(y_{\text{obs}}|(X_t)_{0\le t\le T})$.) The Bayesian approach to statistics dictates that a posteriori belief that takes into account the measurement is encoded by the posterior probability

$$p((X_t)_{0\le t\le T}|y_{\text{obs}}) \propto p((X_s)_{0\le t\le T})p(y_{\text{obs}}|(X_t)_{0\le t\le T}). \tag{7.13}$$

Direct simulation from (7.13) is possible by reweighting samples from $(X_s)_{0\le s\le T}$ according to $p(y_{\text{obs}}|(X_s)_{0\le s\le T})$. In high dimensions, however, or when $p((X_s)_{0\le t\le T})$ and $p(y_{\text{obs}}|(X_t)_{0\le t\le T})$ barely overlap, this approach becomes computationally infeasible as the effective sample size decreases sharply (cf. also Chapter 3). Note that the challenges posed by rare event sampling in molecular dynamics and data assimilation are completely analogous. Not surprisingly, posterior approximation can be addressed solving the Problems 1.1-1.5.

Proposition 7.2. *[134, 246] For fixed y_{obs}, assume that $p(y_{obs}|X_T) > 0$ and let g be the negative log-likelihood*

$$g(x) = -\log p(y_{obs}|X_T = x), \qquad x \in \mathbb{R}^d. \tag{7.14}$$

Set $f = 0$ and let V be the value function solving (1.20). Set $u^* = -\sigma^\top \nabla V$ and define the reweighted initial distribution

$$\widehat{\rho}_0(x) = \frac{\rho_0(x)e^{-V(x,0)}}{\int_{\mathbb{R}^d} \rho_0(x')e^{-V(x',0)}\,\mathrm{d}x'}. \tag{7.15}$$

Then the path measure induced by the controlled SDE (4.4) with initial condition $X_0^{u^*} \sim \widehat{\rho}_0$ coincides with (7.13).

The reweighted initial condition $\widehat{\rho}_0$ represents the updated belief about the distribution at $t = 0$ with data incorporated at $t = T$. Proposition 7.2 explains how to efficiently sample from the Bayesian posterior (7.13) based on the solution to any of the Problems 1.1-1.5. Conversely, it may open up the possibility to develop new sequential Monte Carlo or particle filtering algorithms for solving large-scale HJB equations.

The log-variance divergence for the approximation of densities

We have applied the log-variance divergence to densities in the context of variational approximations of Bayesian posteriors in Chapter 5. This resulted in an efficient way of computing a low-variance gradient estimator, termed *VarGrad*, which can be seen as a control variate version of its naive counterpart. We have demonstrated that this estimator brings computational advantages in real-world numerical examples and showed theoretically that under certain conditions it is close to the optimal control variate scaling. Moreover, we have established conditions under which VarGrad is guaranteed to exhibit lower variance than the naive estimator. For future work it might be interesting to explore the direct optimization of the log-variance loss for alternative choices of the reference distribution r. It might also be worthwhile to combine the log-variance loss with the reparametrization trick for gradient computations [168, 299], aiming for further variance reductions.

Of course the approximation of probability densities is relevant in many more fields and we expect that relying on the log-variance loss in further applications might be fruitful, in particular keeping in mind the convenient property that no knowledge of normalizing constants is required.

Learning optimal importance sampling proposal densities

One application for approximating densities is related to importance sampling in $\mathbb{R}^d$. We can aim to learn proposal densities that significantly reduce the variance of Monte Carlo estimators, similar to attempts in [213] and [227, Section 6.2.1]. Let us say we want to compute the expectation

$$\mathcal{Z} = \mathbb{E}\left[e^{-g(X)}\right], \tag{7.16}$$

where the random variable $X \in \mathbb{R}^d$ is distributed according to some density p and $g \in C(\mathbb{R}^d, \mathbb{R})$ is a given function. We recall the idea of importance sampling, e.g. from Section 2.3.1, namely sampling from an alternative probability density and reweighting to get

$$\mathcal{Z} = \mathbb{E}\left[e^{-g(\widetilde{X})}\frac{p}{\widetilde{p}}(\widetilde{X})\right], \tag{7.17}$$

where $\widetilde{X} \sim \widetilde{p}$, and we note that an optimal, zero-variance importance sampling proposal density $\widetilde{p}$ is given by

$$q(x) = \frac{e^{-g(x)}}{\mathcal{Z}}p(x). \tag{7.18}$$

Hence a notorious goal in importance sampling is to aim at approximations $\widetilde{p} \approx q$, which can be approached by minimizing divergences between $\widetilde{p}$ and q, leading to implementable losses, similar to what we have considered in Chapter 4 for path space measures and in Chapter 5 in the context of Bayesian variational inference. We shall focus on the log-variance divergence

$$D_r^{\mathrm{Var(log)}}(\widetilde{p}|q) = \mathrm{Var}_r\left(\log\frac{q}{\widetilde{p}}(X)\right), \tag{7.19}$$

where r specifies an arbitrary reference density, leading to the log-variance loss

$$\mathcal{L}^{\log}_{\mathrm{Var}_r}(\widetilde{p}) = \mathrm{Var}_r\left(\log p(X) - \log\widetilde{p}(X) - g(X)\right), \tag{7.20}$$

which admits the convenient property that it is independent of $\mathcal{Z}$ (see Definition 4.4). In all of the following demonstrations we will choose the reference distribution $r = \widetilde{p}$ to be the current approximation of the target density, however, in analogy to Chapters 4 and 5, we will not differentiate with respect to this term. One idea could be to parametrize $\widetilde{p} = \widetilde{p}_\theta$ with some parameter $\theta \in \mathbb{R}^p$ and iteratively minimize the above loss with gradient descent methods in θ. In practice, the optimal proposal q as defined in (7.18) is unknown and it makes sense to consider a class of flexible proposal densities $\widetilde{p}$, in particular keeping in mind that our importance sampling ambitions require that it is both easy to sample from as well as easy to evaluate $\widetilde{p}$. We will therefore rely on the concept of so-called *normalizing flows* [175], where the idea is to not learn the function $\widetilde{p}$ directly, but rather to learn a deterministic transformation of a random variable in such a way that the transformed variable is distributed according to $\widetilde{p}$. To be precise, we take any base density p_Z, draw a random variable

$Z \sim p_Z$ from this density and consider the transformation $X = \varphi(Z) \sim \widetilde{p}$, where $\varphi : \mathbb{R}^d \to \mathbb{R}^d$ is a function to be specified. We recall that

$$\widetilde{p}(x) = p_Z(\varphi^{-1}(x))|\det D\varphi^{-1}(x)|, \tag{7.21}$$

where $D\varphi^{-1}$ is the Jacobi matrix of φ^{-1}. For computing $\widetilde{p}$ as in (7.21) we identify two main challenges: the function φ needs to be invertible and computing its Jacobi determinant should be not too complicated. In recent years special types of neural networks have been designed that fulfill these two properties, while providing the usual flexible function approximation qualities [175]. We will rely on so-called coupling flows [71], in particular on an attempt suggested in [72]. Let us emphasize that the base density p_Z is arbitrary and that sampling from $\widetilde{p}$ is straightforward as long as it is easy to sample from p_Z. The following example illustrates this idea as a poof of concept for learning optimal proposal densities and we refer to Appendix B.11 for further examples.

Example 7.3 (Non-Gaussian target, high-dimensional)**.** *Let p be the standard Gaussian and consider $g(x) = x_1^4 + \cdots + x_d^4$ in dimension $d = 10$ such that the optimal proposal q is not Gaussian. We consider the normalizing flow attempt as described above and use the log-variance loss* (7.20) *as our objective to minimize. In Figure 7.2 we display the learning progress and the approximation of q along the curve $\{(x, \dots, x)^\top : x \in [-1, 1]\}$, where we see good agreement with the target. Using this approximation as an importance sampling proposal we can bring the relative error of the estimator from roughly 5 in the naive Monte Carlo attempt down to 10^{-1}.*

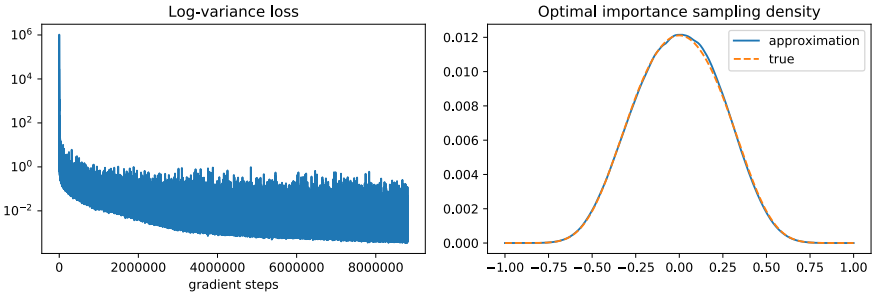


Figure 7.2: Left: Log-variance loss along the training iterations. Right: Approximation of the optimal (non-Gaussian) proposal density.

Computing partition functions

Along the lines of computing optimal importance sampling densities we can further aim to approximate partition functions with normalizing flows and the log-variance loss. This is illustrated in the following example.

Example 7.4 (Computing partition functions)**.** *It is often of interest to compute the partition function*

$$\mathcal{A} = \int_{\mathbb{R}^d} e^{-\beta\Psi(x)} \mathrm{d}x, \tag{7.22}$$

where $\Psi : \mathbb{R}^d \to \mathbb{R}$ is a given potential and $\beta > 0$ a temperature, yielding the Gibbs-Boltzmann density

$$p(x) = \frac{e^{-\beta\Psi(x)}}{\mathcal{A}}. \tag{7.23}$$

We recall the Donsker-Varadhan variational formula from Theorem 2.44,

$$-\log \int e^{-g(x)} p(x) \mathrm{d}x = \inf_{\widetilde{p} \in \mathcal{P}(\mathbb{R}^d)} \left\{ \int g(x)\widetilde{p}(x)\mathrm{d}x + \mathrm{KL}(\widetilde{p}|p) \right\}. \tag{7.24}$$

Taking $g = 0$ and p as in (7.23) *brings*

$$0 = \inf_{\widetilde{p} \in \mathcal{P}(\mathbb{R}^d)} \mathbb{E}_{\widetilde{p}} \left[\log \widetilde{p}(X) + \beta\Psi(X)\right] + \log \mathcal{A} \tag{7.25}$$

and with the optimal $\widetilde{p} = q$ we get

$$\mathcal{A} = e^{-\mathbb{E}_q[\log q(X) + \beta\Psi(X)]}, \tag{7.26}$$

noting that in fact $q = p$. Instead of minimizing (7.25) *directly, we again rely on the log-variance loss as our objective, which we can now write as*

$$\mathcal{L}^{\log}_{\mathrm{Var}_r}(\widetilde{p}) = \mathrm{Var}_r\left(\log \widetilde{p}(X) + \beta\Psi(X)\right), \tag{7.27}$$

where we should once again appreciate the fact that the constant $\mathcal{A}$ vanishes[44]*. Let us first consider a convex potential $\Psi(x) = |x|^2$ and set $\beta = 1$, then p is a multidimensional standard Gaussian and we know that $\mathcal{A} = \sqrt{\pi^d}$. We choose $d = 20$ and display the learning progress in Figure 7.3. We can see that we can approximate the partition function quite well with a relative error $\frac{|\mathcal{A}-\widehat{\mathcal{A}}|}{\mathcal{A}}$ of roughly 10^{-4}.*

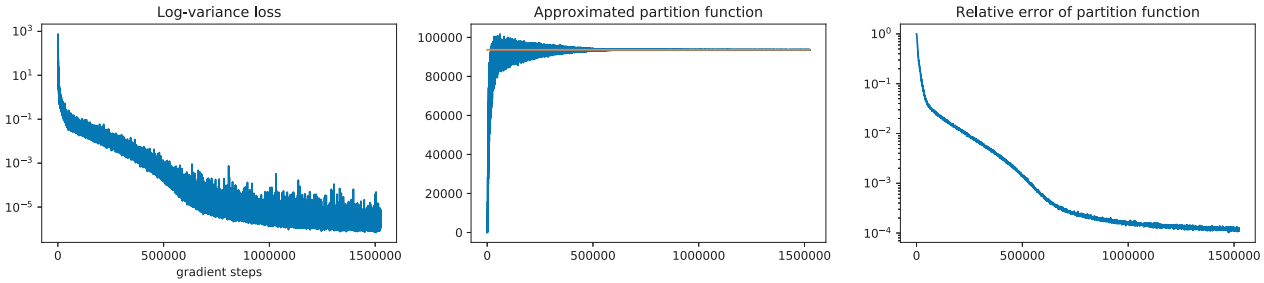


Figure 7.3: Left: Log-variance loss over gradient steps for learning the partition function $\mathcal{A}$. Middle: We compare the current approximation of $\mathcal{A}$ according to (7.26) with to the true value indicated by the orange line. Right: We plot a moving average of the relative error with a smoothing window of 1000.

Next, let us consider the d-dimensional double well potential

$$V(x) = \left(b - \frac{a}{2}\right) \sum_{i=1}^{d} (x_i^2 - 1)^2 + \frac{a}{2} \sum_{i=1}^{d} (x_i - 1), \tag{7.30}$$

where we choose $a = 0.1$ and $b = 1$. Since the dimensions do not interact we can compute a reference value for $\mathcal{A}$ by numerical integration. In Figure 7.4 we display the learning progress for the choice of $d = 4, \beta = 0.2$. Due to the non-convexity of the potential the learning becomes harder, but we can still reach a relative error of roughly 10^{-2} for the approximation of $\mathcal{A}$.

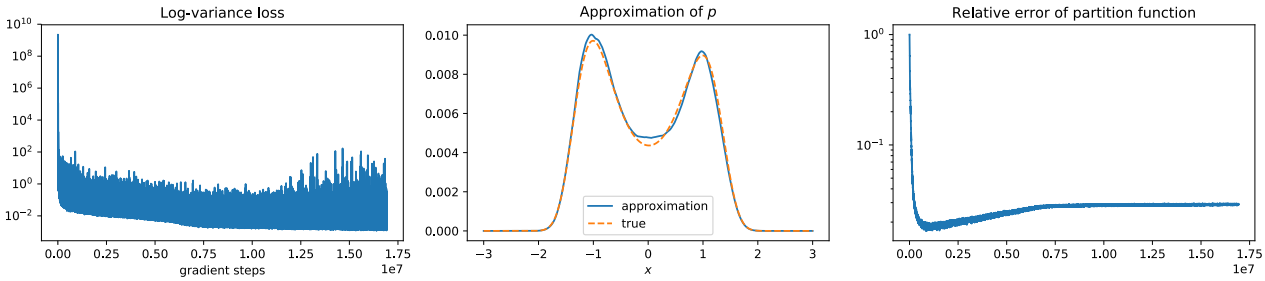


Figure 7.4: Left: Log-variance loss over the gradient steps for learning the partition function $\mathcal{A}$ of the double well example. Middle: We compare the approximation $\widetilde{p}$ of $q = p$ with the reference solution along the curve $\{(x, \dots, x)^\top : x \in [-1, 1]\}$. Right: We plot a moving average of the relative error with a smoothing window of 1000.

For the approximation of partition functions as well as for learning optimal importance sampling densities future research might study how to approach high-dimensional and more metastable examples in a robust way.

Backward iterations using the tensor train format

We have addressed and improved algorithms for solving parabolic PDEs on unbounded domains in Section 6.2. In particular, we have proposed to rely on the tensor train format for function approximation in backward iteration schemes, potentially leveraging their efficient approximation capabilities whenever low-rank structures are present. We have considered both explicit and implicit schemes, allowing for a trade-off between approximation

[44]Note that minimizing the KL divergence based losses, on the other hand, brings additional challenges. When considering

$$\mathrm{KL}(p|\widetilde{p}) = \mathbb{E}_p\left[-\Psi(X) - \widetilde{p}(X)\right] - \log \mathcal{A} \tag{7.28}$$

we need to be able to sample from p, which is usually non-trivial, and when considering

$$\mathrm{KL}(\widetilde{p}|p) = \mathbb{E}_{\widetilde{p}}\left[\Psi(X) + \widetilde{p}(X)\right] + \log \mathcal{A} \tag{7.29}$$

the random variables are distributed according to $\widetilde{p}$ w.r.t. which we need to differentiate, which often leads to numerical challenges.

accuracy and computational cost. Notably, the tensor train format specifically allows us to take advantage of the additional structure inherent in least-squares based formulations, particularly in the explicit case. This led to substantial computational advantages compared to state-of-the-art neural network based approaches in various numerical experiments. In this context it will be particularly interesting to better understand structures of PDE solutions, in particular in high dimensions. We believe that the "blessing of dimensionality" observed in Section 6.2.3.1 deserves a mathematically rigorous explanation; progress in this direction may further inform the design of scalable schemes for high-dimensional PDEs. It is further appealing to combine backward iterations with the log-variance loss, which is feasible whenever the nonlinearity in the PDE only depends on the solution through its gradient (cf. Remark 6.7). Finally, an analysis and systematic numerical study on how backward iteration algorithms can be applied to bounded domains is an interesting topic for future research.

Variational formulations of elliptic and parabolic boundary value problems

We have reviewed the approximation of semi-linear elliptic and parabolic boundary value problems from the perspective of variational formulations, incorporating and generalizing residual and BSDE based methods in Section 6.3. The novel *diffusion loss*, which combines ideas from both methods, brought some substantial advantages in certain numerical experiments, in particular targeting issues of the BSDE based method in large domains as well as close to the boundary. Here it will be interesting to study more systematic choices of the weights appearing in the loss terms as well as to consider sampling strategies other than the uniform distribution in order to emphasize certain regions of interest or to speed up convergence. For the PINN method it will be worthwhile to incorporate variance reduction ideas similar to the ones that we have applied in the BSDE case.

We have further extended our algorithms to the approximation of principal eigenpairs to linear and nonlinear eigenvalue problems. While having provided a proof for the linear case, a rigorous treatment for the nonlinear setting is still open. We expect that the approximation of eigenfunctions and eigenvalues other than the principal one might be tackled by integrating appropriate additional terms into the loss functions.

Appendix A

Notation

$\mathbb{N}$	set of natural numbers
$\mathbb{Z}$	set of integer numbers
$\mathbb{R}$	set of real numbers
$x^\top$	transpose of the variable $x \in \mathbb{R}^d$
$x \cdot y$	scalar product of the vectors $x, y \in \mathbb{R}^d$, defined as $x^\top y$
$\lvert x \rvert$	Euclidean norm of the vector $x \in \mathbb{R}^d$, defined as $\sqrt{x \cdot x}$
$x \odot y$	elementwise multiplication of the vectors $x, y \in \mathbb{R}^d$
$\bigotimes$	tensor product
Tr	trace operator
$A : B$	$\mathrm{Tr}(A^\top B)$, where A and B are matrices
$X \sim \nu$	random variable X is distributed according to probability measure ν
$A \simeq B$	A is asymptotically equivalent to B
$f \in o(g)$	f is asymptotically negligible in comparison to g, i.e. $\lim_{x \to a} \left\lvert \frac{f(x)}{g(x)} \right\rvert = 0$
$f \in \mathcal{O}(g)$	g is an asymptotic bound for f, i.e. $\limsup_{x \to a} \left\lvert \frac{f(x)}{g(x)} \right\rvert < \infty$
∂_i	partial derivative w.r.t. $x_i \in \mathbb{R}$, also denoted by ∂_{x_i}
∂_i^2	second partial derivative w.r.t. $x_i \in \mathbb{R}$, also denoted by $\partial_{x_i}^2$
∇f	gradient of a function $f : \mathbb{R}^d \to \mathbb{R}$, defined as $(\partial_1 f, \dots, \partial_d f)^\top$
$\nabla_\theta f$	gradient of a function $f : \mathbb{R}^p \to \mathbb{R}$ w.r.t. $\theta \in \mathbb{R}^p$, defined as $(\partial_{\theta_1} f, \dots, \partial_{\theta_p} f)^\top$
$\nabla^2 f$	Hessian matrix of a function $f : \mathbb{R}^d \to \mathbb{R}$, defined as $(\partial_i \partial_j f)_{1 \le i,j \le d}$
Δf	Laplace operator of a function $f : \mathbb{R}^d \to \mathbb{R}$, defined as $\sum_{i=1}^d \partial_i^2 f$
$D^\alpha f$	generic derivative, defined as $\frac{\partial^{\lvert\alpha\rvert} f}{\partial_{x_1}^{\alpha_1} \cdots \partial_{x_d}^{\alpha_d}}$, where $\alpha = (\alpha_1, \dots, \alpha_d)$ with $\alpha_i \in \mathbb{N}$ is a multi-index and $\lvert\alpha\rvert = \alpha_1 + \cdots + \alpha_n$ is the order of the index
D^k	collection of k-th order partial derivatives
$\frac{\delta}{\delta u} f(u)$	Gateaux derivative of functional f w.r.t. function u as defined in Definition 4.17
$f \circ g$	composition of functions f and g
$f(x)\big\rvert_{x=y}$	function evaluated at $x = y$
$\det A$	determinant of the matrix A
$\mathbb{E}[X]$	expectation value of random variable X
$\mathbb{E}_\nu[X]$	expectation value of random variable X w.r.t. the probability measure ν
$\mathbb{E}[X \lvert Y]$	expectation value of random variable X conditioned on the event Y
$\mathrm{Var}(X)$	variance of random variable X
$\mathrm{Cov}(X, Y)$	covariance between random variables X and Y
$\mathrm{Kurt}[X]$	Kurtosis of the random variable X as defined in (5.25)
$\mathbb{1}_A$	indicator function that is 1 of A is true and 0 otherwise

$C(A, B)$	set of continuous functions mapping from set A into set B
$C_b(A, B)$	set of continuous and bounded functions mapping from set A into set B
$C^k(A, B)$	set of k times differentiable functions mapping from set A into set B
$C^{k,l}(A, B)$	set of functions mapping from the set A into set B, which are k times differentiable in the first set of variables and l times in the second set of variables
inf	infimum
sup	spremum
min	minimum
$\arg\min$	minimizer, i.e. argument of the minimum
max	maximum
$x \wedge y$	$\min\{x, y\}$, i.e. minmum of $x \in \mathbb{R}$ and $y \in \mathbb{R}$
$\partial\mathcal{D}$	boundary of the domain $\mathcal{D}$

Special symbols used in this thesis

d	state space dimension
K	sample size
T	time horizon
N	number of grid points or time-steps
f	a given function $f : \mathbb{R}^d \to \mathbb{R}$
g	a given function $g : \mathbb{R}^d \to \mathbb{R}$
b	drift function $b : \mathbb{R}^d \times [0, T] \to \mathbb{R}^d$ in a stochastic process
σ	diffusion function $\sigma : \mathbb{R}^d \times [0, T] \to \mathbb{R}^{d\times d}$ in a stochastic process
u	control function $u : \mathbb{R}^d \times [0, T] \to \mathbb{R}^d$, see (1.4)
$\mathcal{U}$	set of admissible control functions, see (1.5)
W_s	standard d-dimensional Brownian motion
X_s	uncontrolled stochastic process as in (1.2)
X_s^u	stochastic process controlled with control function u as in (1.4)
$\mathcal{C}$	set of continuous paths
$\mathcal{P}$	set of probability measures
$\mathcal{Z}$	expectation value, quantity of interest
$\mathbb{P}$	path space measure, often related to the uncontrolled stochastic process (1.2)
$\mathbb{P}^u$	path space measure, often related to the controlled stochastic process (1.4)
$\mathbb{Q}$	target path space measure, usually defined as in (1.13)
ν, μ, λ	probability measures
$\mathcal{W}$	work functional, usually defined as in (1.8)
τ	random stopping time of a stochastic process
L	infinitesimal generator of a stochastic process, as defined in (1.19)
V	function $V : \mathbb{R}^d \times [0, T] \to \mathbb{R}$ that solves a general PDE
ψ	function $\psi : \mathbb{R}^d \times [0, T] \to \mathbb{R}$ that solves a linear PDE
h	function $h : \mathbb{R}^d \times [0, T] \times \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}$ specifying the nonlinearity in a PDE
J	expected control costs as defined in (1.16) or (2.6)
Δt	time increment $\Delta t > 0$
KL	Kullback-Leibler divergence as defined in Definition 3.1
χ^2	χ^2 divergence as defined in Definition 3.2
ELBO	evidence lower bound as defined in (5.2)

$\mathcal{N}(\mu, \Sigma)$	normal distribution with mean $\mu \in \mathbb{R}^d$ and covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$
$\mathcal{L}$	loss function
$D(\mathbb{P}_1 \vert \mathbb{P}_2)$	generic divergence between the measures $\mathbb{P}_1$ and $\mathbb{P}_2$
$p(x\vert y)$	conditional probability density
$p(x, y)$	joint probability density
$\mathbf{0}$	a vector full of zeros, i.e. $(0, \dots, 0)^\top$
$\mathbf{1}$	a vector full of ones, i.e. $(1, \dots, 1)^\top$
$\mathrm{Id}_{d \times d}$	$d \times d$ identity matrix
$\mathcal{D}$	domain in $\mathbb{R}^d$

Appendix B

Supplementary material and helpful theorems

B.1 Strong solutions of SDEs

Theorem B.1 (Strong solutions of SDEs)**.** *Consider the SDE*

$$\mathrm{d}X_s = b(X_s, s)\,\mathrm{d}s + \sigma(X_s, s)\,\mathrm{d}W_s, \qquad X_0 = x_{\text{init}}, \tag{B.1}$$

where W_s is a standard m-dimensional Brownian motion and $b : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d, \sigma : \mathbb{R}^d \times [0,T] \to \mathbb{R}^{d\times m}$ are measurable functions, for which we assume there exists $C > 0$ such that for all $x \in \mathbb{R}^d$ and $s \in [0,T]$

$$|b(x,s)| + |\sigma(x,s)|_F \leq C(1+|x|), \tag{B.2}$$

and for all $x, y \in \mathbb{R}^d$ and $s \in [0,T]$

$$|b(x,s) - b(y,s)| + |\sigma(x,s) - \sigma(y,s)|_F \leq C|x-y|. \tag{B.3}$$

We assume further that the initial condition x_{init} is independent of the Brownian Motion W_s with $\mathbb{E}\left[|x_{\text{init}}|^2\right] < \infty$. Then the SDE has a unique strong solution X_s with

$$\mathbb{E}\left[\int_0^t |X_s|^2 \mathrm{d}s\right] < \infty \tag{B.4}$$

for all $t \in [0,T]$.

Proof. See for instance [219, Theorem 5.2.1]. □

B.2 Itô formula

Theorem B.2 (Itô formula)**.** *Consider the stochastic process X_t defined by the SDE*

$$\mathrm{d}X_t = b(X_t, t)\,\mathrm{d}t + \sigma(X_t, t)\,\mathrm{d}W_t, \tag{B.5}$$

and define $Y_t := f(X_t, t)$. Then

$$\mathrm{d}Y_t = \left(\partial_t f(X_t,t) + \nabla f(X_t,t)\cdot b(X_t,t) + \frac{1}{2}(\sigma\sigma^\top)(X_t,t) : \nabla^2 f(X_t,t)\right)\mathrm{d}t + \nabla f(X_t,t)\cdot\sigma(X_t,t)\,\mathrm{d}W_t \tag{B.6a}$$

$$= (\partial_t + L)\, f(X_t,t)\,\mathrm{d}t + \nabla f(X_t,t)\cdot\sigma(X_t,t)\,\mathrm{d}W_t. \tag{B.6b}$$

Proof. [219, Theorem 4.2.1] □

B.3 Girsanov theorem

The Radon-Nikodym derivatives between two path space measures can be computed explicitly.

Theorem B.3 (Girsanov)**.** *Let $u \in \mathcal{U}$ and denote by $\mathbb{P}$ the path space measure associated to the diffusion* (1.2) *and by $\mathbb{P}^u$ the path space measure associated to* (1.4)*. Then $\mathbb{P}$ and $\mathbb{P}^u$ are equivalent. Moreover, the Radon-Nikodym derivative satisfies*

$$\frac{\mathrm{d}\mathbb{P}^u}{\mathrm{d}\mathbb{P}}(X) = \exp\left(\int_0^T \left(u^\top \sigma^{-1}\right)(X_s, s) \cdot \mathrm{d}X_s - \int_0^T (\sigma^{-1} b \cdot u)(X_s, s)\,\mathrm{d}s - \frac{1}{2}\int_0^T |u(X_s, s)|^2\,\mathrm{d}s\right) \tag{B.7}$$

Proof. The fact that the two measures are equivalent follows from the linear growth assumption on u (see (1.5)), combining Beneš' theorem with Girsanov's theorem, see [285, Proposition 2.2.1 and Theorem 2.1.1]. According to a slight generalization of [285, Theorem 2.4.2], we have

$$\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}_{\mathrm{W}}}(X) = \exp\left(\int_0^T (b(X_s, s) \cdot \sigma^{-2}(X_s, s)\,\mathrm{d}X_s - \frac{1}{2}\int_0^T (b \cdot \sigma^{-2} b)(X_s, s)\,\mathrm{d}s\right), \tag{B.8}$$

and

$$\frac{\mathrm{d}\mathbb{P}^u}{\mathrm{d}\mathbb{P}_{\mathrm{W}}}(X) = \exp\left(\int_0^T (b + \sigma u)(X_s, s) \cdot \sigma^{-2}(X_s, s)\,\mathrm{d}X_s - \frac{1}{2}\int_0^T \left((b + \sigma u) \cdot \sigma^{-2}(b + \sigma u)\right)(X_s, s)\,\mathrm{d}s\right), \tag{B.9}$$

where $\mathbb{P}_{\mathrm{W}}$ denotes the measure on $\mathcal{C}$ induced by

$$\mathrm{d}X_s = \sigma(X_s, s)\,\mathrm{d}W_s, \qquad X_0 = x_{\mathrm{init}}. \tag{B.10}$$

Using

$$\frac{\mathrm{d}\mathbb{P}^u}{\mathrm{d}\mathbb{P}}(X) = \frac{\mathrm{d}\mathbb{P}^u}{\mathrm{d}\mathbb{P}_{\mathrm{W}}}\frac{\mathrm{d}\mathbb{P}_{\mathrm{W}}}{\mathrm{d}\mathbb{P}}(X), \tag{B.11}$$

and inserting (B.8) and (B.9), we obtain the desired result. □

B.4 Auxiliary statements for the analysis on suboptimal importance sampling

In this section, we recall some known statements and provide some helpful additional results for the analysis of importance sampling with suboptimal proposal measures as studied in Chapter 3.

Corollary B.4 (Formula for path space relative error in a special case)**.** *If the difference $u^* - u$ does not depend on x, then*

$$r(u) = \left(\exp\left(\int_0^T |u^* - u|^2(s)\mathrm{d}s\right) - 1\right)^{\frac{1}{2}}. \tag{B.12}$$

Proof. This is a direct consequence of (3.36). For the reader's convenience, we provide an alternative proof. If $u^* - u$ does not depend on x, then the random variable

$$Y = -\int_0^T |u^* - u|^2(s)\mathrm{d}s + 2\int_0^T (u^* - u)(s) \cdot \mathrm{d}W_s \tag{B.13}$$

is normally distributed, with mean and variance given by

$$\mu = -\int_0^T |u^* - u|^2(s)\mathrm{d}s, \qquad \sigma^2 = 4\int_0^T |u^* - u|^2(s)\mathrm{d}s, \tag{B.14}$$

where the second expression follows from the Itô isometry. The random variable $\left(\frac{\mathrm{d}\mathbb{P}^{u^*}}{\mathrm{d}\mathbb{P}^u}(X^u)\right)^2 = e^Y$ is then log-normally distributed and we compute

$$\mathbb{E}\left[e^Y\right] = e^{\mu + \frac{\sigma^2}{2}} = e^{\frac{\sigma^2}{4}}, \tag{B.15}$$

which gives the desired statement. □

Lemma B.5. *Let $n, p, q > 1$ with $\frac{1}{p} + \frac{1}{q} = 1$, then it holds that*

$$\mathbb{E}\left[\left(\frac{\mathrm{d}\mathbb{P}^{u^*}}{\mathrm{d}\mathbb{P}^u}(X^u)\right)^n\right] \leq \mathbb{E}\left[\exp\left(\frac{nq(np-1)}{2}\int_0^T |u^* - u|^2(X_s^u, s)\mathrm{d}s\right)\right]^{\frac{1}{q}}. \tag{B.16}$$

Proof. Let us write $\delta(x,s) := (u^* - u)(x,s)$, and let $n, p, q > 1$, then, using the Hölder inequality with $\frac{1}{p} + \frac{1}{q} = 1$, it holds

$$\mathbb{E}_{\mathbb{P}^u}\left[\left(\frac{\mathrm{d}\mathbb{P}^{u^*}}{\mathrm{d}\mathbb{P}^u}\right)^n\right] = \mathbb{E}_{\mathbb{P}^u}\left[\exp\left(n\int_0^T \delta(X_s,s)\cdot \mathrm{d}W_s - \frac{n^2 p}{2}\int_0^T |\delta(X_s,s)|^2\mathrm{d}s + \frac{n(np-1)}{2}\int_0^T |\delta(X_s,s)|^2\mathrm{d}s\right)\right] \tag{B.17a}$$

$$\leq \mathbb{E}_{\mathbb{P}^u}\left[\exp\left(\int_0^T np\,\delta(X_s,s)\cdot \mathrm{d}W_s - \frac{1}{2}\int_0^T |np\,\delta(X_s,s)|^2\mathrm{d}s\right)\right]^{\frac{1}{p}} \tag{B.17b}$$

$$\mathbb{E}_{\mathbb{P}^u}\left[\exp\left(\frac{nq(np-1)}{2}\int_0^T |\delta(X_s,s)|^2\mathrm{d}s\right)\right]^{\frac{1}{q}} \tag{B.17c}$$

$$= \mathbb{E}_{\mathbb{P}^u}\left[\exp\left(\frac{nq(np-1)}{2}\int_0^T |\delta(X_s,s)|^2\mathrm{d}s\right)\right]^{\frac{1}{q}}. \tag{B.17d}$$

Note that, even though Hölder's inequality holds for $p, q \in [1, \infty]$, the inequality becomes useless for $q = 1$ and $p = \infty$. □

Proposition B.6 (Zero-variance property)**.** *We get a vanishing relative error $r(u) = 0$ if and only if $\delta = u^* - u = 0$, i.e. when having the optimal control $u = u^* = -\sigma^\top \nabla V$.*

Proof. The fact that $\delta = 0$ implies $r(u) = 0$ follows directly from (3.36) or (3.50). For the other direction note that $r(u) = 0$ implies $M_u(x,t) = \psi^2(x,t)$ (as defined in Proposition 3.22) for all $(x,t) \in \mathbb{R}^d \times [0,T]$ and therefore equation (3.46a) becomes

$$(\partial_t + L - \sigma u(x,t)\cdot\nabla - 2f(x,t) + |u(x,t)|^2)\psi^2(x,t) = 0. \tag{B.18}$$

Further note that due to the Kolmogorov backward equation it holds

$$(\partial_t + L - 2f(x,t))\psi^2(x,t) - |\sigma^\top \nabla\psi(x,t)|^2 = 0. \tag{B.19}$$

Combining these two PDEs brings

$$\psi^2(x,t)|u(x,t)|^2 - 2(\psi\sigma u\cdot\nabla\psi)(x,t) + |\sigma^\top\nabla\psi(x,t)|^2 = |(\psi u)(x,t) - \sigma^\top\nabla\psi(x,t)|^2 = 0, \tag{B.20}$$

which implies that

$$u = \sigma^\top \frac{\nabla\psi}{\psi} = \sigma^\top \nabla \log\psi = -\sigma^\top \nabla V. \tag{B.21}$$

□

B.4.1 Relative error of log-normal random variables

Let $Y \sim \mathcal{N}(\overline{\mu}, \overline{\Sigma})$ with arbitrary $\overline{\mu} \in \mathbb{R}^d, \overline{\Sigma} \in \mathbb{R}^{d\times d}$ and take $\gamma \in \mathbb{R}^d, c \in \mathbb{R}$, then $e^{\gamma\cdot Y + c}$ is log-normally distributed and its relative error is

$$r(\gamma, \overline{\Sigma}) = \sqrt{\frac{\mathbb{E}\left[e^{2(\gamma\cdot Y + c)}\right]}{\mathbb{E}\left[e^{\gamma\cdot Y + c}\right]^2} - 1} = \sqrt{\frac{\mathbb{E}\left[e^{2\gamma\cdot Y}\right]}{\mathbb{E}\left[e^{\gamma\cdot Y}\right]^2} - 1} = \sqrt{e^{\gamma\cdot\overline{\Sigma}\gamma} - 1}, \tag{B.22}$$

independent of c. With the setting and notation from Example 3.12 we can now for instance compute

$$e^{-g(\widetilde{X})}\frac{p}{\widetilde{p}^\varepsilon}(\widetilde{X}) = \exp\left(-\alpha\cdot\widetilde{X} + \log\frac{p}{\widetilde{p}^\varepsilon}\right) = \exp\left(\varepsilon\cdot\widetilde{X} - \mu\cdot(\alpha+\varepsilon) + \frac{1}{2}(\alpha+\varepsilon)\cdot\Sigma(\alpha+\varepsilon)\right) \tag{B.23}$$

and with $\gamma = \varepsilon, c = -\mu\cdot(\alpha+\varepsilon) + \frac{1}{2}(\alpha+\varepsilon)\cdot\Sigma(\alpha+\varepsilon), \overline{\Sigma} = \Sigma$ one therefore gets the relative error

$$r(\widetilde{p}^\varepsilon) = \sqrt{e^{\varepsilon\cdot\Sigma\varepsilon} - 1} \tag{B.24}$$

as stated in (3.22).

B.4.2 Asymptotic expansion in small noise diffusions

To get further intuition on the small noise diffusions defined in Section 3.2.2, let us consider the formal expansion of the solution to the HJB equation (3.60)

$$V = v_0 + \eta v_1 + \eta^2 v_2 + \dots. \tag{B.25}$$

Inserting into (3.60) (with $\sigma = \mathrm{Id}_{d\times d}$) and comparing the powers of η yields the PDEs

$$\partial_t v_0 + b \cdot \nabla v_0 - \frac{1}{2} |\nabla v_0|^2 = 0, \tag{B.26a}$$

$$\partial_t v_1 + \frac{1}{2}\Delta v_0 + b \cdot \nabla v_1 - \nabla v_0 \cdot \nabla v_1 = 0, \tag{B.26b}$$

$$\partial_t v_2 + \frac{1}{2}\Delta v_1 + b \cdot \nabla v_2 - \nabla v_0 \cdot \nabla v_2 - \frac{1}{2} |\nabla v_1|^2 = 0, \tag{B.26c}$$

and so on, where all but the first PDE are transport equations (see [274]). We note that (given some appropriate assumptions) we have $v_0 = V^0$, with V^0 being to solution to (3.62). In [96] it is proven that

$$\nabla V = \nabla V^0 + \eta \nabla v_1 + o(\eta), \tag{B.27}$$

where v_1 fulfills the PDE above and V is the solution to the original HJB equation (3.60).

B.5 The Ornstein-Uhlenbeck process and a solvable control problem

A convenient (toy) example for the analysis of stochastic processes is the Ornstein-Uhlenbeck process,

$$\mathrm{d}X_s = AX_s \,\mathrm{d}s + B \,\mathrm{d}W_s, \qquad X_t = x, \tag{B.28}$$

where $A, B \in \mathbb{R}^{d\times d}$ are given matrices and W_s is a d-dimensional Brownian motion. It is one of the rare case of processes for which the distribution of X_T for any given $T \in [t, \infty)$ with $t \geq 0$ is known explicitly, namely

$$(X_T | X_t = x) \sim \mathcal{N}(\mu_t, \Sigma_t) \tag{B.29}$$

with

$$\mu_t = e^{A(T-t)}x, \qquad \Sigma_t = \int_0^{T-t} e^{As} BB^\top e^{A^\top s} \,\mathrm{d}s = \int_t^T e^{A(T-s)} BB^\top e^{A^\top (T-s)} \,\mathrm{d}s. \tag{B.30}$$

In the following we will use this fact to conduct explicit calculations that might be helpful for illustrative purposes and for gaining further intuition on some of the questions that we discuss in this thesis.

B.5.1 Optimal control for Ornstein-Uhlenbeck dynamics with linear cost

Let us consider the Ornstein-Uhlenbeck process as defined in (B.28) and let us take $f(x) = 0$ and a linear observable $g(x) = \gamma \cdot x$ in the observable (1.8), where $\gamma \in \mathbb{R}^d$ is a prescribed vector, leading to the quantity of interest

$$\psi(x,t) = \mathbb{E}\left[e^{-\gamma \cdot X_T} | X_t = x\right]. \tag{B.31}$$

Since we know the distribution of X_T, we can compute

$$\psi(x,t) = \exp\left(-\gamma \cdot \left(\mu_t - \frac{1}{2}\Sigma_t \gamma\right)\right), \tag{B.32}$$

with μ_t and Σ_t as specified in (B.30). As demonstrated in Section 4.4.2 this quantity can be associated to an optimal control problem with linear terminal costs. It is one of the few examples where the optimal control function can be computed analytically. Using Lemma 2.11 we recall that the value function solving the HJB PDE (1.20) that corresponds to the optimal control problem fulfills $V(x,t) = -\log \psi(x,t)$, which brings

$$V(x,t) = \gamma \cdot \left(\mu_t - \frac{1}{2}\Sigma_t \gamma\right), \tag{B.33}$$

and therefore with (1.22) we note that

$$u^*(x,t) = -B^\top \nabla V(x,t) = -B^\top e^{A^\top (T-t)} \gamma \tag{B.34}$$

for the optimal control. We note it does not depend on the space variable.

B.6 Explicit calculations and illustrations for iterative diffusion optimizations

We extend the discussion on path space approximations from Chapter 4 by conducting some explicit calculations in order to get a better understanding of the iterative diffusion optimization (IDO) that we have introduced earlier. In particular we will again contrast the log-variance with the moment divergence and demonstrate potential advantages of the former.

B.6.1 IDO calculations for the Ornstein-Uhlenbeck process

Let us consider the Ornstein-Uhlenbeck process

$$\mathrm{d}X_s = AX_s\,\mathrm{d}s + B\,\mathrm{d}W_s, \qquad X_t = x_{\text{init}}, \tag{B.35}$$

as e.g. in Appendix B.5 and choose $f(x) = 0, g(x) = \gamma \cdot x$. We recall that the backward process as in (4.19) with $v = 0$ is given by

$$Y_T^u(y_0) = \widetilde{Y}_T^u + y_0, \qquad \widetilde{Y}_T^u = -\int_0^T u(X_s, s) \cdot \mathrm{d}W_s + \frac{1}{2}\int_0^T |u(X_s, s)|^2\,\mathrm{d}s. \tag{B.36}$$

Let us first consider the moment loss

$$\mathcal{L}_{\text{moment}}(u, y_0) = \mathbb{E}\left[(Y_T^u(y_0) - g(X_T))^2\right], \tag{B.37}$$

which we have already introduced in (4.27) and (4.28). Since we know from Appendix B.5.1 that the optimal control in our Ornstein-Uhlenbeck example is constant in x, it is reasonable to make the ansatz

$$u(x, t_n) = \theta_n \in \mathbb{R}^d \tag{B.38}$$

for $n \in \{0, \dots, N-1\}$. Let us collect all parameters $\mathbb{R}^{Nd+1} \ni (\theta, y_0) = (\theta_0, \dots, \theta_{N-1}, y_0)^\top$ and consider the Monte Carlo estimator of (B.37),

$$\widehat{\mathcal{L}}^K_{\text{moment}}(\theta, y_0) = \frac{1}{K}\sum_{k=1}^K \left(\widehat{Y}_N^{\theta,(k)}(y_0) - g(\widehat{X}_N^{(k)})\right)^2, \tag{B.39}$$

where $\widehat{X}$ is the Euler-Maruyama discretization of the forward process as defined in (2.89) and $\widehat{Y}_N^\theta$ is the discrete backward process with the optimal u^* replaced by u (which is again parametrized by θ) as in (B.36). Note that we can write the backward process explicitly as

$$\widehat{Y}_N(y_0) = y_0 - \sum_{n=0}^{N-1}\left(-\frac{1}{2}|\widehat{Z}_n|^2\Delta t + \widehat{Z}_n \cdot \xi_{n+1}\sqrt{\Delta t}\right) \tag{B.40}$$

which with our ansatz for u brings

$$\widehat{Y}_N^\theta(y_0) = y_0 + \sum_{n=0}^{N-1}\left(\frac{1}{2}|\theta_n|^2\Delta t - \theta_n \cdot \xi_{n+1}\sqrt{\Delta t}\right). \tag{B.41}$$

We can now explicitly compute the partial derivatives that we need for the minimization of the loss,

$$\partial_{y_0}\widehat{\mathcal{L}}^K_{\text{moment}}(\theta, y_0) = \frac{2}{K}\sum_{k=1}^K \left(\widehat{Y}_N^{\theta,(k)}(y_0) - g(\widehat{X}_N^{(k)})\right), \tag{B.42}$$

$$\partial_{\theta_{n,i}}\widehat{\mathcal{L}}^K_{\text{moment}}(\theta, y_0,) = \frac{2}{K}\sum_{k=1}^K \left(\theta_{n,i}\Delta t - \xi_{n+1,i}^{(k)}\sqrt{\Delta t}\right)\left(\widehat{Y}_N^{\theta,(k)}(y_0) - g(\widehat{X}_N^{(k)})\right) \tag{B.43}$$

for $n \in \{0, \dots, N-1\}, i \in \{1, \dots, d\}$. Alternatively, we can consider the log-variance loss

$$\mathcal{L}^{\log}_{\text{Var}}(u) = \text{Var}\left(\widetilde{Y}_T^u - g(X_T)\right) \tag{B.44}$$

as defined in (4.21), whose Monte Carlo version reads

$$\widehat{\mathcal{L}}^{\log,K}_{\text{Var}}(\theta) = \frac{1}{K-1}\sum_{k=1}^K \left(\widehat{\widetilde{Y}}_N^{\theta,(k)} - g(\widehat{X}_N^{(k)}) - \frac{1}{K}\sum_{k=1}^K\left(\widehat{\widetilde{Y}}_N^{\theta,(k)} - g(\widehat{X}_N^{(k)})\right)\right)^2. \tag{B.45}$$

In the particular setting considered above we can compute the partial derivatives of this loss to be

$$\partial_{\theta_{n,i}}\widehat{\mathcal{L}}_{\mathrm{Var}}^{\log,K}(\theta)=\frac{2}{K-1}\sum_{k=1}^{K}\left(-\xi_{n+1,i}^{(k)}\sqrt{\Delta t}+\frac{1}{K}\sum_{k=1}^{K}\xi_{n+1,i}^{(k)}\sqrt{\Delta t}\right)\left(\widehat{\widehat{Y}}_N^{\theta,(k)}-g(\widehat{X}_N^{(k)})-\frac{1}{K}\sum_{k=1}^{K}\left(\widehat{\widehat{Y}}_N^{\theta,(k)}-g(\widehat{X}_N^{(k)})\right)\right), \tag{B.46}$$

where compared to (B.43) we note the vanishing of the $\theta_{n,i}$ terms and the corresponding centerings of the two terms in the sum as explained in Remark 4.21.

Discretization errors

When considering the Ornstein-Uhlenbeck dynamics the discrete forward process can be written down explicitly as well, namely by

$$\widehat{X}_N=(\mathrm{I}+A\Delta t)^N\widehat{X}_0+\sum_{n=0}^{N-1}(\mathrm{I}+A\Delta t)^{N-n-1}B\,\xi_{n+1}\sqrt{\Delta t}, \tag{B.47}$$

where we use the shorthand notation $\mathrm{I}=\mathrm{Id}_{d\times d}$. Looking at our objective

$$\begin{aligned}\widehat{Y}_N(y_0)-g(\widehat{X}_N)=y_0+\sum_{n=0}^{N-1}\left(\frac{1}{2}|\widehat{u}_n^*(\widehat{X}_n)|^2\Delta t-\widehat{u}_n^*(\widehat{X}_n)\cdot\xi_{n+1}\sqrt{\Delta t}\right)\\ -\gamma\cdot\left((\mathrm{I}+A\Delta t)^N\widehat{X}_0+\sum_{n=0}^{N-1}(\mathrm{I}+A\Delta t)^{N-n-1}B\,\xi_{n+1}\sqrt{\Delta t}\right)\end{aligned} \tag{B.48}$$

we can deduce that the discrete optimal control corresponding to the problem above (that brings the corresponding losses to zero) is given by

$$\widehat{u}_n^*(x)=-B^\top(\mathrm{I}+A^\top\Delta t)^{N-n-1}\gamma, \tag{B.49}$$

which cancels the noise in the expression $\widehat{Y}_N(y_0)-g(\widehat{X}_N)$. This then leads to the optimal y_0 being

$$y_0=-\frac{1}{2}\sum_{n=0}^{N-1}|B^\top(\mathrm{I}+A^\top\Delta t)^{N-n-1}\gamma|^2\Delta t \tag{B.50}$$

in the discrete case. We note that due to

$$(\mathrm{I}+A^\top\Delta t)^{N-n-1}=(\mathrm{I}+A^\top\Delta t)^{\frac{T}{\Delta t}-\frac{t}{\Delta t}-1}\longrightarrow e^{A^\top(T-t)} \tag{B.51}$$

for $\Delta t\to 0$ and comparing to (B.34), we indeed have

$$\widehat{u}_n^*(x)\longrightarrow u^*(x,t). \tag{B.52}$$

Similarly we note

$$\sum_{n=0}^{N-1}|B^\top(\mathrm{I}+A^\top\Delta t)^{N-n-1}\gamma|^2\Delta t\longrightarrow\gamma\cdot\int_0^T e^{A(T-s)}BB^\top e^{A^\top(T-s)}\,\mathrm{d}s\,\gamma \tag{B.53}$$

and therefore $y_0\to V(X_0,0)$ for $\Delta t\to 0$. In Figure B.1 we illustrate the convergence of the discrete optimal control $\widehat{u}_n^*$ to its continuous version for decreasing step sizes Δt in the left panel and show that the convergence of the discrete optimal y_0 to the continuous one as in (B.53) is linear in Δt. For the Ornstein-Uhlenbeck case this can be seen by comparing $V(X_0,0)$ to expression (B.50) using a Taylor expansion in Δt around 0, namely

$$\begin{aligned}(I+A\Delta t)^{N-n-1}-e^{A(N-n)\Delta t}=\Delta t+\frac{(\Delta t)^2}{2}\left((n-N+1)(n-N+2)-(n-N)^2\right)\\ +\frac{(\Delta t)^3}{6}\left((n-N+1)(n-N+2)(n-N+3)-(n-N)^3\right)+\dots.\end{aligned} \tag{B.54}$$

Note that in general we have $|\widehat{Y}_0-Y_0|=\mathcal{O}(\sqrt{\Delta t})$ according to Theorem 2.32.

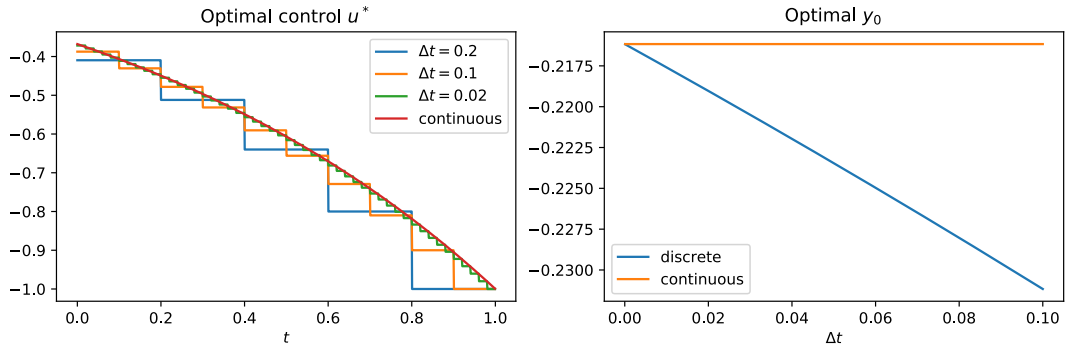


Figure B.1: Left: Convergence of the discrete optimal control $\widehat{u}_n^*(x)$ to its continuous version for decreasing step-sizes Δt. Right: We plot the discrete version of the optimal y_0 as a function of Δt noting its linear dependence.

In Figure B.2 we show approximations of the optimal control during different learning stages in IDO when either relying on the constant approximation as in (B.38) or on a neural network approximation, where time is modelled as an additional input dimension as described in Section 4.4.1. We use a step size of $\Delta t = 0.1$ and see that both approaches converge to the discrete optimal solution at the grid points. Interestingly the neural network approach seems to interpolate the times in between the grid points in a reasonable way, however still not agreeing with the continuous optimal control due to the expected discretization error.

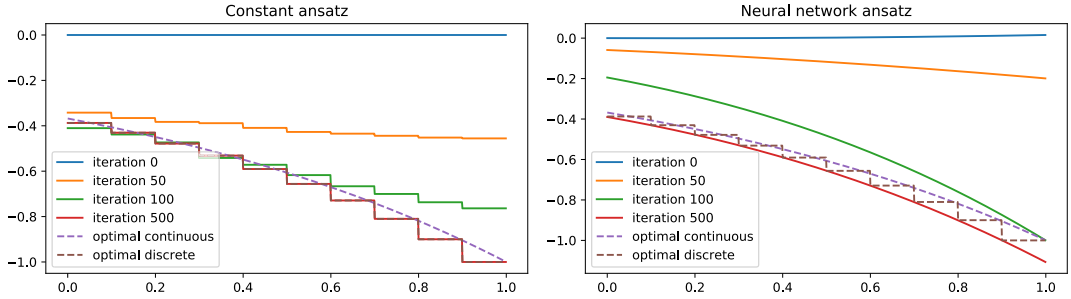


Figure B.2: Left: Approximations of the optimal control with constant ansatz functions for different learning iterations. Right: Approximations of the optimal control with a single neural network displayed at different learning iterations.

More generally, instead of taking the constant ansatz for the control as in (B.38) we can consider a linear combination of ansatz functions $\{\varphi_1, \dots, \varphi_M\} \subset C^1(\mathbb{R}^d, \mathbb{R})$, as e.g. in Section 6.2.1.1 and equation (4.65), aiming for

$$V(x, t_n) \approx \sum_{m=1}^{M} \theta_{n,m} \varphi_m(x), \qquad \text{i.e.} \qquad u^*(x, t_n) \approx -B^\top \sum_{m=1}^{M} \theta_{n,m} \nabla \varphi_m(x). \tag{B.55}$$

In this case we can compute the partial derivatives of the moment loss to be

$$\partial_{y_0} \widehat{\mathcal{L}}^K_{\text{moment}}(\theta, y_0) = \frac{2}{K} \sum_{k=1}^{K} \left(\widehat{Y}_N^{\theta,(k)}(y_0) - g(\widehat{X}_N^{(k)}) \right), \tag{B.56}$$

$$\begin{aligned} \partial_{\theta_{n,j}} \widehat{\mathcal{L}}^K_{\text{moment}}(\theta, y_0) = \frac{2}{K} \sum_{k=1}^{K} \Bigg(& BB^\top \nabla \varphi_j(\widehat{X}_n) \cdot \sum_{m=1}^{M} \theta_{n,m} \nabla \varphi_m(\widehat{X}_n^{(k)}) \Delta t + B^\top \nabla \varphi_j(\widehat{X}_n^{(k)}) \cdot \xi_{n+1}^{(k)} \sqrt{\Delta t} \Bigg) \\ & \left(\widehat{Y}_N^{\theta,(k)}(y_0) - g(\widehat{X}_N^{(k)}) \right) \end{aligned} \tag{B.57}$$

for $n \in \{0, \dots, N-1\}, j \in \{1, \dots, M\}$. The partial derivatives of the log-variance loss can be computed analogously.

Moment vs. log-variance loss: learning y_0 can slow down the optimization

Let us now consider a simple toy example that shall demonstrate the effect of learning the additional parameter y_0. To this end, let $d = 1$, consider the Ornstein-Uhlenbeck process (B.35), set $A = B = \gamma = T = 1$ and take

the ansatz

$$u(x,t) = -e^{\sin(2\pi\theta)(T-t)}, \tag{B.58}$$

which contains only one parameter $\theta \in \mathbb{R}$, noting that with $\theta \in \left\{\frac{4k-1}{4} : k \in \mathbb{Z}\right\}$ we recover the optimal control (B.34). Let us compare the moment loss with the log-variance loss, where we recall that the moment loss requires the optimization of the additional parameter y_0, for which in this example we can compute the optimal value to be $y_0 = -\frac{1}{4}\left(1 - e^{-2T}\right) \approx -0.216$. In Figure B.3 we display the L^2 error as defined in (4.67) along the gradient steps and see that using the log-variance loss is significantly faster, as we have similarly observed in more realistic examples (cf. Section 4.4, in particular Figure 4.5). In the central and right panels we visualize the loss landscapes of the moment and log-variance losses respectively by varying θ and y_0, noting again that the log-variance loss does not depend on y_0. We realize that the additional parameter y_0 in the moment loss complicates the loss landscape significantly. We further plot the trace of the parameters during the optimization process and realize that for the moment loss the optimization of θ depends on the current value of y_0 and vice versa, leading to harder optimization constraints and slower convergence.

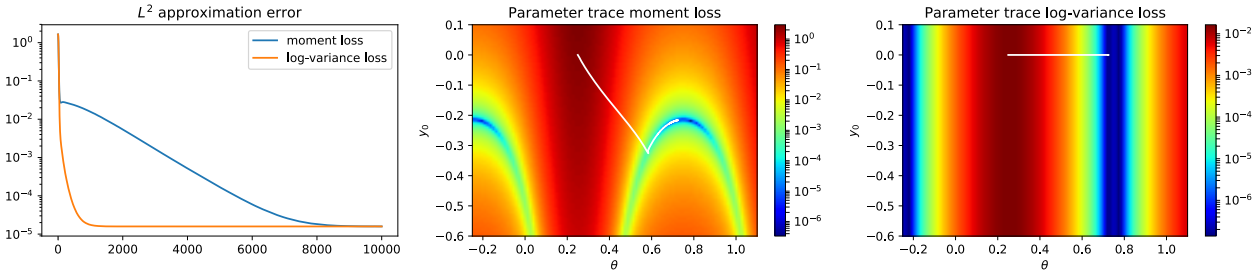


Figure B.3: Left: L^2 approximation error using either the moment or the log-variance loss for an Ornstein-Uhlenbeck toy example in $d = 1$. Middle: We plot the loss landscape of the moment loss depending on the two parameters y_0 and θ as well as the parameter trace during optimization (white line). Right: The loss landscape of the log-variance loss depends only on θ, which leads to a faster convergence of this parameter during training.

B.7 Dimension-dependence of the KL-divergence

The following lemma shows that the KL-divergence increases with the number of dimensions. This result follows from the chain-rule of KL divergence, see, e.g., [58].

Lemma B.7. *Let $u^{(D)}(z_1, \dots, z_D)$ and $v^{(D)}(z_1, \dots, z_D)$ be two arbitrary probability distributions on $\mathbb{R}^D$. For $J \in \{1 \dots, D\}$ denote their marginals on the first J coordinates by $u^{(J)}$ and $v^{(J)}$, i.e.*

$$u^{(J)}(z_1, \dots, z_J) = \int \cdots \int u^{(D)}(z_1, \dots, z_D)\, \mathrm{d}z_{J+1} \dots \mathrm{d}z_D, \tag{B.59}$$

and

$$v^{(J)}(z_1, \dots, z_J) = \int \cdots \int v^{(D)}(z_1, \dots, z_D)\, \mathrm{d}z_{J+1} \dots \mathrm{d}z_D. \tag{B.60}$$

Then

$$\mathrm{KL}(u^{(1)} \mid v^{(1)}) \leq \mathrm{KL}(u^{(2)} \mid v^{(2)}) \leq \dots \leq \mathrm{KL}(u^{(D)} \mid v^{(D)}), \tag{B.61}$$

i.e. the function $J \mapsto \mathrm{KL}(u^{(J)} \mid v^{(J)})$ is increasing.

B.8 The tensor train format

In this section we discuss the functional approximations $\widehat{\varphi}_n$, that we have used in Section 6.2 in order to solve high-dimensional parabolic PDEs, in terms of the tensor train format, leading to efficient optimization procedures for the schemes (6.27) and (6.42). This has been done in collaboration with Leon Sallandt and Nikolas Nüsken and published in [251] .

Encoding functions defined on high-dimensional spaces using traditional methods such as finite elements, splines or multi-variate polynomials leads to a computational complexity that scales exponentially in the state space dimension d. However, interpreting the coefficients of such ansatz functions as entries in a high-dimensional tensor allows us to use tensor compression methods to reduce the number of parameters. To this end, we define

a set of functions $\{\phi_1, \dots, \phi_m\}$ with $\phi_i : \mathbb{R} \to \mathbb{R}$, e.g. one-dimensional polynomials or finite elements. The approximation $\widehat{\varphi}$ of $V : \mathbb{R}^d \to \mathbb{R}$ takes the form

$$\widehat{\varphi}(x_1, \dots, x_d) = \sum_{i_1=1}^{m} \cdots \sum_{i_d=1}^{m} c_{i_1,\dots,i_d} \phi_{i_1}(x_1) \cdots \phi_{i_d}(x_d), \tag{B.62}$$

motivated by the fact that polynomials and other tensor product bases are dense in many standard function spaces [268]. Note that for the sake of simplicity we choose the set of ansatz functions to be the same in every dimension. As expected, the coefficient tensor $c \in \mathbb{R}^{m \times m \times \cdots \times m} \equiv \mathbb{R}^{m^d}$ suffers from the curse of dimensionality since the number of entries increases exponentially in the dimension d. In what follows, we review the tensor train format to compress the tensor c.

For the sake of readability we will henceforth write $c_{i_1,\dots,i_d} = c[i_1, \dots, i_d]$ and represent the contraction of the last index of a tensor $w_1 \in \mathbb{R}^{r_1 \times m \times r_2}$ with the first index of another tensor $w_2 \in \mathbb{R}^{r_2 \times m \times r_3}$ by

$$w = w_1 \circ w_2 \in \mathbb{R}^{r_1 \times m \times m \times r_3}, \tag{B.63a}$$

$$w[i_1, i_2, i_3, i_4] = \sum_{j=1}^{r_2} w_1[i_1, i_2, j] w_2[j, i_3, i_4]. \tag{B.63b}$$

In the literature on tensor methods, graphical representations of general tensor networks are widely used. In these pictorial descriptions, the contractions $\circ$ of the component tensors are indicated as edges between vertices of a graph. As an illustration, we provide the graphical representation of an order-4 tensor and a tensor train representation (see Definition B.8 below) in Figure B.4.

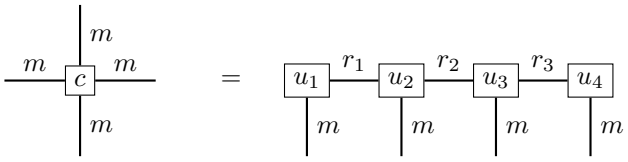


Figure B.4: An order 4 tensor and a tensor train representation.

Tensor train representations of c can now be defined as follows [220].

Definition B.8 (Tensor Train)**.** Let $c \in \mathbb{R}^{m \times \cdots \times m}$. A factorization

$$c = u_1 \circ u_2 \circ \cdots \circ u_d, \tag{B.64}$$

where $u_1 \in \mathbb{R}^{m \times r_1}$, $u_i \in \mathbb{R}^{r_{i-1} \times m \times r_i}$, $2 \le i \le d-1$, $u_d \in \mathbb{R}^{r_{d-1} \times m}$, is called *tensor train representation* of c. We say that u_i are *component tensors*. The tuple of the dimensions $(r_1, \dots, r_{d-1})$ is called the representation rank and is associated with the specific representation (B.64). In contrast to that, the tensor train rank (TT-rank) of c is defined as the minimal rank tuple $\mathbf{r} = (r_1, \dots, r_{d-1})$, such that there exists a TT representation of c with representation rank equal to $\mathbf{r}$. Here, minimality of the rank is defined in terms of the partial order relation on $\mathbb{N}^d$ given by

$$\mathbf{s} \preceq \mathbf{t} \iff s_i \le t_i \text{ for all } 1 \le i \le d,$$

for $\mathbf{r} = (r_1, \dots, r_d)$, $\mathbf{s} = (s_1, \dots, s_d) \in \mathbb{N}^d$.

It can be shown that every tensor has a TT-representation with minimal rank, implying that the TT-rank is well defined [140]. An efficient algorithm for computing a minimal TT-representation is given by the Tensor-Train-Singular-Value-Decomposition (TT-SVD) [221]. Additionally, the set of tensor trains with fixed TT-rank forms a smooth manifold, and if we include lower ranks, an algebraic variety is formed [183].

Introducing the compact notation

$$\phi : \mathbb{R} \to \mathbb{R}^m, \quad \phi(x) = [\phi_1(x), \dots, \phi_m(x)],$$

the TT-representation of (B.62) is then given as

$$\widehat{\varphi}(x) = \sum_{i_1}^{m} \cdots \sum_{i_d}^{m} \sum_{j_1}^{r_1} \cdots \sum_{j_{d-1}}^{r_{d-1}} u_1[i_1, j_1] u_2[j_1, i_2, j_2] \cdots u_d[j_{d-1}, i_d] \phi(x_1)[i_1] \cdots \phi(x_d)[i_d]. \tag{B.65}$$

The corresponding graphical TT-representation (with $d = 4$ for definiteness) is then given as follows:

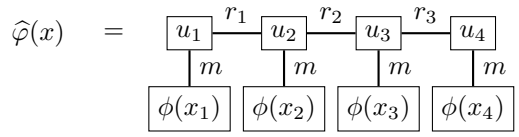


Figure B.5: Graphical representation of $\widehat{\varphi} : \mathbb{R}^4 \to \mathbb{R}$.

B.8.1 Optimization on the TT manifold

The multilinear structure of the tensor product enables efficient optimization of (6.27) and (6.42) within the manifold structure by means of reducing a high-dimensional linear equation in the coefficient tensor to small linear subproblems on the component tensors[45]. For this, we view (6.27) and (6.42) abstractly as least squares problems on a linear space $\mathcal{V} \subset L^2(\mathcal{D})$, where $\mathcal{D} \subset \mathbb{R}^d$ is a bounded Lipschitz domain. Our objective is then to find

$$\arg\min_{\widehat{\varphi} \in \mathcal{V}} \sum_{j=1}^{J} |\widehat{\varphi}(x_j) - R(x_j)|^2, \tag{B.66}$$

where $\{x_1, \dots, x_J\} \subset \mathcal{D}$ are data points obtained from samples of $\widehat{X}_n$, and $R : \mathcal{D} \to \mathbb{R}$ stands for the terms in (6.27) and (6.42) that are not varied in the optimization. Choosing a basis $\{b_1, \dots, b_M\}$ of $\mathcal{V}$ we can represent any function $w \in \mathcal{V}$ by $w(x) = \sum_{m=1}^{M} c_m b_m(x)$ and it is well known that the solution to (B.66) is given in terms of the coefficient vector

$$c = (A^\top A)^{-1} A^\top r \in \mathbb{R}^M, \tag{B.67}$$

where $A = [a_{ij}] \in \mathbb{R}^{J \times M}$ with $a_{ij} = b_j(x_i)$ and $r_j = R(x_j) \in \mathbb{R}^J$.

The *alternating least-squares (ALS) algorithm* [139] reduces the high-dimensional system (B.67) in the coefficient tensor c to small linear subproblems in the component tensors u_i as follows: Since the tensor train format (B.64) is a multilinear parametrization of c, fixing every component tensor but one (say u_i) isolates a remaining low-dimensional linear parametrization with associated local linear subspace $\mathcal{V}_{\mathrm{loc},i}$. The number M_i of remaining parameters (equivalently, the dimension of $\mathcal{V}_{\mathrm{loc},i}$) is given by the number of coefficients in the component tensor u_i, i.e. $M_i = r_{i-1}\, m\, r_i$. If the ranks r_i, r_{i-1} are significantly smaller than M, this results in a low-dimensional hence efficiently solvable least-squares problem. Iterating over the component tensors u_i then leads to an efficient scheme for solving high-dimensional least-squares problems with low rank structure. Basis functions in $\mathcal{V}_{\mathrm{loc},i}$ are obtained from the order 3 tensor b^{loc} depicted in Figure B.6 (note the three open edges). A simple reshape to an order one tensor then yields the desired basis functions, stacked onto each other, i.e. $b^{\mathrm{loc},i}(x) = [b_1^{\mathrm{loc},i}(x), b_2^{\mathrm{loc},i}(x), \dots, b_{M_i}^{\mathrm{loc},i}(x)]$.

More precisely, the local basis functions can be identified using the open edges in Figure B.6 as follows. Assuming u_2 is being optimized, we notice that the tensor $\phi(x_1) \circ u_1$ is a mapping from $\mathbb{R} \to \mathbb{R}^{r_1}$, which means that we can identify r_1 many one-dimensional functions. Note that this corresponds to the left part of the tensor picture in Figure B.6. Further, we have that $\phi(x_2)$ is a vector consisting of m one-dimensional functions, which is the middle part of the above tensor picture. The right part, consisting of the contractions between $\phi(x_2)$, u_3, u_4, and $\phi(x_4)$, is a set of two-dimensional functions with cardinality r_2. Taking the tensor product of the above functions yields an $r_1 m r_2$ dimensional function space of four-dimensional functions, which is exactly the span of the local basis functions.

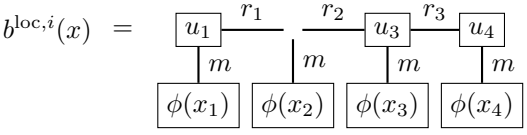


Figure B.6: Graphical representation of the local basis functions for $i = 2$.

In many situations the terminal condition g, defined in (6.8), is not part of the ansatz space just defined. This is always the case if g is not in tensor-product form. However, as the ambient space $\mathbb{R}^{m^d}$ is linear, g can be straightforwardly added[46] to the ansatz space, potentially increasing its dimension to $m^d + 1$. Whenever a

[45] In the case of (6.42), an additional nested iterative procedure is required, see Section B.8.2.

[46] We note that the idea of enhancing the ansatz space has been suggested in [307] in the context of linear parametrizations.

component tensor u_i is optimized in the way described above, we simply add g to the set of local basis functions, obtaining as a new basis

$$b_g^{\mathrm{loc},i} = \{b_1^{\mathrm{loc},i}, \dots, b_m^{\mathrm{loc},i}, g\}, \tag{B.68}$$

only marginally increasing the complexity of the least-squares problem. In our numerical tests we have noticed substantial improvements using the extension (B.68). Incorporating the terminal condition, the representation of the PDE solution takes the form depicted in Figure B.7, for some $c_g \in \mathbb{R}$.

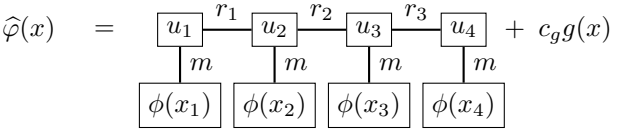


Figure B.7: Graphical representation of $\widehat{\varphi} : \mathbb{R}^4 \to \mathbb{R}$.

Summing up, we briefly state a basic ALS algorithm with our adapted basis $b^{\mathrm{loc},i}$:

Algorithm 5: Simple ALS algorithm

Input: initial guess $u_1 \circ u_2 \circ \cdots \circ u_d$.
repeat
  for $i = 1$ **to** d **do**
    identify the local basis functions (B.68), parametrized by u_k, $k \neq j$
    optimize u_i using the local basis by solving the local least squares problem
  end
until *noChange is true*;
Result: result $u_1 \circ u_2 \circ \cdots \circ u_d$.

The drawback of Algorithm 5 is that the ranks of the tensor approximation have to be chosen in advance. However, there are more involved rank-adaptive versions of the ALS algorithm, providing a convenient way of finding suitable ranks. Here we make use of the rank-adaptive *stable alternating least-squares algorithm (SALSA)* [115]. However, as seen in Section 6.2.3, we can in fact oftentimes find good solutions by setting the rank to be $(1, \dots, 1) \in \mathbb{N}^{d-1}$, enabling highly efficient computations.
By straightforward extensions, adding the terminal condition g to to set of local ansatz functions can similarly be implemented into more advanced, rank adaptive ALS algorithms, which is exactly what we do for our version of SALSA.

B.8.2 Handling implicit regression problems

The algorithms described in the previous section require the regression problem to be explicit such as in (6.27). In contrast, the optimization in (6.42) is of implicit type, as $\widehat{h}_n$ contains the unknown $\widehat{\varphi}_n$. In order to solve (6.42), we therefore choose an initial guess $\widehat{\varphi}_n^0$ and iterate the optimization of

$$\mathbb{E}[(\widehat{\varphi}_n^{k+1}(\widehat{X}_n) - h(\widehat{X}_n, t_n, \widehat{Y}_n^k, \widehat{Z}_n^k)\Delta t + \widehat{Z}_n^k \cdot \xi_{n+1}\sqrt{\Delta t} - \widehat{\varphi}_{n+1}(\widehat{X}_{n+1}))^2] \tag{B.69}$$

with respect to $\widehat{\varphi}_n^{k+1}$ until convergence (see Appendix B.9 for a discussion of appropriate stopping criteria). In the above display, $\widehat{Y}_n^k = \widehat{\varphi}_n^k(\widehat{X}_n)$ and $\widehat{Z}_n^k = \sigma^\top \nabla \widehat{\varphi}_n^k(\widehat{X}_n)$ are computed according to (6.9). For theoretical foundation, we guarantee convergence of the proposed scheme when the step size Δt is small enough.

Theorem B.9. *Assume that $\mathcal{V} \subset L^2(\mathcal{D}) \cap C_b^\infty(\mathcal{D})$ is a finite dimensional linear subspace, that $\sigma(x,t)$ is nondegenerate for all $(x,t) \in [0,T] \times \mathbb{R}^d$, and that h is globally Lipschitz continuous in the last two arguments. Then there exists $\delta > 0$ such that the iteration* (B.69) *converges for all $\Delta t \in (0, \delta)$.*

Proof. In this proof, we denote the underlying probability measure by $\mathbb{P}$, and the corresponding Hilbert space of random variables with finite second moments by $L^2(\mathbb{P})$. We define the linear subspace $\widetilde{\mathcal{V}} \subset L^2(\mathbb{P})$ by

$$\widetilde{\mathcal{V}} = \left\{ f(\widehat{X}_n) : f \in \mathcal{V} \right\}, \tag{B.70}$$

noting that $\widetilde{\mathcal{V}}$ is finite-dimensional by the assumption on $\mathcal{V}$, hence closed. The corresponding $L^2(\mathbb{P})$-orthogonal projection onto $\widetilde{\mathcal{V}}$ will be denoted by $\Pi_{\widetilde{\mathcal{V}}}$. By the nondegeneracy of σ, the law of $\widehat{X}_n$ has full support on $\mathcal{D}$, and

so $\|\cdot\|_{L^2(\mathbb{P})}$ is indeed a norm on $\widetilde{\mathcal{V}}$. Since $\widetilde{\mathcal{V}}$ is finite-dimensional, the linear operators

$$\widetilde{\mathcal{V}} \ni f(\widehat{X}_n) \mapsto \frac{\partial f}{\partial x_i}(\widehat{X}_n) \in L^2(\mathbb{P}) \tag{B.71}$$

are bounded, and consequently there exists a constant $C_1 > 0$ such that

$$\left\| \frac{\partial f}{\partial x_i}(\widehat{X}_n) \right\|_{L^2(\mathbb{P})} \leq C_1 \left\| f(\widehat{X}_n) \right\|_{L^2(\mathbb{P})}, \tag{B.72}$$

for all $i = 1, \dots, d$ and $f \in \mathcal{V}$. Furthermore, there exists a constant $C_2 > 0$ such that

$$\mathbb{E}\left[f^4(\widehat{X}_n) \right]^{1/4} := \left\| f(\widehat{X}_N) \right\|_{L^4(\mathbb{P})} \leq C_2 \left\| f(\widehat{X}_n) \right\|_{L^2(\mathbb{P})}, \tag{B.73}$$

for all $f \in \mathcal{V}$, again by the finite-dimensionality of $\widetilde{\mathcal{V}}$ and the fact that on finite dimensional vector spaces, all norms are equivalent. By standard results on orthogonal projections, the solution to the iteration (B.69) is given by

$$\varphi_n^{k+1}(\widehat{X}_n) = \Pi_{\widetilde{\mathcal{V}}}\big[-h(\widehat{X}_n, t_n, \widehat{Y}_n^k, \widehat{Z}_n^k)\Delta t + \widehat{Z}_n^k \cdot \xi_{n+1}\sqrt{\Delta t} - \widehat{\varphi}_{n+1}(\widehat{X}_{n+1}) \big]. \tag{B.74a}$$

We now consider the map $\Psi : \widetilde{\mathcal{V}} \to \widetilde{\mathcal{V}}$ defined by

$$f(\widehat{X}_n) \mapsto \Pi_{\widetilde{\mathcal{V}}}\big[-h(\widehat{X}_n, t_n, f(\widehat{X}_n), \sigma^\top \nabla f(\widehat{X}_n))\Delta t + \sigma^\top \nabla f(\widehat{X}_n) \cdot \xi_{n+1}\sqrt{\Delta t} - \widehat{\varphi}_{n+1}(\widehat{X}_{n+1}) \big]. \tag{B.75a}$$

For $F_1, F_2 \in \widetilde{\mathcal{V}}$ with $F_i = f_i(\widehat{X}_n)$, $f_i \in \mathcal{V}$, we see that

$$\begin{aligned}
\|\Psi F_1 - \Psi F_2\|_{L^2(\mathbb{P})} &= \big\|\Pi_{\widetilde{\mathcal{V}}}\big[-h(\widehat{X}_n, t_n, f_1(\widehat{X}_n), \sigma^\top \nabla f_1(\widehat{X}_n))\Delta t + h(\widehat{X}_n, t_n, f_2(\widehat{X}_n), \sigma^\top \nabla f_2(\widehat{X}_n))\Delta t && \text{(B.76a)} \\
&\quad + \sqrt{\Delta t}\left(\sigma^\top \nabla f_1(\widehat{X}_n) - \sigma^\top \nabla f_2(\widehat{X}_n) \right) \cdot \xi_{n+1} \big] \big\|_{L^2(\mathbb{P})} && \text{(B.76b)} \\
&\leq C_3 \left\|\Pi_{\widetilde{\mathcal{V}}}\right\|_{L^2(\mathbb{P})\to L^2(\mathbb{P})} \Bigg(\Delta t \left\| F_1 - F_2 \right\|_{L^2(\mathbb{P})} \\
&\qquad + \sqrt{\Delta t} \left\| \left(\sigma^\top \nabla f_1(\widehat{X}_n) - \sigma^\top \nabla f_2(\widehat{X}_n) \right) \cdot \xi_{n+1} \right\|_{L^2(\mathbb{P})} \Bigg) && \text{(B.76c)}
\end{aligned}$$

for some constant C_3 that does not depend on Δt, where we have used the triangle inequality, the Lipschitz assumption on h, the boundedness of σ, and the estimate (B.72). Using the Cauchy-Schwarz inequality, boundedness of σ as well as (B.72) and (B.73), the last term can be estimated as follows,

$$\begin{aligned}
\left\| \left(\sigma^\top \nabla f_1(\widehat{X}_n) - \sigma^\top \nabla f_2(\widehat{X}_n) \right) \cdot \xi_{n+1} \right\|_{L^2(\mathbb{P})} &\leq \left\| \left(\sigma^\top \nabla f_1(\widehat{X}_n) - \sigma^\top \nabla f_2(\widehat{X}_n) \right)^2 \right\|_{L^2(\mathbb{P})}^{1/2} \left\| \xi_{n+1}^2 \right\|_{L^2(\mathbb{P})}^{1/2} && \text{(B.77a)} \\
&\leq C_4 \left\| F_1 - F_2 \right\|_{L^2(\mathbb{P})}, && \text{(B.77b)}
\end{aligned}$$

where C_4 is a constant independent of Δt. Collecting the previous estimates, we see that $\delta > 0$ can be chosen such that for all $t \in (0, \delta)$, the mapping Ψ is a contraction on $\widetilde{\mathcal{V}}$ when equipped with the norm $\|\cdot\|_{L^2(\mathbb{P})}$, that is,

$$\|\Psi F_1 - \Psi F_2\| \leq \lambda \|F_1 - F_2\|, \tag{B.78}$$

for some $\lambda < 1$ and all $F_1, F_2 \in \widetilde{\mathcal{V}}$. Finally, the statement follows from the Banach fixed point theorem. □

Remark B.10. In order to ensure the boundedness assumption in Theorem B.9 and to stabilize the computation we add a regularization term involving the Frobenius norm of the coefficient tensor to the objective in (B.69). Choosing an orthonormal basis we can then relate the Frobenius norm to the associated norm in the function space by Parseval's identity. In our numerical tests we set our one-dimensional ansatz functions to be approximately $H^2(a,b)$-orthonormal[47], where a and b are set to be approximately equal to the minimum and maximum of the samples $\widehat{X}_n$, respectively. The corresponding tensor space $(H^2(a,b))^{\otimes d} = H^2_{\text{mix}}([a,b])^d$ can be shown to be continuously embedded in $W^{1,\infty}(\mathcal{D})$, guaranteeing boundedness of the approximations and their derivatives [268].

[47] Here, $H^2(a,b)$ refers to the second-order Sobolev space, see [268].

Remark B.11 (Parameter initializations). Since we expect $V(\cdot, t_n)$ to be close to $V(\cdot, t_{n+1})$ for any $n \in \{0, \dots, N-1\}$, we initialize the parameters of φ_n^0 as those obtained for $\widehat{\varphi}_{n+1}$ identified in the preceding time step.

Clearly, the iterative optimization of (B.69) is computationally more costly than the explicit scheme described in Section B.8.1 that relies on a single optimization of the type (B.66) per time step. However, implicit schemes typically ensure improved convergence orders as well as robustness [172] and therefore hold the promise of more accurate approximations (see Section 6.2.3 for experimental confirmation). We note that the NN based approaches considered as baselines in Section 6.2.3 perform gradient descent for both the explicit and implicit schemes and therefore no significant differences in the corresponding runtimes are expected.

B.9 Implementation details for backward iteration schemes

For the evaluation of the backward approximations we rely on reference values of $V(x_0, 0)$ and further define the following two loss metrics, which are zero if and only if the PDE is fulfilled along the samples generated by the discrete forward SDE (6.11). In the spirit of [242], we define the *PDE loss* as

$$\mathcal{L}_{\text{PDE}} = \frac{1}{KN} \sum_{n=1}^{N} \sum_{k=1}^{K} \Big((\partial_t + L) V(\widehat{X}_n^{(k)}, t_n) + h(\widehat{X}_n^{(k)}, t_n, V(\widehat{X}_n^{(k)}, t_n), \sigma^\top \nabla V(\widehat{X}_n^{(k)}, t_n)) \Big)^2, \tag{B.79}$$

where $\widehat{X}_n^{(k)}$ are realizations of (6.11), the time derivative is approximated with finite differences and the space derivatives are computed analytically (or with automatic differentiation tools). We leave out the first time step $n = 0$ since the regression problem within the explicit and the implicit schemes for the tensor trains are not well-defined due to the fact that $\widehat{X}_0^{(k)} = x_0$ has the same value for all k. We still obtain a good approximation since the added regularization term brings a minimum norm solution with the correct point value $V(x_0, 0)$. Still, this does not aim at the PDE being entirely fulfilled at this point in time.

Further, we define the *relative reference loss* as

$$\mathcal{L}_{\text{ref}} = \frac{1}{K(N+1)} \sum_{n=0}^{N} \sum_{k=1}^{K} \left| \frac{V(\widehat{X}_n^{(k)}, t_n) - V_{\text{ref}}(\widehat{X}_n^{(k)}, t_n)}{V_{\text{ref}}(\widehat{X}_n^{(k)}, t_n)} \right|, \tag{B.80}$$

whenever a reference solution for all x and t is available.

All computation times in the reported tables are measured in seconds.

Our experiments have been performed on a desktop computer containing an AMD Ryzen Threadripper 2990 WX 32x 3.00 GHz mainboard and an NVIDIA Titan RTX GPU, where we note that only the NN optimizations were run on this GPU, since our TT framework does not include GPU support. It is expected that running the TT approximations on a GPU will improve time performances in the future [1].

B.9.1 Details on tensor train approximation

For the implementation of the tensor networks we rely on the C++ library *xerus* [143] and the Python library *numpy* [125].

Within the optimization we have to specify the regularization parameter as noted in Remark B.10, which we denot here by $\eta > 0$. We adapt this parameter in dependence of the current residual in the regression problem (B.69), i.e. $\eta = cw$, where $c > 0$ and w is the residual from the previous sweep of SALSA. In every all our experiments we set $c_\eta = 1$. Further, we have to specify the condition "*noChange* is *true*" within Algorithm 5. To this end we introduce a test set with equal size as our training set. We measure the residual within a single run of SALSA on the test set and the training set. If the change of the residual on either of this sets is below $\delta = 0.0001$ we set *noChange* = *true*. For the fixed-point iteration we have a two-fold stopping condition. We stop the iteration if either the Frobenius norm of the coefficients has a smaller relative difference than $\gamma_1 < 0.0001$ or if the values $\widehat{\varphi}_n^{k+1}$ and $\widehat{\varphi}_n^k$ and their gradients, evaluated at the points of the test set, have a relative difference smaller than $\gamma_2 < 0.00001$. Note that the second condition is essentially a discrete H^1 norm, which is necessary since by adding the final condition into the ansatz space the orthonormal basis property is violated.

B.9.2 Details on neural network approximation

For the neural network architecture we rely on the *DenseNet*, which we have defined in Definition 2.50. We introduce the vector $r := (d_{\text{in}}, r_1, \dots, r_{L-1}, d_{\text{out}})$ to represent a certain choice of a DenseNet architecture, where in our setting $d_{\text{in}} = d$ and $d_{\text{out}} = 1$. If not otherwise stated we fix the parameter θ to be 1. For the activation function $\varrho : \mathbb{R} \to \mathbb{R}$, that is to be applied componentwise, we choose tanh.
For the gradient descent optimization we choose the Adam optimizer with the default parameters $\beta_1 = 0.9, \beta_2 = 0.999, \varepsilon = 10^{-8}$ [169]. In most of our experiments we chose a fixed learning rate η_{N-1} for the approximation of the first backward iteration step to approximate $\widehat{\varphi}_{N-1}$ and another fixed learning rate η_n for all the other iteration steps to approximate $\widehat{\varphi}_n$ for $0 \leq n \leq N-2$ (cf. Remark B.11). Similarly, we denote with G_{N-1} and G_n the amount of gradient descent steps in the corresponding optimizations.
In Tables B.1 and B.2 we list our hyperparameter choices for the neural network experiments that we have conducted.

HJB, $d = 10$, NN_{impl} Figure 6.1	HJB, $d = 100$, NN_{impl} Table 6.1, Figures 6.2, 6.3	HJB, $d = 100$, NN_{expl} Table 6.1, Figures 6.2, 6.3
$K = 2000, \Delta t = 0.01$	$K = 2000, \Delta t = 0.01$	$K = 2000, \Delta t = 0.01$
$r = (100, 110, 110, 50, 50, 1)$	$r = (100, 130, 130, 70, 70, 1)$	$r = (100, 110, 110, 50, 50, 1)$
$G_n = 8000, G_{N-1} = 40000$	$G_n = 5000, G_{N-1} = 40000$	$G_n = 500, G_{N-1} = 7000$
$\eta_n = 0.0001, \eta_{N-1} = 0.0001$	$\eta_n = 0.0001, \eta_{N-1} = 0.0003$	$\eta_n = 0.00005, \eta_{N-1} = 0.0003$

HJB double well $d = 50$, NN_{impl}, Table 6.3	HJB interacting double well $d = 20$, NN_{impl}, Table 6.4	CIR, $d = 100$, NN_{impl} Table 6.5
$K = 2000, \Delta t = 0.01$	$K = 2000, \Delta t = 0.01$	$K = 1000, \Delta t = 0.01$
$r = (50, 30, 30, 1)$	$r = (50, 20, 20, 20, 20, 1)$	$r = (100, 110, 110, 50, 50, 1)$
$G_n = 2000, G_{N-1} = 25000$	$G_n = 3000, G_{N-1} = 30000$	$G_n = 2000$ for $0 \leq n \leq 15$
$\eta_n = 0.0002, \eta_{N-1} = 0.0005$	$\eta_n = 0.0007, \eta_{N-1} = 0.001$	$G_n = 300$ for $16 \leq n \leq N-2$
		$G_{N-1} = 10000$
		$\eta_n = 0.00005, \eta_{N-1} = 0.0001$

Table B.1: Neural network hyperparameters for the experiments.

PDE with unbounded solution $d = 10$, NN_{impl}, Table 6.7	Allen-Cahn $d = 100$, NN_{impl}, Table 6.8
$K = 1000, \Delta t = 0.001$	$K = 8000, \Delta t = 0.01$
$r = (10, 30, 30, 1)$	$r = (10, 30, 30, 1)$
$G_n = 100, G_{N-1} = 10000$	$G_n = 10000$ for $0 \leq n \leq 5$
$\eta_n = 0.0001, \eta_{N-1} = 0.0001$	$G_n = 6000$ for $6 \leq n \leq N-2$
	$G_{N-1} = 15000$
	$\eta_n = 0.0002, \eta_{N-1} = 0.001$

Table B.2: Neural network hyperparameters for the additional experiments.

B.10 Conditioning of stochastic processes and the Schrödinger problem

Let us discuss how one can solve Problem 7.1, often also called *Schrödinger problem*, and how it is related to importance sampling of diffusions. We start with some preliminary statements that consider processes starting and ending at prescribed points, i.e. let us consider Problem 7.1 with $\mu_0 = \delta_{x_0}, \mu_T = \delta_z$ being Dirac distributions. Here a central object is the transition density connected to the uncontrolled stochastic process

$$\mathrm{d}X_s = b(X_s, s)\,\mathrm{d}s + \sigma(X_s, s)\,\mathrm{d}W_s, \qquad X_0 = x_{\text{init}}, \tag{B.81}$$

namely

$$p(s, y; t, x) = \mathbb{P}\left(X_s = y | X_t = x\right) \tag{B.82}$$

for $s > t$ (assuming again Assumption 1 for definiteness). The idea is to change this transition density in such a way that the target $\mu_T = \delta_z$ is reached. We note that for the conditioned transition probability it holds

$$\mathbb{P}\left(X_s = y | X_t = x, X_T = z\right) = \frac{\mathbb{P}\left(X_s = y, X_T = z | X_t = x\right)}{\mathbb{P}\left(X_T = z | X_t = x\right)} = \frac{\mathbb{P}\left(X_T = z | X_s = y\right)\mathbb{P}\left(X_s = y | X_t = x\right)}{\mathbb{P}\left(X_T = z | X_t = x\right)}, \tag{B.83}$$

which motivates to consider the transformation [153]

$$p^h(s, y; t, x) = \frac{p(s, y; t, x)p(T, z; s, y)}{p(T, z; t, x)}. \tag{B.84}$$

In fact one can show that this transformation leads to a controlled SDE of the form (7.4). The following theorem establishes this relation in a slightly more general form.

Theorem B.12 (Change of transition density)**.** *Let X_t be a solution of* (B.81) *and let*[48] *$h(x, t)$ be a strictly positive classical solution of*

$$(\partial_t + L)h(x, t) = 0, \tag{B.85}$$

i.e. $\mathbb{E}[h(X_s, s)|X_t = x] = h(x, t)$ for any $0 \le t < s < T$. Then the SDE

$$\mathrm{d}X_s = \left(b(X_s, s) + \sigma\sigma^\top \nabla \log h(X_s, s)\right) ds + \sigma(X_s, s)\,\mathrm{d}W_s \tag{B.86}$$

admits a solution and its transition density is given by

$$p^h(s, y; t, x) = \frac{p(s, y; t, x)h(y, s)}{h(x, t)}. \tag{B.87}$$

Proof. See [60, Theorem 2.1]. □

Choosing $h(x, t) = p(T, z; t, x)$ in Theorem B.12, noting that it solves (B.85) as required and that via the Chapman-Kolmogorov equation it holds

$$\mathbb{E}[p(T, z; X_s, s)|X_t = x] = \int_{\mathbb{R}^d} p(T, z; y, s)p(y, s; x, t)\mathrm{d}y = p(T, z; t, x) \tag{B.88}$$

justifies that the transformation (B.84) indeed leads to a conditioned stochastic process, as we make precise in the following corollary.

Corollary B.13 (Conditioning on point)**.** *For a fixed $z \in \mathbb{R}^d$ consider the strictly positive function*

$$h(x, t) = p(T, z; t, x) \tag{B.89}$$

and define $u^*(x, t) = \sigma^\top \nabla \log h(x, t)$, then there exists a solution to

$$\mathrm{d}X_s^{u^*} = \left(b(X_s^{u^*}, s) + \sigma u^*(X_s^{u^*}, s)\right)\mathrm{d}s + \sigma(X_s^{u^*}, s)\,\mathrm{d}W_s, \qquad X_0^{u^*} = x_{\text{init}}. \tag{B.90}$$

If X_s is a solution of (B.81)*, then $X_s^{u^*}$ has the conditioned law of X_s in the sense that*

$$\mathbb{E}[g(X_s^{u^*})] = \mathbb{E}[g(X_s)|X_T = z] \tag{B.91}$$

for any $s \in [0, T)$ and any bounded measurable function g.

[48] For historical reasons we are slightly inconsistent with our notation and note that h must not be confused with the nonlinearity appearing in our semi-linear PDEs as for instance stated in Definition 2.19.

Proof. See [60, Corollary 2.1]. □

Remark B.14 (Doob's h-transform and discrete path measures). The transformation (B.84) has originally been suggested in [75] and is therefore often termed *Doob's h-transform*. We note that we can write (B.87) as

$$p^h(s,y;t,x) = \frac{p(s,y;t,x)h(y,s)}{\mathbb{E}[h(X_s,s)|X_t = x]}, \tag{B.92}$$

which should be compared to the formula for the optimal change of measures, as stated e.g. in (1.13) and discussed in detail for path spaces in Chapter 4. In fact, the h-transform of the transition probability motivates to look at the discrete counterparts of path space measures. To this end, let us consider the discrete stochastic process $\widehat{X}_n$ on a time grid $0 = t_0 < t_1 < \cdots < t_N = T$ and note that the Markov property brings

$$\mathbb{P}(\widehat{X}_N = x_N, \dots, \widehat{X}_0 = x_0) = \prod_{n=0}^{N-1} p(t_{n+1}, x_{n+1}; t_n, x_n). \tag{B.93}$$

Similarly, we can define the discrete target measure as the product of h-transformed transition densities, namely

$$\mathbb{Q}(\widehat{X}_N = x_N, \dots, \widehat{X}_0 = x_0) = \prod_{n=0}^{N-1} p^h(t_{n+1}, x_{n+1}; t_n, x_n) = \prod_{n=0}^{N-1} \frac{p(t_{n+1}, x_{n+1}; t_n, x_n)h(x_{n+1}, t_{n+1})}{h(x_n, t_n)} \tag{B.94a}$$

$$= \mathbb{P}(\widehat{X}_N = x_N, \dots, \widehat{X}_0 = x_0)\frac{h(x_N, t_N)}{h(x_0, 0)}. \tag{B.94b}$$

In the importance sampling application (as discussed in Section 2.3.2) the choice $h(x,t) = \mathbb{E}[e^{-g(X_T)}|X_t = x]$ defines the optimal change of measure and we note that (B.94) can be understood as the discrete version of its continuous counterpart[49],

$$\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}(X) = \frac{e^{-g(X_T)}}{\mathbb{E}\left[e^{-g(X_T)}\right]}, \tag{B.95}$$

similarly to (1.13), see also [136]. A generalization to the path dependent functional $\mathcal{W}(X) = \int_0^T f(X_s, s)\mathrm{d}s + g(X_T)$ as for instance defined in (1.8) can be acquired by considering the function $h(x,t) = \mathbb{E}\left[e^{-\int_0^T f(X_s,s)\mathrm{d}s - g(X_T)}\middle| X_t = x\right]$ and the change of the transition density defined by [62]

$$p^h(s,y;t,x) = p(s,y;t,x)\frac{h(y,s)}{h(x,t)}e^{-\int_t^s f(X_r,r)\mathrm{d}r}, \tag{B.96}$$

where now $h(x,t)$ fulfills the PDE $(\partial_t + L - f)h(x,t) = 0$ (see Section 2.2.1), corresponding to the change of path measures defined by

$$\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}(X) = \frac{e^{-\mathcal{W}(X)}}{\mathcal{Z}}, \tag{B.97}$$

as frequently considered in this thesis.

Example B.15 (Brownian bridge)**.** *A prominent example for a conditioned processes is the so-called Brownian bridge, which conditions Brownian motion to end at a specified point $z \in \mathbb{R}^d$. Let us for instance consider the process $X_s = W_s + x_0$, which admits the transition density*

$$p(T,z;t,x) = \frac{1}{\sqrt{(2\pi)^d(T-t)}}\exp\left(-\frac{|x-z|^2}{2(T-t)}\right). \tag{B.98}$$

According to Corollary B.13 the conditioned process can be readily computed as[50]

$$\mathrm{d}\widetilde{X}_s = -\frac{\widetilde{X}_s - z}{T-s}\mathrm{d}s + \mathrm{d}W_s, \qquad \widetilde{X}_0 = x_0, \tag{B.99}$$

where we note that the control is small at the beginning and can get infinitely large for $s \to T$ unless $\widetilde{X}_s \to z$.

[49] We slightly abuse notation with $\mathbb{P}$ and $\mathbb{Q}$ either denoting probabilities of discrete events or probabilities of continuous paths of stochastic processes, as used throughout this thesis.

[50] Note that this Ornstein-Uhlenbeck process can be explicitly written as $\widetilde{X}_t = x_0\left(1 - \frac{t}{T}\right) + z\frac{t}{T} + (T-t)\int_0^t \frac{\mathrm{d}W_s}{T-s}$, from which one can immediately read off the desired conditioning.

Let us move on towards approaching Problem 7.1 and now consider stochastic processes that are conditioned to end in a prescribed target distribution rather than at a given point. To this end, let us define the operator S_t acting on densities by

$$S_t\mu(x) = \int_{\mathbb{R}^d} p(t,x;0,y)\mu(y)\mathrm{d}y. \tag{B.100}$$

The following theorem shows how Problem 7.1 can be solved if μ_0 is a Dirac measure concentrated at $x_0 \in \mathbb{R}^d$.

Theorem B.16 (Conditioning on target density)**.** *Let $\mu_0 = \delta_{x_0}$ and assume that $\mathrm{KL}(\mu_T|S_T\mu_0) < \infty$. Define the function*

$$h(x,t) = \int_{\mathbb{R}^d} p(T,z;t,x)\frac{\mu_T}{S_T\mu_0}(z)\mathrm{d}z = \mathbb{E}\left[\frac{\mu_T}{S_T\mu_0}(X_T)\middle| X_t = x\right]. \tag{B.101}$$

Then $u^* = \sigma^\top \nabla \log h$ solves Problem 7.1.

Proof. With (B.87) it can easily be seen that

$$p^h(T,z;0,x_0) = \frac{p(T,z;0,x_0)h(z,T)}{h(x_0,0)} = p(T,z;0,x_0)\frac{\mu_T}{S_T\mu_0}(z) = \mu_T(z). \tag{B.102}$$

For further rigorous details see [60, Theorem 3.1]. □

Remark B.17 (Control interpretation). We have seen before that PDEs of Feynman-Kac type, such as the one in (B.85), can be related to stochastic optimal control problems by means of a logarithmic transformation, see e.g. Lemma 2.11. Likewise, it can readily be seen that identifying the drift in Theorem B.16 that pushes the diffusion to the specified target distribution at terminal time while having minimal control costs corresponds to solving an optimal control problem with terminal costs $g(x) = -\log \frac{\mu_T}{S_T\mu_0}(x)$. To be precise it holds that

$$-\log h(x,t) = \min_{u\in\mathcal{U}} J(u;x,t) = \min_{u\in\mathcal{U}} \mathbb{E}\left[\frac{1}{2}\int_t^T |u(X_s^u,s)|^2\mathrm{d}s - \log\frac{\mu_T}{S_T\mu_0}(X_T^u)\middle| X_t^u = x\right]. \tag{B.103}$$

It is interesting to note that the optimal control costs of this problem are given by

$$J(u^*;x_0,0) = \mathrm{KL}(\mathbb{P}^{u^*}|\mathbb{P}) = \mathrm{KL}(\mu_T|S_T\mu_0), \tag{B.104}$$

i.e., the "global" KL divergence equals the KL divergence between the final densities [60].

Remark B.18 (Relation to importance sampling). Note that we can relate Theorem B.16 to our importance sampling considerations from Section 2.3.2 in the special case where $\mathcal{W}$ does not depend on f. We have already argued in Remark 2.35 that in this case the reweighting on path space can be reduced to a reweighting of the terminal density given by

$$q_T(x) = \frac{e^{-g(x)}p_T(x)}{\mathbb{E}\left[e^{-g(x)}\right]}, \tag{B.105}$$

which motivates defining our target density to be $\mu_T = q_T$. Going back to the path space perspective we can now use formula (B.101) to get

$$h(x,t) = \mathbb{E}\left[\frac{\mu_T}{S_T\mu_0}(X_T)\middle| X_t = x\right] = \frac{\mathbb{E}\left[e^{-g(X_T)}\middle| X_t = x\right]}{\mathbb{E}\left[e^{-g(X_T)}\right]} \tag{B.106}$$

and therefore the optimal control

$$u^*(x,t) = \sigma^\top\nabla\log h(x,t) = \sigma^\top\nabla\log\mathbb{E}\left[e^{-g(X_T)}\middle| X_t = x\right], \tag{B.107}$$

which according to Theorem 2.33 is just the zero-variance control in path space importance sampling. In other words, for the case $f = 0$ the problem of identifying an optimal importance sampling control corresponds exactly to the problem of reaching the target density q_T as defined in (B.105) while minimizing quadratic control costs, as stated in Problem 7.1.

Example B.19 (Conditioning and importance sampling of Brownian motion)**.** *We shall illustrate the connection between conditioning and importance sampling in the following toy example. Let us consider Brownian motion*

$$X_s = W_s \tag{B.108}$$

and let us aim at computing[51] $\mathbb{E}\left[|X_T|^2\right]$ *by sampling. We can determine an optimal target density to be*

$$q_T(x) = \frac{|x|^2}{\mathbb{E}\left[|X_T|^2\right]} p_T(x) = \frac{|x|^2}{dT\sqrt{(2\pi)^d T}} \exp\left(-\frac{|x|^2}{2T}\right) \tag{B.109}$$

and an optimal importance sampling control to be

$$u^*(x,t) = \nabla \log \mathbb{E}\left[|X_T|^2 | X_t = x\right] = \nabla \log\left(d(T-t) + |x|^2\right) = \frac{2x}{d(T-t)+|x|^2}. \tag{B.110}$$

For such a control the quantity $|X_T^{u^*}|^2 \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^{u^*}}(X^{u^*})$ *must be almost surely constant according to Theorem 2.33. Let us do a simulation in* $d = 2$ *for* $T = 1$*. In Figure B.8 we can see that* $|X_t^{u^*}|^2 \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^{u^*}}(X^{u^*})$ *as a function of time indeed starts and ends in one point and that the target density as given by* (B.109) *is reached.*

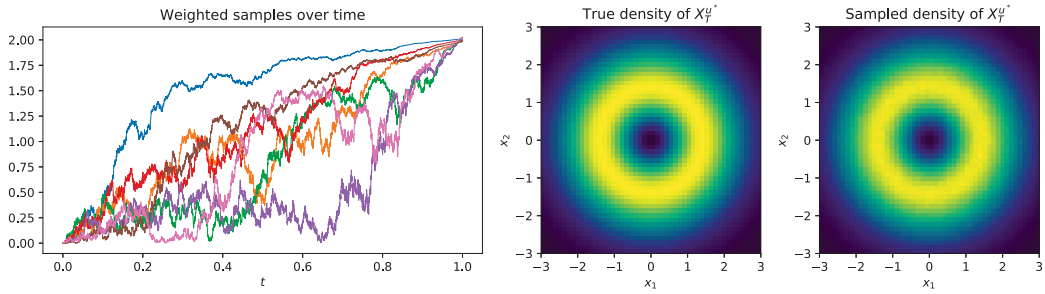


Figure B.8: In the left panel we display the quantity $|X_t^{u^*}|^2 \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^{u^*}}(X^{u^*})$ depending on time. In the two right panels we compare the true density of $X_T^{u^*}$ given by (B.109) with its histogram approximation using an Euler-Maruyama discretization of the controlled stochastic process.

Figure B.9 provides an illustration of the conditioning. Due to the constraint of $|X_T^{u^*}|^2 \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^{u^*}}(X^{u^*})$ *being almost surely constant, the different reweighted trajectories that start at the origin are conditioned to be placed on a straight line at terminal time.*

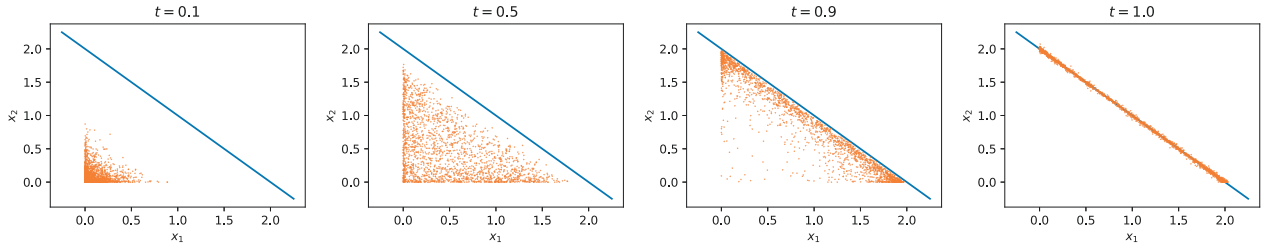


Figure B.9: We display the components of the controlled and reweighted process $(X_t^{u^*})^2 \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^{u^*}}(X_t^{u^*})$ (where the square is to be understood componentwise) at different times. For $t = T$ the quantity $|X_T^{u^*}|^2 \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^{u^*}}(X_T^{u^*})$ is almost surely constant, to be precise $(X_T^{u^*})_1^2 \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^{u^*}}(X_t^{u^*}) + (X_T^{u^*})_2^2 \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^{u^*}}(X_t^{u^*}) = 2$, which implies that the different realizations are conditioned to lie on the blue line.

At the end of this section let us briefly discuss how one can approach Problem 7.1 in full generality, i.e. condition a process on starting from and ending at arbitrary densities μ_0 and μ_T, where we will however leave algorithms that are e.g. based on a control formulation for future work. First, let us state the following representation of μ_0 and μ_T that will motivate a fruitful approach.

Theorem B.20. *Given two probability measures* μ_0, μ_T *on* $\mathbb{R}^d$ *and a transition density* $p(s,y;t,x)$ *there exists a unique pair of probability densities* (ν_0, ν_T) *such that the joint density defined by*

$$p(T,y;0,x)\nu_0(x)\nu_T(y) \tag{B.111}$$

has marginals μ_0, μ_T*.*

[51] Note that comparing to the notation from e.g. (2.102) this corresponds to choosing $g(x) = -\log(|x|^2)$, which is ∞ for $x = 0$, however $e^{-\infty} = 0$ justifies this choice.

Proof. See [31, 152, 153]. □

The above theorem suggests considering the so-called Schrödinger system defined by the two equations

$$\mu_0(x) = \nu_0(x) \int_{\mathbb{R}^d} p(T, y; 0, x)\nu_T(y)\mathrm{d}y, \tag{B.112a}$$

$$\mu_T(y) = \nu_T(y) \int_{\mathbb{R}^d} p(T, y; 0, x)\nu_0(x)\mathrm{d}x, \tag{B.112b}$$

in the unknowns ν_0, ν_T. Those quantities will turn out to be the essential ingredient in solving Problem 7.1. Motivated by (B.112) let us define

$$h(x,t) = \int_{\mathbb{R}^d} p(T, y; t, x)\nu_T(y)\mathrm{d}y, \tag{B.113}$$

$$\bar{h}(y,s) = \int_{\mathbb{R}^d} p(s, y; 0, z)\nu_0(z)\mathrm{d}z. \tag{B.114}$$

We see that h is the solution of the parabolic backward PDE

$$(\partial_t + L)\, h(x,t) = 0, \qquad h(x,T) = \nu_T(x) \tag{B.115}$$

and $\bar{h}$ solves the forward PDE

$$(\partial_t - L)\, \bar{h}(x,t) = 0, \qquad \bar{h}(x,0) = \nu_0(x). \tag{B.116}$$

The backward PDE (B.115) should be compared to (B.85), where however now the terminal condition is unknown.

Finally, the following theorem shows that with h as defined in (B.113) we can indeed solve Problem 7.1.

Theorem B.21 (Schrödinger problem with general distributions)**.** *Let*

$$\int_{\mathbb{R}^d} x^2\mu_0(x)\mathrm{d}x < \infty, \qquad \mathrm{KL}(\mu_T|S_T\nu_0) < \infty, \qquad \int_{\mathbb{R}^d} \frac{\nu_0}{\mu_0}(x)\mu_0(x)\mathrm{d}x < \infty \tag{B.117}$$

and

$$h(x,t) = \int_{\mathbb{R}^d} p(T, z; x, t)\nu_T(z)\mathrm{d}z = \mathbb{E}\left[\nu_T(X_T)\middle| X_t = x\right]. \tag{B.118}$$

Then $u^* = \sigma^\top \nabla \log h$ solves Problem 7.1.

Proof. See [60, Theorem 3.2]. □

We expect fruitful connections potentially leading to useful algorithms by studying control theoretic formulations of the general Schrödinger problem, similar to the one discussed in Remark B.17 that is related to the special case of starting from a Dirac distribution. We will leave this topic for future research.

B.11 Learning optimal importance sampling proposal densities

In Chapter 7 we have stated how one can aim to learn optimal importance sampling densities by relying on the log-variance loss. Here we will provide some additional examples and comment on alternative losses that might by suitable.

Remark B.22 (Alternative losses for learning optimal importance sampling densities)*.* In analogy to Section 4.1.1 one can also take others losses for learning optimal importance sampling densities. We can for instance consider the variance loss

$$\mathcal{L}_{\mathrm{Var}_r}(\widetilde{p}) = \mathrm{Var}_r\left(\frac{p}{\widetilde{p}}(X)e^{-g(X)}\right), \tag{B.119}$$

however we expect numerical issues in high-dimensional settings, as demonstrated in Proposition 4.29. Alternatively, the KL divergence leads to either the relative entropy loss

$$\mathcal{L}_{\mathrm{RE}}(\widetilde{p}) = \mathrm{KL}(\widetilde{p}|q) = \mathbb{E}_{\widetilde{p}}\left[\log\left(\frac{\widetilde{p}(X)\mathcal{Z}}{p(X)e^{-g(X)}}\right)\right] = \mathbb{E}_{\widetilde{p}}\left[\log\widetilde{p}(X) - \log p(X) + g(X)\right] + \log\mathcal{Z}, \tag{B.120}$$

or, by reversing the arguments, to the cross-entropy loss

$$\mathcal{L}_{\mathrm{CE}}(\widetilde{p}) = \mathrm{KL}(q|\widetilde{p}) = \mathbb{E}_{p}\left[\log\left(\frac{p(X)e^{-g(X)}}{\widetilde{p}(X)\mathcal{Z}}\right)\frac{e^{-g(X)}}{\mathcal{Z}}\right] = \mathbb{E}_{p}\left[\Big(\log p(X) - g(X) - \log\widetilde{p}(X)\Big)\frac{e^{-g(X)}}{\mathcal{Z}}\right] - \log\mathcal{Z}. \tag{B.121}$$

In contrast to the variance-based losses, those two losses do not naturally posses the property of allowing for arbitrary reference distributions r. For the cross-entropy loss we expect numerical issues in high-dimensional settings (cf. Proposition 4.29) and for the relative entropy loss we note that differentiating can be challenging since the random variables themselves are distributed according to $\widetilde{p}$ (cf. Chapter 5).

For the computation of the optimal proposal density as defined in (7.17) one can consider an example where an analytical solution is available and where an easy parametrization of $\widetilde{p}$ can be identified.

Example B.23 (Sampling from high-dimensional Gaussians)**.** *Suppose we want to compute $\mathbb{E}\left[e^{-\alpha\cdot X}\right]$, where $\alpha \in \mathbb{R}^d$ is a given vector and where $X \sim \mathcal{N}(\mu, \Sigma) =: p$ is distributed according to a multidimensional Gaussian with mean $\mu \in \mathbb{R}^d$ and covariance matrix $\Sigma \in \mathbb{R}^{d\times d}$. Naive Monte Carlo estimators might suffer from large relative errors, in particular in high-dimensional settings, cf. Example 3.12. We therefore rely on importance sampling, where the optimal importance sampling density can be computed to be*

$$q(x) = \frac{e^{-\alpha\cdot x}}{\mathcal{Z}}p(x) = \mathcal{N}(x; \mu - \Sigma\alpha, \Sigma). \tag{B.122}$$

Let us define our proposal to be Gaussian as well, namely

$$\widetilde{p}(x) = \mathcal{N}(x; \widetilde{\mu}, \widetilde{\Sigma}), \tag{B.123}$$

and aim to learn its parameters $\widetilde{\mu}, \widetilde{\Sigma}$. Since the covariance matrix can be written in its Cholesky decomposition $\widetilde{\Sigma} = \widetilde{K}\widetilde{K}^\top$, where $\widetilde{K}$ is a triangular matrix, we have to learn $p = d + \frac{d(d+1)}{2}$ parameters. We run an experiment in dimension $d = 9$, where the entries of μ and K are drawn from a standard normal distribution once at the beginning of the experiment. We initialize $\widetilde{\mu}$ and $\widetilde{K}$ with μ and K and minimize the log-variance loss (7.20) with the Adam optimizer using the learning rate $2 \cdot 10^{-4}$. In Figure B.10 we can see that the log-variance loss decreases over the iteration steps. In the center we plot the Euclidean distance of the current $\widetilde{\mu}$ to the optimal $\mu^* = \mu - \Sigma\alpha$, noting that it approaches zero. In the right panel we display the variance of the importance sampling estimator over the gradient steps noting that it decreases significantly. In fact, compared to the naive Monte Carlo estimator the relative error of the learnt importance sampling estimator reduces from roughly 10 to 10^{-4}.

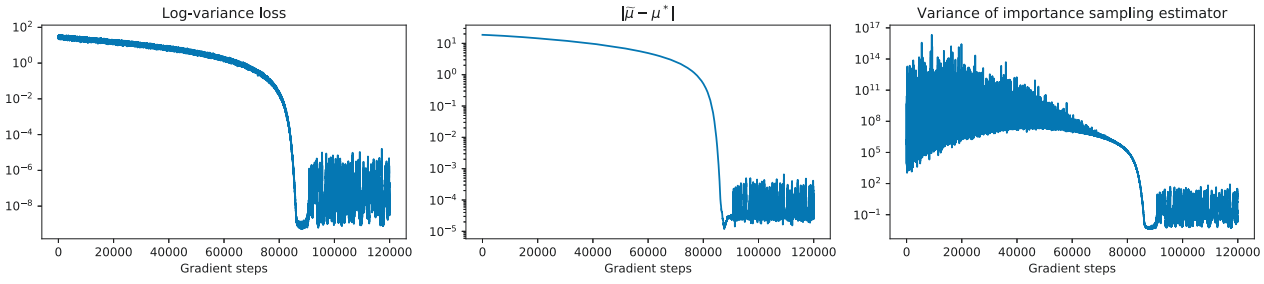


Figure B.10: Left: Log-variance loss over the iteration steps. Middle: Euclidean distance of the current $\widetilde{\mu}$ to the optimal $\mu^* = \mu - \Sigma\alpha$ over the gradient steps. Right: Variance of the importance sampling estimator using the corresponding proposal approximation.

In addition to Example 7.3, the following example relies on the normalizing flow attempt described in Chapter 7.

Example B.24 (Non-Gaussian target, more complex shape)**.** *Let us consider an example where the target density q is slightly more complicated by taking*

$$g(x) = -\log(|x|) - \alpha\cdot x. \tag{B.124}$$

We consider $d = 2$ *and for* p *again the standard Gaussian. In Figure B.11 we see that we are still able to learn the optimal importance sampling density quite well. The relative error of the corresponding importance sampling estimator reduces from roughly* 1 *to* 10^{-2}.

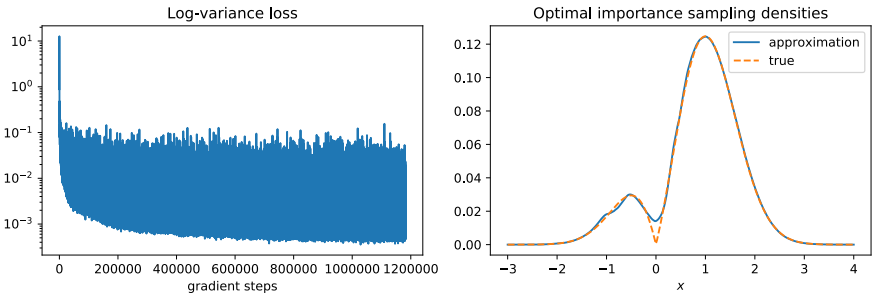


Figure B.11: Left: Log-variance loss along the training iterations. Right: Approximation of the optimal (non-Gaussian) proposal density.

Appendix C

Proofs

C.1 Proofs for Chapter 2

Proof of Lemma 2.11. We consider the transformation $\psi = e^{-V}$ and compute

$$Le^{-V(x,t)} = -b(x,t)\cdot\nabla V(x,t)e^{-V(x,t)} - \frac{1}{2}\left(\sum_{i,j=1}^{d}(\sigma\sigma^\top(x,t))_{ij}\partial_i\left(\partial_j V(x,t)e^{-V(x,t)}\right)\right) \tag{C.1a}$$

$$= -e^{-V(x,t)}\Bigg(b(x,t)\cdot\nabla V(x,t) + \frac{1}{2}\Bigg(\sum_{i,j=1}^{d}(\sigma\sigma^\top(x,t))_{ij}\partial_i\partial_j V(x,t) - \sum_{i,j=1}^{d}(\sigma\sigma^\top(x,t))_{ij}\partial_i V(x,t)\partial_j V(x,t)\Bigg)\Bigg) \tag{C.1b}$$

$$= -e^{-V(x,t)}\Bigg(b(x,t)\cdot\nabla V(x,t) + \frac{1}{2}\Bigg(\sum_{i,j=1}^{d}(\sigma\sigma^\top(x,t))_{ij}\partial_i\partial_j V(x,t) - \sum_{i,j,k=1}^{d}\sigma_{ik}\sigma_{jk}(x,t)\partial_i V(x,t)\partial_j V(x,t)\Bigg)\Bigg) \tag{C.1c}$$

$$= -e^{-V(x,t)}\left(b(x,t)\cdot\nabla V(x,t) + \frac{1}{2}\left(\sum_{i,j=1}^{d}(\sigma\sigma^\top(x,t))_{ij}\partial_i\partial_j V(x,t) - \sum_{k=1}^{d}\left(\sum_{i=1}^{d}\sigma_{ik}(x,t)\partial_i V(x,t)\right)^2\right)\right) \tag{C.1d}$$

$$= -e^{-V(x,t)}\left(b(x,t)\cdot\nabla V(x,t) + \frac{1}{2}(\sigma\sigma^\top)(x,t) : \nabla^2 V(x,t) - \frac{1}{2}\left|\sigma^\top\nabla V(x,t)\right|^2\right). \tag{C.1e}$$

The Feynman-Kac PDE (2.44) therefore becomes

$$0 = (\partial_t + L - f(x,t))e^{-V(x,t)} = -e^{-V(x,t)}\left((\partial_t + L)V(x,t) + f(x,t) - \frac{1}{2}\left|\sigma^\top\nabla V(x,t)\right|^2\right), \tag{C.2}$$

which is equivalent to the HJB PDE in (2.29). □

Proof of Theorem 2.33. The proof is based on the Feynman-Kac formula and Itô's Lemma, where we refer to [98, Sec. VI.5] for all technical details regarding the regularity of solutions of PDEs. By the Feynman-Kac formula from Theorem 2.14, the function ψ solves the parabolic boundary value problem

$$(\partial_t + L - f)\,\psi(x,t) = 0 \qquad (x,t)\in\mathbb{R}^d\times[0,T), \tag{C.3a}$$

$$\psi(x,T) = e^{-g(x)} \qquad x\in\mathbb{R}^d. \tag{C.3b}$$

Now let us define the process

$$\zeta_s^u = -\log\psi(X_s^u, s), \tag{C.4}$$

with X_s^u given by (2.105). Then, using Itô's Lemma and introducing the shorthands

$$\psi_s^u = \psi(X_s^u, s), \qquad b_s^u = b(X_s^u, s), \qquad \sigma_s^u = \sigma(X_s^u, s), \tag{C.5}$$

we see that $(\zeta_s^u)_{0\le s<T}$ satisfies the SDE

$$\begin{aligned}
\mathrm{d}\zeta_s^u =& -\partial_s \log\psi_s^u \mathrm{d}s - \nabla\log\psi_s^u\cdot(b_s^u+\sigma_s^u u_s)\,\mathrm{d}s \\
&-\frac{1}{2}\sigma_s^u(\sigma_s^u)^\top : \nabla^2(\log\psi_s^u)\mathrm{d}s - \left((\sigma_s^u)^\top\nabla\log\psi_s^u\right)\cdot \mathrm{d}W_s^u \qquad \text{(C.6a)}\\
=& -\left(\frac{(\partial_t+L)\psi_s^u}{\psi_s^u} + \left((\sigma_s^u)^\top\frac{\nabla\psi_s^u}{\psi_s^u}\right)\cdot u_s - \frac{1}{2}\frac{|(\sigma_s^u)^\top\nabla\psi_s^u|^2}{(\psi_s^u)^2}\right)\mathrm{d}s - \left((\sigma_s^u)^\top\frac{\nabla\psi_s^u}{\psi_s^u}\right)\cdot\mathrm{d}W_s^u \qquad \text{(C.6b)}\\
=& -\left(f(X_s^u,s) + \left((\sigma_s^u)^\top\frac{\nabla\psi_s^u}{\psi_s^u}\right)\cdot u_s - \frac{1}{2}\frac{|(\sigma_s^u)^\top\nabla\psi_s^u|^2}{(\psi_s^u)^2}\right)\mathrm{d}s - \left((\sigma_s^u)^\top\frac{\nabla\psi_s^u}{\psi_s^u}\right)\cdot\mathrm{d}W_s^u. \qquad \text{(C.6c)}
\end{aligned}$$

In the last equation, we have used the Feynman-Kac PDE (C.3a). Now, choosing $u_s = u_s^*$ for $0\le s\le T$ to be the optimal control

$$u_s^* = \sigma(X_s^{u^*},s)^\top\nabla\log\psi(X_s^{u^*},s), \tag{C.7}$$

the last equation can be recast as

$$\mathrm{d}\zeta_s^{u^*} = -\left(f(X_s^{u^*},s)+\frac{1}{2}|u_s^*|^2\right)\mathrm{d}s - u_s^*\cdot\mathrm{d}W_s^{u^*}. \tag{C.8}$$

If we introduce

$$Z_{s,T}^{u^*} = \int_s^T u_r^*\cdot\mathrm{d}W_r^{u^*} + \frac{1}{2}\int_s^T |u_r^*|^2\mathrm{d}r, \tag{C.9}$$

we have

$$\mathrm{d}\zeta_s^{u^*} = -f(X_s^{u^*},s)\mathrm{d}s - \mathrm{d}Z_{s,T}^{u^*}. \tag{C.10}$$

As a consequence, using the continuity of the process as $s\downarrow 0$,

$$\zeta_T^{u^*} = \zeta_0^{u^*} - Z_{0,T}^{u^*} - \int_0^T f(X_s^{u^*},s)\mathrm{d}s. \tag{C.11}$$

By definition of ζ_s^u, the initial value $\zeta_0^{u^*} = -\log\psi(X_0^{u^*},0) = -\log\psi(x,0)$ is deterministic. Moreover $\zeta_T^{u^*} = -\log\psi(X_T^{u^*},T) = g(X_T^{u^*})$, which in combination with (C.11) yields

$$-\log\psi(x,0) = g(X_T^{u^*}) + \int_0^T f(X_s^{u^*},s)\mathrm{d}s - Z_{0,T}^{u^*}. \tag{C.12}$$

Rearranging the terms in the last equation, we find

$$\psi(x,0) = \exp\left(-Z_{0,T}^{u^*} - \int_0^T f(X_s^{u^*},s)\mathrm{d}s - g(X_T^{u^*})\right), \tag{C.13}$$

with probability one, which yields the assertion of the statement. □

C.2 Proofs for Chapter 3

Proof of Proposition 3.8. We adapt a proof of [209]. Assume first that $m\ge 1$, then for any $E\in\mathcal{F}$

$$\nu(E)-\lambda(E)\ge\nu(E)-m\lambda(E)\ge 0, \tag{C.14}$$

where the last inequality follows from the definition of m. On the other hand, if $E=\widetilde{\Omega}$, then $\nu(E)-\lambda(E)=0$ and therefore it follows that $m=1$, i.e. $\nu=\lambda$.

Let now $m<1$. We want to show $m\mathcal{J}(f,\lambda,\varphi)\le\mathcal{J}(f,\nu,\varphi)$, which is equivalent to

$$\mathbb{E}_\nu[f(\varphi)] - m\,\mathbb{E}_\lambda[f(\varphi)] + mf\left(\mathbb{E}_\lambda[\varphi]\right)\ge f\left(\mathbb{E}_\nu[\varphi]\right). \tag{C.15}$$

We compute

$$\mathbb{E}_\nu[f(\varphi)] - m\,\mathbb{E}_\lambda[f(\varphi)] + mf\left(\mathbb{E}_\lambda[\varphi]\right) \geq \left(\mathbb{E}_\nu[1] - m\,\mathbb{E}_\lambda[1]\right) f\left(\frac{\mathbb{E}_\nu[\varphi] - m\,\mathbb{E}_\lambda[\varphi]}{\mathbb{E}_\nu[1] - m\,\mathbb{E}_\lambda[1]}\right) + mf\left(\mathbb{E}_\lambda[\varphi]\right) \tag{C.16a}$$

$$= (1-m) f\left(\frac{\mathbb{E}_\nu[\varphi] - m\,\mathbb{E}_\lambda[\varphi]}{1-m}\right) + mf\left(\mathbb{E}_\lambda[\varphi]\right) \tag{C.16b}$$

$$\geq f\left(\mathbb{E}_\nu[\varphi] - m\,\mathbb{E}_\lambda[\varphi] + m\,\mathbb{E}_\lambda[\varphi]\right) \tag{C.16c}$$

$$= f\left(\mathbb{E}_\nu[\varphi]\right), \tag{C.16d}$$

where we used two times the convexity of f. The other inequality follows analogously. □

Proof of Proposition 3.15. We compute

$$\mathbb{E}\left[e^{-2\mathcal{W}(X^u)}\left(\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^u}(X^u)\right)^2\right] = \mathbb{E}\left[e^{-2\mathcal{W}(X^u)}\left(\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^{u^*}}(X^u)\frac{\mathrm{d}\mathbb{P}^{u^*}}{\mathrm{d}\mathbb{P}^u}(X^u)\right)^2\right] \tag{C.17a}$$

$$= \mathcal{Z}^2\,\mathbb{E}\left[\left(\frac{\mathrm{d}\mathbb{P}^{u^*}}{\mathrm{d}\mathbb{P}^u}(X^u)\right)^2\right], \tag{C.17b}$$

where we used

$$\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^{u^*}}(X^u) = e^{\mathcal{W}(X^u)}\mathcal{Z}. \tag{C.18}$$

Equation (3.35) now follows by the Girsanov formula (see Theorem B.3) and the definition of the variance. For equation (3.36) note that we can write

$$\mathbb{E}_{\mathbb{P}^u}\left[\left(\frac{\mathrm{d}\mathbb{P}^{u^*}}{\mathrm{d}\mathbb{P}^u}\right)^2\right] = \mathbb{E}_{\mathbb{P}^{u+2\delta}}\left[\left(\frac{\mathrm{d}\mathbb{P}^{u^*}}{\mathrm{d}\mathbb{P}^u}\right)^2\frac{\mathrm{d}\mathbb{P}^u}{\mathrm{d}\mathbb{P}^{u+2\delta}}\right]. \tag{C.19}$$

We compute

$$\frac{\mathrm{d}\mathbb{P}^{u^*}}{\mathrm{d}\mathbb{P}^u}(X^{u+2\delta}) = \exp\left(\frac{3}{2}\int_0^T |\delta(X_s^{u+2\delta},s)|^2\mathrm{d}s + \int_0^T \delta(X_s^{u+2\delta},s)\cdot\mathrm{d}W_s\right) \tag{C.20}$$

and

$$\frac{\mathrm{d}\mathbb{P}^u}{\mathrm{d}\mathbb{P}^{u+2\delta}}(X^{u+2\delta}) = \exp\left(-2\int_0^T |\delta(X_s^{u+2\delta},s)|^2\mathrm{d}s - 2\int_0^T \delta(X_s^{u+2\delta},s)\cdot\mathrm{d}W_s\right), \tag{C.21}$$

from which the desired formula immediately follows. □

Alternative proof of Corollary 3.18. We follow the reasoning in [170, Thm. 2.1] and apply Grönwall's inequality to the square integrable exponential martingale Z.[52] To this end, we define the shorthands $\delta(x,t) := (u^* - u)(x,t)$ and

$$Z_t := \exp\left(-\frac{1}{2}\int_0^t |\delta(X_s,s)|^2\mathrm{d}s + \int_0^t \delta(X_s,s)\cdot\mathrm{d}W_s\right). \tag{C.22}$$

Then, by Itô's formula,

$$Z_t^2 = 1 + 2\int_0^t Z_s\mathrm{d}Z_s + \int_0^t Z_s^2|\delta(X_s,s)|^2\mathrm{d}s, \tag{C.23}$$

and therefore, after taking expectations,

$$\mathbb{E}\left[Z_t^2\right] = 1 + \mathbb{E}\left[\int_0^t Z_s^2|\delta(X_s,s)|^2\mathrm{d}s\right] \tag{C.24a}$$

$$\leq 1 + \int_0^t \mathbb{E}\left[Z_s^2\right] h_2^2(s)\mathrm{d}s. \tag{C.24b}$$

[52] See also Theorem 2 in `http://math.ucsd.edu/~pfitz/downloads/courses/spring05/math280c/expmart.pdf`.

We can now apply Grönwall's inequality to get

$$\mathbb{E}\left[Z_t^2\right] \leq \exp\left(\int_0^t h_2^2(s)\mathrm{d}s\right) \tag{C.25}$$

and therefore the desired statement after applying Proposition 3.15. The other direction follows analogously by noting that

$$-\mathbb{E}\left[Z_t^2\right] \leq -1 - \int_0^t \mathbb{E}\left[Z_s^2\right] h_1^2(s)\mathrm{d}s. \tag{C.26}$$

□

Remark C.1. Yet another alternative to prove Corollary 3.18 is by computing

$$\mathbb{E}\left[\left(\frac{\mathrm{d}\mathbb{P}^{u^*}}{\mathrm{d}\mathbb{P}^u}(X^u)\right)^2\right] = \mathbb{E}\left[\exp\left(-\int_0^T |\delta(X_s^u,s)|^2\mathrm{d}s + 2\int_0^T \delta(X_s^u,s)\cdot \mathrm{d}W_s\right)\right] \tag{C.27a}$$

$$= \mathbb{E}\left[\exp\left(\int_0^T |\delta(X_s^u,s)|^2\mathrm{d}s - 2\int_0^T |\delta(X_s^u,s)|^2\mathrm{d}s + 2\int_0^T \delta(X_s^u,s)\cdot \mathrm{d}W_s\right)\right] \tag{C.27b}$$

$$\leq \exp\left(\int_0^T h_2^2(s)\mathrm{d}s\right)\ \mathbb{E}\left[\exp\left(-\frac{1}{2}\int_0^T |2\delta(X_s^u,s)|^2\mathrm{d}s + \int_0^T 2\delta(X_s^u,s)\cdot \mathrm{d}W_s\right)\right] \tag{C.27c}$$

$$= \exp\left(\int_0^T h_2^2(s)\mathrm{d}s\right), \tag{C.27d}$$

where we used the constant expectation property of the exponential martingale in the last step. The other direction follows analogously.

Proof of Proposition 3.20. From Lemma B.5 it holds for $n, p, q > 1$ with $\frac{1}{p} + \frac{1}{q} = 1$ that

$$\mathbb{E}\left[\left(\frac{\mathrm{d}\mathbb{P}^{u^*}}{\mathrm{d}\mathbb{P}^u}(X^u)\right)^n\right] \leq \mathbb{E}\left[\exp\left(\frac{nq(np-1)}{2}\int_0^T |u^* - u|^2(X_s^u,s)\mathrm{d}s\right)\right]^{\frac{1}{q}}. \tag{C.28}$$

We write $q = \frac{p}{p-1}$ and note that $q(np-1) = \frac{p(np-1)}{p-1}$ is minimized by $p^* = 1 \pm \sqrt{1-\frac{1}{n}}$, from which we are only allowed to take the positive part due to the constraint $p \geq 1$. For $n = 2$ this yields $p^* = \frac{\sqrt{2}+1}{\sqrt{2}}$ and $q^* = \sqrt{2}+1$, and we get the desired statement by recalling

$$r^2(u) = \mathrm{Var}_{\mathbb{P}^u}\left(\frac{\mathrm{d}\mathbb{P}^{u^*}}{\mathrm{d}\mathbb{P}^u}\right) = \mathbb{E}_{\mathbb{P}^u}\left[\left(\frac{\mathrm{d}\mathbb{P}^{u^*}}{\mathrm{d}\mathbb{P}^u}\right)^2\right] - 1. \tag{C.29}$$

□

C.3 Proofs for Chapter 4

Proof of Proposition 4.7. Using (1.13) and (B.7) (or arguing as in the proof of Theorem 1.2) we compute

$$\mathcal{L}_{\mathrm{RE}}(u) = \mathbb{E}_{\mathbb{P}^u}\left[\log\frac{\mathrm{d}\mathbb{P}^u}{\mathrm{d}\mathbb{Q}}\right] = \mathbb{E}_{\mathbb{P}^u}\left[\log\left(\frac{\mathrm{d}\mathbb{P}^u}{\mathrm{d}\mathbb{P}}\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}}\right)\right] \tag{C.30a}$$

$$= \mathbb{E}\left[\int_0^T u(X_s^u,s)\cdot\ \mathrm{d}W_s + \frac{1}{2}\int_0^T |u(X_s^u,s)|^2\ \mathrm{d}s + \int_0^T f(X_s^u,s)\mathrm{d}s + g(X_T^u)\right] + \log\mathcal{Z} \tag{C.30b}$$

$$= \mathbb{E}\left[\frac{1}{2}\int_0^T |u(X_s^u,s)|^2\ \mathrm{d}s + \int_0^T f(X_s^u,s)\mathrm{d}s + g(X_T^u)\right] + \log\mathcal{Z}. \tag{C.30c}$$

□

Proof of Proposition 4.9. Similarly, we compute

$$\mathcal{L}_{\mathrm{CE}}(u) = \mathbb{E}_{\mathbb{Q}}\left[\log \frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}^u}\right] = \mathbb{E}_{\mathbb{P}^v}\left[\log\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^u}\right)\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^v}\right] \tag{C.31a}$$

$$= \mathbb{E}\Bigg[\left(\frac{1}{2}\int_0^T |u(X_s^v,s)|^2\,\mathrm{d}s - \int_0^T (u\cdot v)(X^v,s)\,\mathrm{d}s - \int_0^T u(X_s^v,s)\cdot \mathrm{d}W_s - \mathcal{W}(X^v) - \log\mathcal{Z}\right) \frac{1}{\mathcal{Z}}\exp\left(-\mathcal{W}(X^v) - \int_0^T v(X_s^v,s)\cdot \mathrm{d}W_s - \frac{1}{2}\int_0^T |v(X_s^v,s)|^2\,\mathrm{d}s\right)\Bigg] \tag{C.31b}$$

$$= \frac{1}{\mathcal{Z}}\mathbb{E}\Bigg[\left(\frac{1}{2}\int_0^T |u(X_s^v,s)|^2\,\mathrm{d}s - \int_0^T (u\cdot v)(X_s^v,s)\mathrm{d}s - \int_0^T u(X_s^v,s)\cdot \mathrm{d}W_s\right) \exp\left(-\int_0^T v(X_s^v,s)\cdot \mathrm{d}W_s - \frac{1}{2}\int_0^T |v(X_s^v,s)|^2\,\mathrm{d}s - \mathcal{W}(X^v)\right)\Bigg] + C, \tag{C.31c}$$

where $C \in \mathbb{R}$ does not depend on u. □

Proof of Proposition 4.19. For $\varepsilon \in \mathbb{R}$ and $\phi \in C_b^1(\mathbb{R}^d \times [0,T], \mathbb{R}^d)$, let us define the change of measure

$$\Xi_T(\varepsilon,\phi) = \exp\left(-\varepsilon\int_0^T \phi(X_s^u,s)\cdot \mathrm{d}W_s - \frac{\varepsilon^2}{2}\int_0^T |\phi(X_s^u,s)|^2\,\mathrm{d}s\right), \qquad \frac{\mathrm{d}\widetilde{\Lambda}}{\mathrm{d}\Lambda} = \Xi_T(\varepsilon,\phi). \tag{C.32}$$

According to Girsanov's theorem, the process $(\widetilde{W}_s)_{0\le s\le T}$, defined as

$$\widetilde{W}_t = W_t + \varepsilon\int_0^t \phi(X_s^u,s)\,\mathrm{d}s, \tag{C.33}$$

is a Brownian motion under $\widetilde{\Lambda}$. We therefore obtain

$$\mathcal{L}_{\mathrm{RE}}(u+\varepsilon\phi) = \mathbb{E}\left[\left(\frac{1}{2}\int_0^T |(u+\varepsilon\phi)(X_s^u,s)|^2\,\mathrm{d}s + \int_0^T f(X_s^u,s)\,\mathrm{d}s + g(X_T^u)\right)\Xi_T^{-1}(\varepsilon,\phi)\right] + \log\mathcal{Z}. \tag{C.34}$$

Using dominated convergence, we can interchange derivatives and integrals (for technical details, we refer to [195]) and compute

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\mathcal{L}_{\mathrm{RE}}(u+\varepsilon\phi) = \mathbb{E}\Bigg[\int_0^T (u\cdot\phi)(X_s^u,s)\,\mathrm{d}s + \left(\frac{1}{2}\int_0^T |u(X_s^u,s)|^2\,\mathrm{d}s + \int_0^T f(X_s^u,s)\,\mathrm{d}s + g(X_T^u)\right)\int_0^T \phi(X_s^u,s)\cdot \mathrm{d}W_s\Bigg] \tag{C.35a}$$

$$= \mathbb{E}\left[\left(g(X_T^u) - \widetilde{Y}_T^{u,u}\right)\int_0^T \phi(X_s^u,s)\cdot \mathrm{d}W_s\right], \tag{C.35b}$$

where we have used Itô's isometry,

$$\mathbb{E}\left[\int_0^T \phi(X_s^u,s)\cdot \mathrm{d}W_s \int_0^T u(X_s^u,s)\cdot \mathrm{d}W_s\right] = \mathbb{E}\left[\int_0^T (u\cdot\phi)(X_s^u,s)\,\mathrm{d}s\right]. \tag{C.36}$$

Turning to the log-variance loss, we see that

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\mathcal{L}_{\mathrm{Var}_v}^{\log}(u+\varepsilon\phi) = \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\left(\mathbb{E}\left[\left(\widetilde{Y}_T^{u+\varepsilon\phi,v} - g(X_T^v)\right)^2\right] - \mathbb{E}\left[\left(\widetilde{Y}_T^{u+\varepsilon\phi,v} - g(X_T^v)\right)\right]^2\right) \tag{C.37a}$$

$$= 2\,\mathbb{E}\left[\left(\widetilde{Y}_T^{u,v} - g(X_T^v)\right)\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\widetilde{Y}_T^{u+\varepsilon\phi,v}\right] - 2\,\mathbb{E}\left[\left(\widetilde{Y}_T^{u,v} - g(X_T^v)\right)\right]\mathbb{E}\left[\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\widetilde{Y}_T^{u+\varepsilon\phi,v}\right], \tag{C.37b}$$

where

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\widetilde{Y}_T^{u+\varepsilon\phi,v} = \int_0^T (\phi\cdot(u-v))(X_s^v,s)\,\mathrm{d}s - \int_0^T \phi(X_s^v,s)\cdot\mathrm{d}W_s. \tag{C.38}$$

Setting $v=u$, we obtain

$$\left(\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\mathcal{L}^{\log}_{\mathrm{Var}_v}(u+\varepsilon\phi)\right)\Big|_{v=u} = 2\,\mathbb{E}\left[\left(g(X_T^u)-\widetilde{Y}_T^{u,u}\right)\int_0^T \phi(X_s^u,s)\cdot\mathrm{d}W_s\right], \tag{C.39}$$

from which the result follows by comparison with (C.35). □

Proof of Proposition 4.22. We compute

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\mathcal{L}_{\mathrm{moment}_v}(u+\varepsilon\phi) = 2\,\mathbb{E}\left[\left(\widetilde{Y}_T^{u,v}+y_0-g(X_T^v)\right)\left(\int_0^T(\phi\cdot(u-v))(X_s^v,s)\,\mathrm{d}s-\int_0^T\phi(X_s^v,s)\cdot\mathrm{d}W_s\right)\right]. \tag{C.40}$$

Setting $v=u$ and using that $\mathbb{E}\left[y_0\int_0^T\phi(X_s^v,s)\cdot\mathrm{d}W_s\right]=0$, the first statement follows by comparison with (4.39). The second statement follows from

$$\left(\frac{\delta}{\delta u}\mathcal{L}_{\mathrm{moment}_v}(u,y_0;\phi)\right)\Big|_{u=u^*} = 2\,\mathbb{E}\left[(y_0+\log\mathcal{Z})\left(\int_0^T(\phi\cdot(u^*-v))(X_s^v,s)\,\mathrm{d}s\right)\right], \tag{C.41}$$

where we have used the fact that $\widetilde{Y}_T^{u^*,v}-g(X_T^v)=\log\mathcal{Z}$, almost surely. □

Proof of Proposition 4.25. 1.) We compute

$$\frac{\delta}{\delta u}\Big|_{u=u^*}\widehat{\mathcal{L}}^{(K)}_{\mathrm{Var}_v}(u;\phi) = 2\left(\frac{1}{K}\sum_{k=1}^K\left[\exp\left(2\left(\widetilde{Y}_T^{u^*,v,(k)}-g\left(X_T^{v,(k)}\right)\right)\right)\frac{\delta\widetilde{Y}_T^{u,v,(k)}}{\delta u}(u^*;\phi)\right]\right. \tag{C.42a}$$

$$\left.-\frac{1}{K}\sum_{k=1}^K\left[\exp\left(\widetilde{Y}_T^{u^*,v,(k)}-g\left(X_T^{v,(k)}\right)\right)\frac{\delta\widetilde{Y}_T^{u,v,(k)}}{\delta u}(u^*;\phi)\right]\frac{1}{K}\sum_{k=1}^K\left[\exp\left(\widetilde{Y}_T^{u^*,v,(k)}-g\left(X_T^{v,(k)}\right)\right)\right]\right), \tag{C.42b}$$

where $\frac{\delta\widetilde{Y}_T^{u,v,(k)}}{\delta u}(u;\phi)$ is given in (4.54). As in the proof for the log-variance estimator, the quantity

$$\exp\left(\widetilde{Y}_T^{u^*,v,(k)}-g\left(X_T^{v,(k)}\right)\right) \tag{C.43}$$

is almost surely constant and thus the statement follows.

2.) Similarly to the computations involved in 1.) we have

$$\frac{\delta}{\delta u}\Big|_{u=u^*}\widehat{\mathcal{L}}^{(K)}_{\mathrm{moment}_v}(u,y_0;\phi) = \frac{2}{K}\sum_{k=1}^K\left(\widetilde{Y}_T^{u^*,v,(k)}+y_0-g\left(X_T^{u^*,(k)}\right)\right)\frac{\delta\widetilde{Y}_T^{u,v,(k)}}{\delta u}(u^*;\phi) \tag{C.44a}$$

$$=\frac{2}{K}\left(-\log\mathcal{Z}+y_0\right)\sum_{k=1}^K\left(\int_0^T\phi(X_s^{v,(k)},s)\cdot\mathrm{d}W_s^{(k)}-\int_0^T(\phi\cdot(u^*-v))\left(X_s^{v,(k)},s\right)\mathrm{d}s\right), \tag{C.44b}$$

where we have used the fact that $\widetilde{Y}_T^{u^*,v,(k)}-g\left(X_T^{u^*,(k)}\right)=-\log\mathcal{Z}$ according to (1.25) and (4.25b). The variance of this expression equals

$$\frac{4}{K}\left(\log\mathcal{Z}-y_0\right)^2\mathbb{E}\left[\left(\int_0^T\phi(X_s^{v,(k)},s)\cdot\mathrm{d}W_s^{(k)}-\int_0^T(\phi\cdot(u^*-v))\left(X_s^{v,(k)},s\right)\mathrm{d}s\right)^2\right], \tag{C.45}$$

implying the claim.

3.) Let $\phi\in C_b^1(\mathbb{R}^d\times[0,T],\mathbb{R}^d)$ and $\varepsilon\in\mathbb{R}$. As usual, we denote by $(X_s^{u^*+\varepsilon\phi})_{0\le s\le T}$ the unique strong solution to (1.4), with u replaced by $u^*+\varepsilon\phi$. By a slight modification of [181, Theorems 3.1 and 3.3] detailed, for instance, in

[225, Section 10.2.2], $X_s^{u^*+\varepsilon\phi}$ is almost surely differentiable as a function of ε. Furthermore, $\frac{\mathrm{d}X_s^{u^*+\varepsilon\phi}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0} =: A_s$ satisfies the SDE (4.52). We calculate

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\left[\frac{1}{2}\int_0^T |u^*+\varepsilon\phi|^2(X_s^{u^*+\varepsilon\phi},s)\,\mathrm{d}s + \int_0^T f(X_s^{u^*+\varepsilon\phi},s)\,\mathrm{d}s + g(X_T^{u^*+\varepsilon\phi})\right] \tag{C.46a}$$

$$= \int_0^T (u^*\cdot\phi)(X_s^{u^*},s)\,\mathrm{d}s + \frac{1}{2}\int_0^T (\nabla|u^*|^2)(X_s^{u^*},s)\cdot A_s\,\mathrm{d}s + \int_0^T \nabla f(X_s^{u^*},s)\cdot A_s\,\mathrm{d}s + \nabla g(X_T^{u^*})\cdot A_T. \tag{C.46b}$$

From (1.20b) and using integration by parts, we see that the last term in (C.46b) satisfies

$$(\nabla g)(X_T^{u^*})\cdot A_T = \nabla V(X_T^{u^*},T)\cdot A_T = \int_0^T \nabla V(X_s^{u^*},s)\cdot \mathrm{d}A_s + \int_0^T A_s\cdot \mathrm{d}(\nabla V(X_s^{u^*},s)) + \Big\langle A_\cdot, \nabla V(X_\cdot^{u^*},\cdot)\Big\rangle_T. \tag{C.47}$$

Next, we employ Itô's formula and Einstein's summation convention to compute

$$\mathrm{d}(\partial_{x_i}V(X_s^{u^*},s)) = \tag{C.48a}$$

$$= \left[\partial_{x_i}\partial_s V + (\partial_{x_i}\partial_{x_j}V)(b+\sigma u^*)_j + \frac{1}{2}(\partial_{x_i}\partial_{x_j}\partial_{x_k}V)\sigma_{jl}\sigma_{kl}\right](X_s^{u^*},s)\,\mathrm{d}s + \left[(\partial_{x_i}\partial_{x_j}V)\sigma_{jk}\right](X_s^{u^*},s)\,\mathrm{d}W_s^k \tag{C.48b}$$

$$= \partial_{x_i}\left[\partial_s V + LV - \frac{1}{2}(\partial_{x_j}V)\sigma_{jk}\sigma_{lk}(\partial_{x_l}V)\right](X_s^{u^*},s)\,\mathrm{d}s + \left[(\partial_{x_i}\partial_{x_j}V)\sigma_{jk}\right](X_s^{u^*},s)\,\mathrm{d}W_s^k$$
$$+ \left[\frac{1}{2}\left((\partial_{x_j}V)(\partial_{x_l}V) - \partial_{x_j}\partial_{x_l}V\right)\partial_{x_i}(\sigma_{jk}\sigma_{lk}) - (\partial_{x_j}V)\partial_{x_i}b_j\right](X_s^{u^*},s)\,\mathrm{d}s \tag{C.48c}$$

$$= \left[\frac{1}{2}\left((\partial_{x_j}V)(\partial_{x_l}V) - \partial_{x_j}\partial_{x_l}V\right)\partial_{x_i}(\sigma_{jk}\sigma_{lk}) - (\partial_{x_j}V)\partial_{x_i}b_j - \partial_{x_i}f\right](X_s^{u^*},s)\,\mathrm{d}s$$
$$+ \left[(\partial_{x_i}\partial_{x_j}V)\sigma_{jk}\right](X_s^{u^*},s)\,\mathrm{d}W_s^k, \tag{C.48d}$$

where we used (4.5) from the second to the third line and (1.20) to manipulate the first term in the third line. Using (4.52) and (C.48), we see that the quadratic variation process satisfies

$$\Big\langle A_\cdot, \nabla V(X_\cdot^{u^*},\cdot)\Big\rangle_T = \frac{1}{2}\int_0^T A_j\left[\partial_{x_j}(\sigma_{ik}\sigma_{lk})(\partial_{x_i}\partial_{x_l}V)\right](X_s^{u^*},s)\,\mathrm{d}s. \tag{C.49}$$

Combining (4.52), (C.47), (C.48) and (C.49), it follows that (C.46) equals

$$\int_0^T \left[A_j(\partial_{x_i}V)\partial_{x_j}\sigma_{ik} + A_j(\partial_{x_i}\partial_{x_j}V)\sigma_{ik}\right](X_s^{u^*},s)\,\mathrm{d}W_s^k = -\int_0^T A_s\cdot(\nabla u^*)(X_s^{u^*},s)\,\mathrm{d}W_s. \tag{C.50}$$

The claim is now implied by Itô's isometry.

4.) With the definition of the cross-entropy loss estimator as in (4.32) we compute

$$\frac{\delta}{\delta u}\Big|_{u=u^*}\widehat{\mathcal{L}}_{\mathrm{CE},v}(u;\phi) = \frac{1}{K}\sum_{k=1}^K\left[\left(\int_0^T (\phi\cdot(u^*-v))(X_s^{v,(k)},s)\,\mathrm{d}s - \int_0^T \phi(X_s^{v,(k)},s)\cdot\mathrm{d}W_s^{(k)}\right)\right.$$
$$\left.\exp\left(-\int_0^T v(X_s^{v,(k)},s)\cdot\mathrm{d}W_s^{(k)} - \frac{1}{2}\int_0^T |v(X_s^{v,(k)},s)|^2\,\mathrm{d}s - \mathcal{W}(X^{v,(k)})\right)\right]. \tag{C.51}$$

Since $\mathbb{E}\left[\frac{\delta}{\delta u}\Big|_{u=u^*}\widehat{\mathcal{L}}_{\mathrm{CE},v}(u;\phi)\right] = 0$ by construction, we see that

$$\operatorname{Var}\left(\frac{\delta}{\delta u}\Big|_{u=u^*}\widehat{\mathcal{L}}_{\mathrm{CE},v}(u;\phi)\right) = \frac{1}{K}\,\mathbb{E}\left[\left(\int_0^T(\phi\cdot(u^*-v))(X_s^v,s)\,\mathrm{d}s - \int_0^T\phi(X_s^v,s)\cdot\mathrm{d}W_s\right)^2 \right.$$
$$\left. \exp\left(-2\int_0^T v(X_s^v,s)\cdot\mathrm{d}W_s - \int_0^T|v(X_s^v,s)|^2\,\mathrm{d}s - 2\,\mathcal{W}(X^v)\right)\right]. \tag{C.52}$$

Let us assume for the sake of contradiction that $\operatorname{Var}\left(\frac{\delta}{\delta u}\Big|_{u=u^*}\widehat{\mathcal{L}}_{\mathrm{CE},v}(u;\phi)\right) = 0$, for all $\phi \in C_b^1(\mathbb{R}^d\times[0,T],\mathbb{R}^d)$. It then follows that

$$\int_0^T(\phi\cdot(u^*-v))(X_s^v,s)\,\mathrm{d}s = \int_0^T\phi(X_s^v,s)\cdot\mathrm{d}W_s, \tag{C.53}$$

which is clearly false, in general. □

Proof of Proposition 4.29. Throughout the proof, we will use the notation

$$\mathbb{P}^M := \bigotimes_{i=1}^M\mathbb{P}_i, \qquad \mathbb{Q}^M := \bigotimes_{i=1}^M\mathbb{Q}_i, \qquad \widetilde{\mathbb{P}}^M = \bigotimes_{i=1}^M\widetilde{\mathbb{P}}_i \tag{C.54}$$

to denote the product measures on $\bigotimes_{i=1}^M C([0,T],\mathbb{R}^d)\simeq C([0,T],\mathbb{R}^{Md})$ associated to $\mathbb{P}$, $\mathbb{Q}$ and $\widetilde{\mathbb{P}}$, where $\mathbb{P}_i$, $\mathbb{Q}_i$ and $\widetilde{\mathbb{P}}_i$ refer to identical copies.

1.) First note that

$$D_{\widetilde{\mathbb{P}}^M}^{\mathrm{Var(log)}}(\mathbb{P}^M|\mathbb{Q}^M) = \operatorname{Var}_{\widetilde{\mathbb{P}}^M}\left(\sum_{i=1}^M\log\left(\frac{\mathrm{d}\mathbb{Q}_i}{\mathrm{d}\mathbb{P}_i}\right)\right) = \sum_{i=1}^M\operatorname{Var}_{\widetilde{\mathbb{P}}_i}\left(\log\left(\frac{\mathrm{d}\mathbb{Q}_i}{\mathrm{d}\mathbb{P}_i}\right)\right) = MD_{\widetilde{\mathbb{P}}}^{\mathrm{Var(log)}}(\mathbb{P}|\mathbb{Q}). \tag{C.55}$$

The sample variance satisfies [56]

$$\operatorname{Var}\left(\widehat{D}_{\widetilde{\mathbb{P}}^M}^{\mathrm{Var(log)},(K)}(\mathbb{P}^M|\mathbb{Q}^M)\right) = \frac{1}{K}\left(\mu_4 - \frac{K-3}{K-1}D_{\widetilde{\mathbb{P}}^M}^{\mathrm{Var(log)}}(\mathbb{P}^M|\mathbb{Q}^M)^2\right), \tag{C.56}$$

where

$$\mu_4 = \mathbb{E}_{\widetilde{\mathbb{P}}^M}\left[\left(\log\left(\frac{\mathrm{d}\mathbb{Q}^M}{\mathrm{d}\mathbb{P}^M}\right) - \mathbb{E}_{\widetilde{\mathbb{P}}^M}\left[\log\left(\frac{\mathrm{d}\mathbb{Q}^M}{\mathrm{d}\mathbb{P}^M}\right)\right]\right)^4\right]. \tag{C.57}$$

We calculate

$$\mu_4 = \mathbb{E}_{\widetilde{\mathbb{P}}^M}\left[\left(\sum_{i=1}^M\left(\log\left(\frac{\mathrm{d}\mathbb{Q}_i}{\mathrm{d}\mathbb{P}_i}\right) - \mathbb{E}_{\widetilde{\mathbb{P}}_i}\left[\log\left(\frac{\mathrm{d}\mathbb{Q}_i}{\mathrm{d}\mathbb{P}_i}\right)\right]\right)\right)^4\right] \tag{C.58a}$$

$$= M\mathbb{E}_{\mathbb{P}}\left[\left(\log\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\right) - \mathbb{E}_{\mathbb{P}}\left[\log\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\right)\right]\right)^4\right] + 6\binom{M}{2}\mathbb{E}_{\mathbb{P}}\left[\left(\log\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\right) - \mathbb{E}_{\mathbb{P}}\left[\log\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\right)\right]\right)^2\right]^2, \tag{C.58b}$$

where we have used the fact that, for instance,

$$\mathbb{E}_{\widetilde{\mathbb{P}}^M}\left[\left(\log\left(\frac{\mathrm{d}\mathbb{Q}_i}{\mathrm{d}\mathbb{P}_i}\right) - \mathbb{E}_{\widetilde{\mathbb{P}}_i}\left[\log\left(\frac{\mathrm{d}\mathbb{Q}_i}{\mathrm{d}\mathbb{P}_i}\right)\right]\right)\left(\log\left(\frac{\mathrm{d}\mathbb{Q}_j}{\mathrm{d}\mathbb{P}_j}\right) - \mathbb{E}_{\widetilde{\mathbb{P}}_j}\left[\log\left(\frac{\mathrm{d}\mathbb{Q}_j}{\mathrm{d}\mathbb{P}_j}\right)\right]\right)^3\right] = 0, \tag{C.59}$$

for $i\neq j$. Combining this with (C.55), it follows that $\operatorname{Var}\widehat{D}_{\widetilde{\mathbb{P}}^M}^{\mathrm{Var(log)},(K)}(\mathbb{P}^M|\mathbb{Q}^M) = \mathcal{O}(M^2)$. The claim is then a consequence of the definition (4.58).

2.) We compute

$$D^{\mathrm{RE}}(\mathbb{P}^M|\mathbb{Q}^M) = \mathbb{E}_{\mathbb{P}^M}\left[\log\frac{\mathrm{d}\mathbb{P}^M}{\mathrm{d}\mathbb{Q}^M}\right] = M\,\mathbb{E}_{\mathbb{P}}\left[\log\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}}\right] = MD^{\mathrm{RE}}(\mathbb{P}|\mathbb{Q}). \tag{C.60}$$

For $\widetilde{\mathbb{P}} = \mathbb{P}$ we have

$$\mathrm{Var}\left(\widehat{D}^{\mathrm{RE},(K)}_{\mathbb{P}^M}(\mathbb{P}^M|\mathbb{Q}^M)\right) = \frac{1}{K}\mathrm{Var}_{\mathbb{P}^M}\left(\log\frac{\mathrm{d}\mathbb{P}^M}{\mathrm{d}\mathbb{Q}^M}\right) = \frac{1}{K}\mathrm{Var}_{\mathbb{P}^M}\left(\sum_{i=1}^{d}\log\frac{\mathrm{d}\mathbb{P}_i}{\mathrm{d}\mathbb{Q}_i}\right) = \frac{M^2}{K}\mathrm{Var}_{\mathbb{P}}\left(\log\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}}\right), \quad \text{(C.61)}$$

from which the robustness follows immediately. For $\widetilde{\mathbb{P}} \neq \mathbb{P}$, on the other hand,

$$\mathrm{Var}\left(\widehat{D}^{\mathrm{RE},(K)}_{\widetilde{\mathbb{P}}^M}(\mathbb{P}^M|\mathbb{Q}^M)\right) = \frac{1}{K}\mathrm{Var}_{\widetilde{\mathbb{P}}^M}\left(\log\left(\frac{\mathrm{d}\mathbb{P}^M}{\mathrm{d}\mathbb{Q}^M}\right)\frac{\mathrm{d}\mathbb{P}^M}{\mathrm{d}\widetilde{\mathbb{P}}^M}\right), \quad \text{(C.62)}$$

and the proof of the non-robustness proceeds as in 4.).

3.) As in the proof of 1.) we have

$$\mathrm{Var}\left(\widehat{D}^{\mathrm{Var},(K)}_{\widetilde{\mathbb{P}}^M}(\mathbb{P}^M|\mathbb{Q}^M)\right) = \frac{1}{K}\left(\mu_4 - \frac{K-3}{K-1}D^{\mathrm{Var}}_{\widetilde{\mathbb{P}}^M}(\mathbb{P}^M|\mathbb{Q}^M)^2\right), \quad \text{(C.63)}$$

where

$$\mu_4 = \mathbb{E}_{\widetilde{\mathbb{P}}^M}\left[\left(\frac{\mathrm{d}\mathbb{Q}^M}{\mathrm{d}\mathbb{P}^M} - \mathbb{E}_{\widetilde{\mathbb{P}}^M}\left[\frac{\mathrm{d}\mathbb{Q}^M}{\mathrm{d}\mathbb{P}^M}\right]\right)^4\right], \quad \text{(C.64)}$$

and

$$D^{\mathrm{Var}}_{\mathbb{P}^M}(\mathbb{P}^M|\mathbb{Q}^M) = \mathrm{Var}_{\widetilde{\mathbb{P}}^M}\left(\frac{\mathrm{d}\mathbb{Q}^M}{\mathrm{d}\mathbb{P}^M}\right) = \mathbb{E}_{\widetilde{\mathbb{P}}}\left[\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\right)^2\right]^M - \mathbb{E}_{\widetilde{\mathbb{P}}}\left[\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\right]^{2M}. \quad \text{(C.65)}$$

We can write the relative error as

$$r^{(K)} = \sqrt{\frac{1}{K}\left(\frac{\mu_4}{D^{\mathrm{Var}}_{\widetilde{\mathbb{P}}^M}(\mathbb{P}^M|\mathbb{Q}^M)^2} - \frac{K-3}{K-1}\right)}, \quad \text{(C.66)}$$

and estimate

$$\frac{\mu_4}{D^{\mathrm{Var}}_{\widetilde{\mathbb{P}}^M}(\mathbb{P}^M|\mathbb{Q}^M)^2} \geq \frac{\mathbb{E}_{\widetilde{\mathbb{P}}^M}\left[\left(\frac{\mathrm{d}\mathbb{Q}^M}{\mathrm{d}\mathbb{P}^M} - \mathbb{E}_{\widetilde{\mathbb{P}}^M}\left[\frac{\mathrm{d}\mathbb{Q}^M}{\mathrm{d}\mathbb{P}^M}\right]\right)^4\right]}{\mathbb{E}_{\widetilde{\mathbb{P}}}\left[\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\right)^2\right]^{2M}} \geq \frac{\frac{1}{8}\,\mathbb{E}_{\widetilde{\mathbb{P}}^M}\left[\left(\frac{\mathrm{d}\mathbb{Q}^M}{\mathrm{d}\mathbb{P}^M}\right)^4\right] - \mathbb{E}_{\widetilde{\mathbb{P}}^M}\left[\frac{\mathrm{d}\mathbb{Q}^M}{\mathrm{d}\mathbb{P}^M}\right]^4}{\mathbb{E}_{\widetilde{\mathbb{P}}}\left[\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\right)^2\right]^{2M}} \quad \text{(C.67a)}$$

$$= \frac{\frac{1}{8}\,\mathbb{E}_{\widetilde{\mathbb{P}}}\left[\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\right)^4\right]^M - \mathbb{E}_{\widetilde{\mathbb{P}}}\left[\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\right]^{4M}}{\mathbb{E}_{\widetilde{\mathbb{P}}}\left[\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\right)^2\right]^{2M}} = \frac{1}{8}\left(\underbrace{\frac{\mathbb{E}_{\widetilde{\mathbb{P}}}\left[\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\right)^4\right]}{\mathbb{E}_{\widetilde{\mathbb{P}}}\left[\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\right)^2\right]^2}}_{=:C_1}\right)^M - \left(\underbrace{\frac{\mathbb{E}_{\widetilde{\mathbb{P}}}\left[\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\right)\right]^4}{\mathbb{E}_{\widetilde{\mathbb{P}}}\left[\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\right)^2\right]^2}}_{=:C_2}\right)^M, \quad \text{(C.67b)}$$

where the second bound is implied by the c_r-inequality [199, Section 9.3]. By Jensen's inequality and since $\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}$ is not $\widetilde{\mathbb{P}}$-almost surely constant by assumption, it holds that $C_1 > 1$ and $C_2 < 1$. The claim therefore follows from combining (C.66) and (C.67).

4.) Employing the notation introduced in (C.54), we see that

$$D^{\mathrm{CE}}(\mathbb{P}^M|\mathbb{Q}^M) = \mathbb{E}_{\mathbb{Q}^M}\left[\log\left(\frac{\mathrm{d}\mathbb{Q}^M}{\mathrm{d}\mathbb{P}^M}\right)\right] = \sum_{i=1}^{M}\mathbb{E}_{\mathbb{Q}_i}\left[\log\left(\frac{\mathrm{d}\mathbb{Q}_i}{\mathrm{d}\mathbb{P}_i}\right)\right] = MD^{\mathrm{CE}}(\mathbb{P}|\mathbb{Q}). \quad \text{(C.68)}$$

Furthermore,

$$\mathrm{Var}\left(\widehat{D}^{\mathrm{CE},(K)}_{\widetilde{\mathbb{P}}^M}(\mathbb{P}^M|\mathbb{Q}^M)\right) = \frac{1}{K}\mathrm{Var}_{\widetilde{\mathbb{P}}^M}\left(\log\left(\frac{\mathrm{d}\mathbb{Q}^M}{\mathrm{d}\mathbb{P}^M}\right)\frac{\mathrm{d}\mathbb{Q}^M}{\mathrm{d}\widetilde{\mathbb{P}}^M}\right) \quad \text{(C.69a)}$$

$$= \frac{1}{K}\left(\mathbb{E}_{\widetilde{\mathbb{P}}^M}\left[\log^2\left(\frac{\mathrm{d}\mathbb{Q}^M}{\mathrm{d}\mathbb{P}^M}\right)\left(\frac{\mathrm{d}\mathbb{Q}^M}{\mathrm{d}\widetilde{\mathbb{P}}^M}\right)^2\right] - \mathbb{E}_{\widetilde{\mathbb{P}}^M}\left[\log\left(\frac{\mathrm{d}\mathbb{Q}^M}{\mathrm{d}\mathbb{P}^M}\right)\frac{\mathrm{d}\mathbb{Q}^M}{\mathrm{d}\widetilde{\mathbb{P}}^M}\right]^2\right) \quad \text{(C.69b)}$$

$$= \frac{1}{K}\left(\mathbb{E}_{\mathbb{Q}^M}\left[\log^2\left(\frac{\mathrm{d}\mathbb{Q}^M}{\mathrm{d}\mathbb{P}^M}\right)\frac{\mathrm{d}\mathbb{Q}^M}{\mathrm{d}\widetilde{\mathbb{P}}^M}\right] - M^2\mathbb{E}_{\mathbb{Q}}\left[\log\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\right)\right]^2\right). \quad \text{(C.69c)}$$

Manipulating the first term, we obtain

$$\mathbb{E}_{\mathbb{Q}^M}\left[\log^2\left(\frac{\mathrm{d}\mathbb{Q}^M}{\mathrm{d}\mathbb{P}^M}\right)\frac{\mathrm{d}\mathbb{Q}^M}{\mathrm{d}\widetilde{\mathbb{P}}^M}\right] = \mathbb{E}_{\mathbb{Q}^M}\left[\left(\sum_{i=1}^M \log\left(\frac{\mathrm{d}\mathbb{Q}_i}{\mathrm{d}\mathbb{P}_i}\right)\right)^2\frac{\mathrm{d}\mathbb{Q}^M}{\mathrm{d}\widetilde{\mathbb{P}}^M}\right] \tag{C.70a}$$

$$= \sum_{i=1}^M \mathbb{E}_{\mathbb{Q}^M}\left[\log^2\left(\frac{\mathrm{d}\mathbb{Q}_i}{\mathrm{d}\mathbb{P}_i}\right)\frac{\mathrm{d}\mathbb{Q}^M}{\mathrm{d}\widetilde{\mathbb{P}}^M}\right] + \sum_{\substack{i,j=1\\ i\neq j}}^M \mathbb{E}_{\mathbb{Q}^M}\left[\log\left(\frac{\mathrm{d}\mathbb{Q}_i}{\mathrm{d}\mathbb{P}_i}\right)\log\left(\frac{\mathrm{d}\mathbb{Q}_j}{\mathrm{d}\mathbb{P}_j}\right)\frac{\mathrm{d}\mathbb{Q}^M}{\mathrm{d}\widetilde{\mathbb{P}}^M}\right] \tag{C.70b}$$

$$= M\left(\mathbb{E}_{\mathbb{Q}}\left[\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\widetilde{\mathbb{P}}}\right]\right)^{M-1}\mathbb{E}_{\mathbb{Q}}\left[\log^2\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\right)\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\widetilde{\mathbb{P}}}\right] + \frac{M(M-1)}{2}\left(\mathbb{E}_{\mathbb{Q}}\left[\log\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\right)\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\widetilde{\mathbb{P}}}\right]\right)^2\left(\mathbb{E}_{\mathbb{Q}}\left[\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\widetilde{\mathbb{P}}}\right]\right)^{M-2}. \tag{C.70c}$$

Notice that

$$\mathbb{E}_{\mathbb{Q}}\left[\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\widetilde{\mathbb{P}}}\right] = \mathbb{E}_{\widetilde{\mathbb{P}}}\left[\left(\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\widetilde{\mathbb{P}}}\right)^2\right] = \chi^2(\mathbb{Q}|\widetilde{\mathbb{P}}) + 1. \tag{C.71}$$

The claim now follows from combining (C.68) and (C.69) in definition (4.58). □

C.4 Proofs for Chapter 5

Proof of Proposition 5.9. We start by defining the short-cuts

$$A = f_\theta(z), \qquad B = \left(\partial_{\theta_i}\log q_\theta\right)(z). \tag{C.72}$$

Let us compute the difference of the variances of the estimators to leading order in K, namely

$$\mathrm{Var}(\widehat{g}_{\mathrm{Reinforce},i}) - \mathrm{Var}(\widehat{g}_{\mathrm{VarGrad},i}) = \frac{1}{K}\mathrm{Var}(AB) + \frac{K-2}{K(K-1)}\mathbb{E}\left[(A-\mathbb{E}[A])(B-\mathbb{E}[B])\right]^2 \tag{C.73a}$$

$$- \frac{\mathrm{Var}(A)\,\mathrm{Var}(B)}{K(K-1)} - \frac{1}{K}\mathbb{E}\left[(A-\mathbb{E}[A])^2(B-\mathbb{E}[B])^2\right] \tag{C.73b}$$

$$= \frac{1}{K}\left(\mathbb{E}\left[A^2B^2\right] - \mathbb{E}[AB]^2\right) + \frac{K-2}{K(K-1)}\mathbb{E}[AB]^2 \tag{C.73c}$$

$$- \frac{1}{K}\left(\mathbb{E}\left[A^2B^2\right] - 2\,\mathbb{E}[A]\,\mathbb{E}\left[AB^2\right] + \mathbb{E}[A]^2\,\mathbb{E}\left[B^2\right]\right) + \mathcal{O}\left(\frac{1}{K^2}\right) \tag{C.73d}$$

$$= -\frac{1}{K(K-1)}\mathbb{E}[AB]^2 - \frac{1}{K}\mathbb{E}[A]\left(\mathbb{E}[A]\,\mathbb{E}\left[B^2\right] - 2\,\mathbb{E}\left[AB^2\right]\right) + \mathcal{O}\left(\frac{1}{K^2}\right) \tag{C.73e}$$

$$= \frac{1}{K}\mathbb{E}[A]\left(2\,\mathbb{E}\left[AB^2\right] - \mathbb{E}[A]\,\mathbb{E}\left[B^2\right]\right) + \mathcal{O}\left(\frac{1}{K^2}\right) \tag{C.73f}$$

$$= \frac{1}{K}\mathbb{E}[A]\,\mathbb{E}\left[B^2\right]\left(2\delta_i^{\mathrm{CV}} + \mathbb{E}[A]\right) + \mathcal{O}\left(\frac{1}{K^2}\right) \tag{C.73g}$$

and we note that with $\mathbb{E}\left[B^2\right] > 0$ the leading term is positive if

$$\mathbb{E}[A]\delta_i^{\mathrm{CV}} + \frac{1}{2}\mathbb{E}[A]^2 > 0, \tag{C.73h}$$

which is equivalent to the statement in the proposition. □

Proof of Lemma 5.14. We compute

$$f_\theta = -\frac{1}{2}\sum_{i=1}^D \log\left(\frac{\sigma_i^2}{\widetilde{\sigma}_i^2}\right) - \frac{1}{2}\sum_{i=1}^D\frac{(z_i-\mu_i)^2}{\sigma_i^2} + \frac{1}{2}\sum_{i=1}^D\frac{(z_i-\widetilde{\mu}_i)^2}{\widetilde{\sigma}_i^2} \tag{C.74}$$

and

$$\partial_{\mu_k}\log q_\theta = \frac{z_k-\mu_k}{\sigma_k^2}. \tag{C.75}$$

We again use the short-cuts

$$A = f_\theta(z), \qquad B = \left(\partial_{\mu_k}\log q_\theta\right)(z), \tag{C.76}$$

and obtain

$$\begin{aligned}
\mathrm{Cov}_{q_\theta}(A, B^2) &= \mathbb{E}_{q_\theta}\left[AB^2\right] - \mathbb{E}_{q_\theta}[A]\,\mathbb{E}_{q_\theta}\left[B^2\right] && \text{(C.77a)}\\
&= \mathbb{E}_{q_\theta}\left[\left(-\frac{1}{2}\sum_{i=1}^{D}\log\left(\frac{\sigma_i^2}{\widetilde{\sigma}_i^2}\right) - \frac{1}{2}\sum_{i=1}^{D}\frac{(z_i-\mu_i)^2}{\sigma_i^2} + \frac{1}{2}\sum_{i=1}^{D}\frac{(z_i-\widetilde{\mu}_i)^2}{\widetilde{\sigma}_i^2}\right)\left(\frac{z_k-\mu_k}{\sigma_k^2}\right)^2\right]\\
&\quad - \mathbb{E}_{q_\theta}\left[\left(-\frac{1}{2}\sum_{i=1}^{D}\log\left(\frac{\sigma_i^2}{\widetilde{\sigma}_i^2}\right) - \frac{1}{2}\sum_{i=1}^{D}\frac{(z_i-\mu_i)^2}{\sigma_i^2} + \frac{1}{2}\sum_{i=1}^{D}\frac{(z_i-\widetilde{\mu}_i)^2}{\widetilde{\sigma}_i^2}\right)\right]\mathbb{E}_{q_\theta}\left[\left(\frac{z_k-\mu_k}{\sigma_k^2}\right)^2\right] && \text{(C.77b)}\\
&= -\frac{1}{2}\left(\frac{3}{\sigma_k^2} + \frac{D-1}{\sigma_k^2}\right) + \frac{1}{2}\left(\frac{1}{\sigma_k^2}\sum_{\substack{i=1\\ i\neq k}}^{D}\frac{\sigma_i^2 + (\mu_i-\widetilde{\mu}_i)^2}{\widetilde{\sigma}_i^2} + \frac{1}{\sigma_k^2\widetilde{\sigma}_k^2}\left(3\sigma_k^2 + (\mu_k-\widetilde{\mu}_k)^2\right)\right) && \text{(C.77c)}\\
&\quad - \left(-\frac{D}{2} + \frac{1}{2}\sum_{i=1}^{D}\frac{\sigma_i^2 + (\mu_i-\widetilde{\mu}_i)^2}{\widetilde{\sigma}_i^2}\right)\frac{1}{\sigma_k^2}\\
&= -\frac{1}{\sigma_k^2} + \frac{1}{2\sigma_k^2\widetilde{\sigma}_k^2}\left(3\sigma_k^2 + (\mu-\widetilde{\mu})^2\right) - \frac{1}{2\sigma_k^2\widetilde{\sigma}_k^2}\left(\sigma_k^2 + (\mu-\widetilde{\mu})^2\right) && \text{(C.77d)}\\
&= \frac{1}{\widetilde{\sigma}_k^2} - \frac{1}{\sigma_k^2}. && \text{(C.77e)}
\end{aligned}$$

For the terms with the partial derivative w.r.t. σ_k^2 we first note that

$$\partial_{\sigma_k^2}\log q_\theta = -\frac{1}{2\sigma_k^2} + \frac{(z_k-\mu_k)^2}{2\sigma_k^4} = \frac{1}{\sigma_k^2}\partial_{\log\sigma_k^2}\log q_\theta. \tag{C.78}$$

We compute

$$\begin{aligned}
\mathbb{E}_{q_\theta}\left[AB^2\right] &= \mathbb{E}_{q_\theta}\left[\frac{(z_k-\mu_k)^2}{4\sigma_k^6}\sum_{i=1}^{D}\frac{(z_i-\mu_i)^2}{\sigma_i^2} - \frac{(z_k-\mu_k)^4}{8\sigma_k^8}\sum_{i=1}^{D}\frac{(z_i-\mu_i)^2}{\sigma_i^2}\right.\\
&\qquad \left. - \frac{(z_k-\mu_k)^2}{4\sigma_k^6}\sum_{i=1}^{D}\frac{(z_i-\widetilde{\mu}_i)^2}{\widetilde{\sigma}_i^2} + \frac{(z_k-\mu_k)^4}{8\sigma_k^8}\sum_{i=1}^{D}\frac{(z_i-\widetilde{\mu}_i)^2}{\widetilde{\sigma}_i^2}\right] && \text{(C.79a)}\\
&= -\frac{1}{8\sigma_k^4}(D+8) + \frac{1}{8\sigma_k^4\widetilde{\sigma}_k^2}(9\sigma_k^2 + (\mu_k-\widetilde{\mu}_k)^2) + \frac{1}{8\sigma_k^4}\sum_{\substack{i=1\\ i\neq k}}^{d}\frac{\sigma_i^2 + (\mu_i-\widetilde{\mu}_i)^2}{\widetilde{\sigma}_i^2}, && \text{(C.79b)}
\end{aligned}$$

and similarly

$$\mathbb{E}_{q_\theta}[A]\,\mathbb{E}_{q_\theta}\left[B^2\right] = -\frac{D}{8\sigma_k^4} + \frac{1}{8\sigma_k^4\widetilde{\sigma}_k^2}\left(\sigma_k^2 + (\mu_k-\widetilde{\mu}_k)^2\right) + \frac{1}{8\sigma_k^4}\sum_{\substack{i=1\\ i\neq k}}^{D}\frac{\sigma_i^2 + (\mu_i-\widetilde{\mu}_i)^2}{\widetilde{\sigma}_i^2}. \tag{C.80}$$

We therefore get the result by again computing $\mathrm{Cov}_{q_\theta}(A, B^2) = \mathbb{E}_{q_\theta}\left[AB^2\right] - \mathbb{E}_{q_\theta}[A]\,\mathbb{E}_{q_\theta}\left[B^2\right]$. The partial derivative w.r.t. $\log\sigma_k^2$ can be recovered from (C.78). □

C.5 Proofs for Chapter 6

Proof of Lemma 6.1. Let $\varphi^C(a) = \mathbb{E}[B|A=a]$. We compute

$$\begin{aligned}
\mathbb{E}\left[(\varphi(A)-B)^2\right] &= \mathbb{E}\left[(\varphi(A)-\varphi^C(A)+\varphi^C(A)-B)^2\right] && \text{(C.81a)}\\
&= \mathbb{E}\left[(\varphi(A)-\varphi^C(A))^2\right] + \mathbb{E}\left[(\varphi^C(A)-B)^2\right] - 2\,\mathbb{E}\left[(\varphi(A)-\varphi^C(A))(\varphi^C(A)-B)\right], && \text{(C.81b)}
\end{aligned}$$

which is minimized by $\varphi = \varphi^C$ since the last term is equal to[53]

$$\mathbb{E}_A\left[\mathbb{E}_{B|A}\left[(\varphi(A)-\varphi^C(A))(\varphi^C(A)-B)\right]\right] = \mathbb{E}_A\left[(\varphi(A)-\varphi^C(A))\mathbb{E}_{B|A}\left[(\varphi^C(A)-B)\right]\right] = 0. \tag{C.82}$$

Therefore $\varphi^* = \varphi^C$. □

[53] Here the notation $\mathbb{E}_A$ refers to the expectation over A, whereas $\mathbb{E}_{B|A}$ refers to the expectation over B conditional on A.

Proof of Lemma 6.10. We assume the generalized FBSDE system as in (6.45). The backward iteration (6.17) then writes

$$Y_{t_n}^v = \mathbb{E}\left[Y_{t_{n+1}}^v + \int_{t_n}^{t_{n+1}} \left(f(X_s^v, s) - \frac{1}{2}|Z_s^v|^2 - Z_s^v \cdot v(X_s^v, s) \right) \mathrm{d}s \,\middle|\, X_{t_n}^v \right]. \tag{C.83}$$

The choice of the nonlinearity implies that the running costs in corresponding the control problem take the form $f(x,s) + \frac{1}{2}|u(x,s)|^2$ and in this case we have $u^*(X_s^v, s) = -\sigma^\top \nabla V(X_s^v, s) = -Z_s^v$ (see also Corollary 2.10). Therefore, taking $v = u^*$ yields

$$Y_{t_n}^{u^*} = \mathbb{E}\left[Y_{t_{n+1}}^{u^*} + \int_{t_i}^{t_{n+1}} \left(f(X_s^{u^*}, s) + \frac{1}{2}|u^*(X_s^{u^*}, s)|^2 \right) \mathrm{d}s \,\middle|\, X_{t_n}^{u^*} \right] \tag{C.84a}$$

$$= \inf_{u \in \mathcal{U}} \mathbb{E}\left[Y_{t_{n+1}}^u + \int_{t_n}^{t_{n+1}} \left(f(X_s^u, s) + \frac{1}{2}|u(X_s^u, s)|^2 \right) \mathrm{d}s \,\middle|\, X_{t_n}^u \right], \tag{C.84b}$$

where the last line follows from the definition of the optimal control u^* and the dynamic programming principle stated in Theorem 2.2. A comparison to (2.8) and noting that $Y_{t_n}^{u^*} = V(X_{t_n}^{u^*}, t_n)$ yields the statement. □

Bibliography

[1] A. Abdelfattah, M. Baboulin, V. Dobrev, J. Dongarra, C. Earl, J. Falcou, A. Haidar, I. Karlin, T. Kolev, I. Masliah, and S. Tomov. High-performance tensor contractions for GPUs. *Procedia Computer Science*, 80:108–118, 2016.

[2] Y. Achdou. Finite difference methods for mean field games. In *Hamilton-Jacobi equations: approximations, numerical analysis and applications*, pages 1–47. Springer, 2013.

[3] S. Agapiou, O. Papaspiliopoulos, D. Sanz-Alonso, and A. M. Stuart. Importance sampling: computational complexity and intrinsic dimension. *Statistical Science*, 32(3), 2015.

[4] Ö. D. Akyildiz and J. Míguez. Convergence rates for optimised adaptive importance samplers. *arXiv preprint arXiv:1903.12044*, 2019.

[5] A. Alfonsi. *Affine diffusions and related processes: simulation, theory and applications*, volume 6. Springer, 2015.

[6] S. Asmussen, P. Dupuis, R. Rubinstein, and H. Wang. Importance sampling for rare events. *Aarhus Univ., Aarhus, Denmark, Tech. Rep.*, 2011.

[7] S. Asmussen and P. W. Glynn. *Stochastic Simulation: Algorithms and Analysis*. Springer, New York, 2007.

[8] F. Bach. Breaking the curse of dimensionality with convex neural networks. *The Journal of Machine Learning Research*, 18(1):629–681, 2017.

[9] F. Baudoin. Conditioned stochastic differential equations: theory, examples and application to finance. *Stochastic Processes and their Applications*, 100(1-2):109–145, 2002.

[10] C. Bayer, M. Eigel, L. Sallandt, and P. Trunschke. Pricing high-dimensional Bermudan options with hierarchical tensor formats. *arXiv preprint arXiv:2103.01934*, 2021.

[11] C. Beck, S. Becker, P. Cheridito, A. Jentzen, and A. Neufeld. Deep splitting method for parabolic PDEs. *arXiv preprint arXiv:1907.03452*, 2019.

[12] C. Beck, S. Becker, P. Grohs, N. Jaafari, and A. Jentzen. Solving stochastic differential equations and Kolmogorov equations by means of deep learning. *arXiv preprint arXiv:1806.00421*, 2018.

[13] C. Beck, W. E, and A. Jentzen. Machine learning approximation algorithms for high-dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations. *Journal of Nonlinear Science*, 29(4):1563–1619, 2019.

[14] C. Beck, F. Hornung, M. Hutzenthaler, A. Jentzen, and T. Kruse. Overcoming the curse of dimensionality in the numerical approximation of Allen-Cahn partial differential equations via truncated full-history recursive multilevel Picard approximations. *arXiv preprint arXiv:1907.06729*, 2019.

[15] S. Becker, P. Cheridito, and A. Jentzen. Deep optimal stopping. *Journal of Machine Learning Research*, 20, 2019.

[16] S. Becker, P. Cheridito, A. Jentzen, and T. Welti. Solving high-dimensional optimal stopping problems using deep learning. *arXiv preprint arXiv:1908.01602*, 2019.

[17] S. Becker, C. Hartmann, M. Redmann, and L. Richter. Error bounds for model reduction of feedback-controlled linear stochastic dynamics on Hilbert spaces. *Stochastic Processes and their Applications*, 2022.

[18] S. Becker and L. Richter. Model order reduction for (stochastic-) delay equations with error bounds. *arXiv preprint arXiv:2008.12288*, 2020.

[19] R. Bellman. *Dynamic programming*. Princeton University Press, 1957.

[20] C. Bender and R. Denk. A forward scheme for backward SDEs. *Stochastic processes and their applications*, 117(12):1793–1812, 2007.

[21] C. Bender and J. Steiner. Least-squares Monte Carlo for backward SDEs. In *Numerical methods in finance*, pages 257–289. Springer, 2012.

[22] P. Beneventano, P. Cheridito, A. Jentzen, and P. von Wurstemberger. High-dimensional approximation spaces of artificial neural networks and applications to partial differential equations. *arXiv preprint arXiv:2012.04326*, 2020.

[23] T. Bengtsson, P. Bickel, and B. Li. Curse-of-dimensionality revisited: collapse of the particle filter in very large scale systems. In *Probability and statistics: Essays in honor of David A. Freedman*, pages 316–334. Institute of Mathematical Statistics, 2008.

[24] A. Bensoussan. *Perturbation methods in optimal control*, volume 5. Wiley, 1988.

[25] H. Berestycki, L. Nirenberg, and S. S. Varadhan. The principal eigenvalue and maximum principle for second-order elliptic operators in general domains. *Communications on Pure and Applied Mathematics*, 47(1):47–92, 1994.

[26] N. Berglund. Kramers' law: validity, derivations and generalisations. *arXiv preprint arXiv:1106.5799*, 2011.

[27] J. Berner, M. Dablander, and P. Grohs. Numerically solving parametric families of high-dimensional Kolmogorov partial differential equations via deep learning. *arXiv preprint arXiv:2011.04602*, 2020.

[28] J. Berner, D. Elbrächter, and P. Grohs. How degenerate is the parametrization of neural networks with the ReLU activation function? *arXiv preprint arXiv:1905.09803*, 2019.

[29] J. Berner, P. Grohs, and A. Jentzen. Analysis of the generalization error: empirical risk minimization over deep artificial neural networks overcomes the curse of dimensionality in the numerical approximation of Black-Scholes partial differential equations. *arXiv preprint arXiv:1809.03062*, 2018.

[30] D. P. Bertsekas. Dynamic programming and optimal control, 3rd edition, volume II. *Belmont, MA: Athena Scientific*, 2011.

[31] A. Beurling. An automorphism of product measures. *Annals of Mathematics*:189–200, 1960.

[32] P. Bickel, B. Li, and T. Bengtsson. Sharp failure rates for the bootstrap particle filter in high dimensions. In *Pushing the limits of contemporary statistics: Contributions in honor of Jayanta K. Ghosh*, pages 318–329. Institute of Mathematical Statistics, 2008.

[33] J. Bierkens and H. J. Kappen. Explicit solution of relative entropy weighted control. *Systems & Control Letters*, 72:36–43, 2014.

[34] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[35] J.-M. Bismut. Conjugate convex functions in optimal stochastic control. *Journal of Mathematical Analysis and Applications*, 44(2):384–404, 1973.

[36] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: a review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

[37] L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

[38] B. Bouchard and S. Menozzi. Strong approximations of BSDEs in a domain. *Bernoulli*, 15(4):1117–1147, 2009.

[39] B. Bouchard and N. Touzi. Discrete-time approximation and Monte-Carlo simulation of backward stochastic differential equations. *Stochastic Processes and their applications*, 111(2):175–206, 2004.

[40] M. Boué and P. Dupuis. A variational representation for certain functionals of Brownian motion. *The Annals of Probability*, 26(4):1641–1659, 1998.

[41] A. Bovier, V. Gayrard, and M. Klein. Metastability in reversible diffusion processes II: precise asymptotics for small eigenvalues. *J. Eur. Math. Soc.*, 7(1):69–99, 2005.

[42] F. Buchmann and W. Petersen. Solving Dirichlet problems numerically using the Feynman-Kac representation. *BIT Numerical Mathematics*, 43(3):519–540, 2003.

[43] J. Bucklew. *Introduction to rare event simulation*. Springer Science & Business Media, 2013.

[44] M. F. Bugallo, V. Elvira, L. Martino, D. Luengo, J. Miguez, and P. M. Djuric. Adaptive Importance Sampling: The past, the present, and the future. *IEEE Signal Processing Magazine*, 34(4):60–79, 2017.

[45] P. Carbonetto, M. King, and F. Hamze. A stochastic approximation method for inference in probabilistic graphical models. In *Advances in Neural Information Processing Systems*, 2009.

[46] R. Carmona. *Lectures on BSDEs, stochastic control, and stochastic differential games with financial applications*, volume 1. SIAM, 2016.

[47] R. Carmona and F. Delarue. *Probabilistic Theory of Mean Field Games with Applications I-II*. Springer, 2018.

[48] R. Carmona and M. Laurière. Convergence analysis of machine learning algorithms for the numerical solution of mean field control and games: I – the ergodic case. *arXiv preprint arXiv:1907.05980*, 2019.

[49] R. Carmona and M. Laurière. Convergence analysis of machine learning algorithms for the numerical solution of mean field control and games: II – the finite horizon case. *arXiv preprint arXiv:1908.01613*, 2019.

[50] Q. Chan-Wai-Nam, J. Mikael, and X. Warin. Machine learning for semilinear PDEs. *Journal of Scientific Computing*, 79(3):1667–1712, 2019.

[51] S. Chatterjee and P. Diaconis. The sample size required in importance sampling. *The Annals of Applied Probability*, 28(2):1099–1135, 2018.

[52] Y. Chen. Another look at rejection sampling through importance sampling. *Statistics & probability letters*, 72(4):277–283, 2005.

[53] P. Cheridito, A. Jentzen, and F. Rossmannek. Efficient approximation of high-dimensional functions with deep neural networks. *arXiv preprint arXiv:1912.04310*, 2019.

[54] P. Cheridito, H. M. Soner, N. Touzi, and N. Victoir. Second-order backward stochastic differential equations and fully nonlinear parabolic PDEs. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 60(7):1081–1110, 2007.

[55] R. Chetrite and H. Touchette. Nonequilibrium Markov processes conditioned on large deviations. In *Annales Henri Poincaré*, volume 16 of number 9, pages 2005–2057. Springer, 2015.

[56] E. Cho, M. J. Cho, and J. Eltinge. The variance of sample variance from a finite population. *International Journal of Pure and Applied Mathematics*, 21(3):389, 2005.

[57] Y. Cong, M. Zhao, K. Bai, and L. Carin. GO gradient for expectation-based objectives. In *International Conference on Learning Representations*, 2019.

[58] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2012.

[59] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[60] P. Dai Pra. A stochastic control approach to reciprocal diffusion processes. *Applied mathematics and Optimization*, 23(1):313–329, 1991.

[61] P. Dai Pra, L. Meneghini, and W. J. Runggaldier. Connections between stochastic control and dynamic games. *Mathematics of Control, Signals and Systems*, 9(4):303–326, 1996.

[62] P. Dai Pra and M. Pavon. On the Markov processes of Schrödinger, the Feynman-Kac formula and stochastic control. In *Realization and Modelling in System Theory*, pages 497–504. Springer, 1990.

[63] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134(1):19–67, 2005.

[64] P. Del Moral. *Feynman-Kac formulae: Genealogical and interacting particle systems with applications*. Springer, 2004.

[65] P. Del Moral. *Mean field simulation for Monte Carlo integration*. Chapman and Hall/CRC, 2013.

[66] P. Del Moral and L. Miclo. Branching and interacting particle systems approximations of Feynman-Kac formulae with applications to non-linear filtering. In *Seminaire de probabilites XXXIV*, pages 1–145. Springer, 2000.

[67] C. Dellacherie and P.-A. Meyer. *Probabilities and potential, C: potential theory for discrete and continuous semigroups*. Elsevier, 2011.

[68] A. Dembo and O. Zeitouni. *Large Deviations Techniques and Applications*. Springer Berlin Heidelberg, 2009.

[69] J.-D. Deuschel and D. W. Stroock. *Large deviations*, volume 342. American Mathematical Soc., 2001.

[70] A. B. Dieng, D. Tran, R. Ranganath, J. Paisley, and D. M. Blei. Variational inference via χ-upper bound minimization. In *Advances in Neural Information Processing Systems*, 2017.

[71] L. Dinh, D. Krueger, and Y. Bengio. Nice: non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[72] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real NVP. *arXiv preprint arXiv:1605.08803*, 2016.

[73] S. V. Dolgov, B. N. Khoromskij, and I. V. Oseledets. Fast solution of parabolic problems in the tensor train/quantized tensor train format with initial application to the Fokker-Planck equation. *SIAM Journal on Scientific Computing*, 34(6):A3016–A3038, 2012.

[74] Z. Dong, A. Mnih, and G. Tucker. DisARM: an antithetic gradient estimator for binary latent variables. In *Advances in Neural Information Processing Systems*, 2020.

[75] J. L. Doob. Conditional Brownian motion and the boundary limits of harmonic functions. *Bulletin de la Société Mathématique de France*, 85:431–458, 1957.

[76] J. L. Doob. *Classical potential theory and its probabilistic counterpart: Advanced problems*, volume 262. Springer Science & Business Media, 2012.

[77] A. Doucet, N. De Freitas, and N. Gordon. An introduction to sequential Monte Carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer, 2001.

[78] S. S. Dragomir and V Gluscevic. Some inequalities for the Kullback-Leibler and χ^2-distances in information theory and applications. *RGMIA research report collection*, 3(2):199–210, 2000.

[79] P. Dupuis, K. Spiliopoulos, and H. Wang. Importance sampling for multiscale diffusions. *Multiscale Modeling & Simulation*, 10(1):1–27, 2012.

[80] P. Dupuis, K. Spiliopoulos, and X. Zhou. Escaping from an attractor: importance sampling and rest points I. *The Annals of Applied Probability*:2909–2958, 2015.

[81] P. Dupuis and H. Wang. Importance sampling, large deviations, and differential games. *Stochastics: An International Journal of Probability and Stochastic Processes*, 76(6):481–508, 2004.

[82] P. Dupuis and H. Wang. Subsolutions of an Isaacs equation and efficient schemes for importance sampling. *Mathematics of Operations Research*, 32(3):723–757, 2007.

[83] R. Durrett and R. Durrett. *Brownian motion and martingales in analysis*. Wadsworth Advanced Books & Software California, 1984.

[84] W. E and E. Vanden-Eijnden. Metastability, conformation dynamics, and transition pathways in complex systems. In *Multiscale modelling and simulation*, pages 35–68. Springer, 2004.

[85] W. E and B. Yu. The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1):1–12, 2018.

[86] W. E, J. Han, and A. Jentzen. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematics and Statistics*, 5(4):349–380, 2017.

[87] W. E, M. Hutzenthaler, A. Jentzen, and T. Kruse. On multilevel Picard numerical approximations for high-dimensional nonlinear parabolic partial differential equations and high-dimensional nonlinear backward stochastic differential equations. *Journal of Scientific Computing*, 79(3):1534–1571, 2019.

[88] W. E, C. Ma, and L. Wu. Barron spaces and the compositional function spaces for neural network models. *arXiv preprint arXiv:1906.08039*, 2019.

[89] M. Eigel, R. Schneider, P. Trunschke, and S. Wolf. Variational Monte Carlo – bridging concepts of machine learning and high-dimensional partial differential equations. *Advances in Computational Mathematics*, 45(5-6):2503–2532, 2019.

[90] N. El Karoui, S. Peng, and M. C. Quenez. Backward stochastic differential equations in finance. *Mathematical finance*, 7(1):1–71, 1997.

[91] D. Elbrächter, P. Grohs, A. Jentzen, and C. Schwab. DNN expression rate analysis of high-dimensional PDEs: application to option pricing. *arXiv preprint arXiv:1809.07669*, 2018.

[92] R. Eldan and O. Shamir. The power of depth for feedforward neural networks. In *Conference on learning theory*, pages 907–940, 2016.

[93] L. C. Evans. *Partial differential equations*. American Mathematical Society, Providence, R.I., 2010. ISBN: 9780821849743 0821849743.

[94] G. Ferré. *Large Deviations Theory in Statistical Physics: Some Theoretical and Numerical Aspects*. PhD thesis, Université Marne La Vallée, 2019.

[95] G. Ferré and H. Touchette. Adaptive sampling of large deviations. *Journal of Statistical Physics*, 172(6):1525–1544, 2018.

[96] W. H. Fleming. Stochastic control for small noise intensities. *SIAM Journal on Control*, 9(3):473–517, 1971.

[97] W. H. Fleming and R. W. Rishel. *Deterministic and stochastic optimal control*, volume 1. Springer Science & Business Media, 2012.

[98] W. H. Fleming and H. M. Soner. *Controlled Markov processes and viscosity solutions*, volume 25. Springer Science & Business Media, 2006.

[99] W. Fleming. Controlled diffusions under polynomial growth conditions. *Control Theory and the Calculus of Variations*:209–234, 1969.

[100] M. I. Freidlin and A. D. Wentzell. *Random perturbations of dynamical systems*. Springer, 1998.

[101] A. Gelman and X.-L. Meng. Simulating normalizing constants: from importance sampling to bridge sampling to path sampling. *Statistical science*:163–185, 1998.

[102] M. Germain, H. Pham, and X. Warin. Deep backward multistep schemes for nonlinear PDEs and approximation error analysis. *arXiv preprint arXiv:2006.01496*, 2020.

[103] S. Ghosal, J. K. Ghosh, and A. W. Van Der Vaart. Convergence rates of posterior distributions. *Annals of Statistics*, 28(2):500–531, 2000.

[104] A. L. Gibbs and F. E. Su. On choosing and bounding probability metrics. *International statistical review*, 70(3):419–435, 2002.

[105] D. Gilbarg and N. S. Trudinger. *Elliptic partial differential equations of second order*. Springer, 2015.

[106] P. Glasserman. *Monte Carlo methods in financial engineering*, volume 53. Springer Science & Business Media, 2013.

[107] P. Glasserman and J. Li. Importance sampling for portfolio credit risk. *Management science*, 51(11):1643–1656, 2005.

[108] P. Glasserman and Y. Wang. Counterexamples in importance sampling for large deviations probabilities. *The Annals of Applied Probability*, 7(3):731–746, 1997.

[109] E. Gobet. *Monte Carlo methods and stochastic processes: from linear to non-linear*. CRC Press, 2016.

[110] E. Gobet, J.-P. Lemor, and X. Warin. A regression-based Monte Carlo method to solve backward stochastic differential equations. *The Annals of Applied Probability*, 15(3):2172–2202, 2005.

[111] E. Gobet and R. Munos. Sensitivity analysis using Itô-Malliavin calculus and martingales, and application to stochastic optimal control. *SIAM Journal on control and optimization*, 43(5):1676–1713, 2005.

[112] E. Gobet and P. Turkedjiev. Linear regression MDP scheme for discrete backward stochastic differential equations under general conditions. *Mathematics of Computation*, 85(299):1359–1391, 2016.

[113] V. Gómez, H. J. Kappen, J. Peters, and G. Neumann. Policy search for path integral control. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 482–497. Springer, 2014.

[114] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[115] L. Grasedyck and S. Krämer. Stable ALS approximation in the TT-format for rank-adaptive tensor completion. *Numerische Mathematik*, 143(4):855–904, 2019.

[116] W. Grathwohl, D. Choi, Y. Wu, G. Roeder, and D. Duvenaud. Backpropagation through the void: optimizing control variates for black-box gradient estimation. In *International Conference on Learning Representations*, 2018.

[117] P. Grohs and L. Herrmann. Deep neural network approximation for high-dimensional elliptic PDEs with boundary conditions. *arXiv preprint arXiv:2007.05384*, 2020.

[118] P. Grohs, F. Hornung, A. Jentzen, and P. Von Wurstemberger. A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of Black-Scholes partial differential equations. *arXiv preprint arXiv:1809.02362*, 2018.

[119] P. Grohs, A. Jentzen, and D. Salimova. Deep neural network approximations for Monte Carlo algorithms. *arXiv preprint arXiv:1908.10828*, 2019.

[120] S. Gu, S. Levine, I. Sutskever, and A. Mnih. MuProp: unbiased backpropagation for stochastic neural networks. In *International Conference on Machine Learning*, 2016.

[121] B. C. Hall. *Quantum theory for mathematicians*, volume 267. Springer, 2013.

[122] J. Han, A. Jentzen, and W. E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018.

[123] J. Han and J. Long. Convergence of the deep BSDE method for coupled FBSDEs. *Probability, Uncertainty and Quantitative Risk*, 5(1):1–33, 2020.

[124] J. Han, J. Lu, and M. Zhou. Solving high-dimensional eigenvalue problems using deep neural networks: a diffusion Monte Carlo like approach. *arXiv preprint arXiv:2002.02600*, 2020.

[125] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del R'ıo, M. Wiebe, P. Peterson, P. G'erard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, Sept. 2020. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

[126] C. Hartmann, R. Banisch, M. Sarich, T. Badowski, and C. Schütte. Characterization of rare events in molecular dynamics. *Entropy*, 16(1):350–376, 2014.

[127] C. Hartmann, O. Kebiri, L. Neureither, and L. Richter. Variational approach to rare event simulation using least-squares regression. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(6):063107, 2019.

[128] C. Hartmann, J. C. Latorre, G. A. Pavliotis, and W. Zhang. Optimal control of multiscale systems using reduced-order models. *J. Computational Dynamics*, 1:279–306, 2014.

[129] C. Hartmann and L. Richter. Nonasymptotic bounds for suboptimal importance sampling. *arXiv preprint arXiv:2102.09606*, 2021.

[130] C. Hartmann, L. Richter, C. Schütte, and W. Zhang. Variational characterization of free energy: Theory and algorithms. *Entropy*, 19(11):626, 2017.

[131] C. Hartmann and C. Schütte. Efficient rare event simulation by optimal nonequilibrium forcing. *Journal of Statistical Mechanics: Theory and Experiment*, 2012(11):P11004, 2012.

[132] C. Hartmann, C. Schütte, M. Weber, and W. Zhang. Importance sampling in path space for diffusion processes with slow-fast variables. *Probability Theory and Related Fields*, 170(1-2):177–228, 2018.

[133] C. Hartmann, C. Schütte, and W. Zhang. Model reduction algorithms for optimal control and importance sampling of diffusions. *Nonlinearity*, 29(8):2298, 2016.

[134] C. Hartmann, C. Schütte, and W. Zhang. Jarzynski's equality, fluctuation theorems, and variance reduction: mathematical analysis and numerical algorithms. *Journal of Statistical Physics*, 175(6):1214–1261, 2019.

[135] E. Hausenblas. A numerical scheme using Itô excursions for simulating local time resp. stochastic differential equations with reflection. *Osaka journal of mathematics*, 36(1):105–137, 1999.

[136] J. Heng, A. N. Bishop, G. Deligiannidis, and A. Doucet. Controlled sequential Monte Carlo. *arXiv preprint arXiv:1708.08396*, 2017.

[137] J. Heng, A. Doucet, and Y. Pokern. Gibbs flow for approximate transport with applications to Bayesian computation. *arXiv preprint arXiv:1509.08787*, 2015.

[138] J. M. Hernández-Lobato, Y. Li, M. Rowland, D. Hernández-Lobato, T. Bui, and R. E. Turner. Black-box α-divergence minimization. In *International Conference on Machine Learning*, 2016.

[139] S. Holtz, T. Rohwedder, and R. Schneider. The alternating linear scheme for tensor optimization in the tensor train format. *SIAM J. Sci. Comput.*, 34(2):A683–A713, 2012. DOI: 10.1137/100818893. eprint: https://doi.org/10.1137/100818893. URL: https://doi.org/10.1137/100818893.

[140] S. Holtz, T. Rohwedder, and R. Schneider. On manifolds of tensors of fixed TT-rank. *Numerische Mathematik*, 120(4):701–731, 2012.

[141] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[142] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[143] B. Huber and S. Wolf. Xerus - a general purpose tensor library. https://libxerus.org/, 2014–2017.

[144] C. Huré, H. Pham, and X. Warin. Deep backward schemes for high-dimensional nonlinear PDEs. *Mathematics of Computation*, 89(324):1547–1579, 2020.

[145] M. Hutzenthaler, A. Jentzen, and T. Kruse. Multilevel Picard iterations for solving smooth semilinear parabolic heat equations. *arXiv preprint arXiv:1607.03295*, 2016.

[146] M. Hutzenthaler, A. Jentzen, and T. Kruse. Overcoming the curse of dimensionality in the numerical approximation of parabolic partial differential equations with gradient-dependent nonlinearities. *arXiv preprint arXiv:1912.02571*, 2019.

[147] M. Hutzenthaler, A. Jentzen, T. Kruse, and T. A. Nguyen. A proof that rectified deep neural networks overcome the curse of dimensionality in the numerical approximation of semilinear heat equations. *arXiv preprint arXiv:1901.10854*, 2019.

[148] M. Hutzenthaler, A. Jentzen, T. Kruse, T. A. Nguyen, and P. von Wurstemberger. Overcoming the curse of dimensionality in the numerical approximation of semilinear parabolic partial differential equations. *arXiv preprint arXiv:1807.01212*, 2018.

[149] M. Hutzenthaler, A. Jentzen, and P. von Wurstemberger. Overcoming the curse of dimensionality in the approximative pricing of financial derivatives with default risks. *arXiv preprint arXiv:1903.05985*, 2019.

[150] M. Hutzenthaler and T. Kruse. Multilevel Picard approximations of high-dimensional semilinear parabolic differential equations with gradient-dependent nonlinearities. *SIAM Journal on Numerical Analysis*, 58(2):929–961, 2020.

[151] C. B. Hyndman. Forward-backward SDEs and the CIR model. *Statistics & probability letters*, 77(17):1676–1682, 2007.

[152] B. Jamison. Reciprocal processes. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 30(1):65–86, 1974.

[153] B. Jamison. The Markov processes of Schrödinger. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 32(4):323–331, 1975.

[154] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with Gumbel-softmax. In *International Conference on Learning Representations*, 2017.

[155] S. Janson and J. Tysk. Feynman-Kac formulas for Black-Scholes-type operators. *Bulletin of the London Mathematical Society*, 38(2):269–282, 2006.

[156] C. Jarzynski. Nonequilibrium equality for free energy differences. *Physical Review Letters*, 78(14):2690, 1997.

[157] A. Jentzen, D. Salimova, and T. Welti. A proof that deep artificial neural networks overcome the curse of dimensionality in the numerical approximation of Kolmogorov partial differential equations with constant diffusion and nonlinear drift coefficients. *arXiv preprint arXiv:1809.07321*, 2018.

[158] Y. Jiang and J. Li. Convergence of the deep BSDE method for FBSDEs with non-Lipschitz coefficients. *arXiv preprint arXiv:2101.01869*, 2021.

[159] H. J. Kappen. An introduction to stochastic control theory, path integrals and reinforcement learning. In *AIP conference proceedings*, volume 887 of number 1, pages 149–181. American Institute of Physics, 2007.

[160] H. J. Kappen, V. Gómez, and M. Opper. Optimal control as a graphical model inference problem. *Machine learning*, 87(2):159–182, 2012.

[161] H. J. Kappen and H. C. Ruiz. Adaptive importance sampling for control and inference. *Journal of Statistical Physics*, 162(5):1244–1266, 2016.

[162] I. Karatzas and S. E. Shreve. *Brownian Motion and Stochastic Calculus*. Springer, 1998.

[163] V. Kazeev, I. Oseledets, M. Rakhuba, and C. Schwab. QTT-finite-element approximation for multiscale problems. In Technical Report 2016-06 Seminar for Applied Mathematics, ETH Zürich, 2016, 2016.

[164] V. A. Kazeev and B. N. Khoromskij. Low-rank explicit QTT representation of the Laplace operator and its inverse. *SIAM journal on matrix analysis and applications*, 33(3):742–758, 2012.

[165] O. Kebiri, L. Neureither, and C. Hartmann. Adaptive importance sampling with forward-backward stochastic differential equations. In *International workshop on Stochastic Dynamics out of Equilibrium*, pages 265–281. Springer, 2017.

[166] Y. Khoo, J. Lu, and L. Ying. Solving for high-dimensional committor functions using artificial neural networks. *Research in the Mathematical Sciences*, 6(1):1, 2019.

[167] B. N. Khoromskij. Tensors-structured numerical methods in scientific computing: survey on recent advances. *Chemometrics and Intelligent Laboratory Systems*, 110(1):1–19, 2012.

[168] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

[169] D. P. Kingma and J. Ba. Adam: a method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[170] F. Klebaner and R. Liptser. When a stochastic exponential is a true martingale. Extension of the Beneš method. *Theory of Probability and its Applications*, 58(1):38–62, 2014.

[171] A. Klenke. *Probability theory: a comprehensive course*. Springer Science & Business Media, 2013.

[172] P. E. Kloeden and E. Platen. Stochastic differential equations. In *Numerical Solution of Stochastic Differential Equations*, pages 103–160. Springer, 1992.

[173] P. E. Kloeden and E. Platen. *Numerical solution of stochastic differential equations*, volume 23. Springer Science & Business Media, 2013.

[174] M. Kobylanski. Backward stochastic differential equations and partial differential equations with quadratic growth. *Annals of Probability*:558–602, 2000.

[175] I. Kobyzev, S. Prince, and M. Brubaker. Normalizing flows: an introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[176] W. König. *Große Abweichungen: Techniken und Anwendungen*. Springer, 2020.

[177] W. Kool, H. van Hoof, and M. Welling. Buy 4 REINFORCE samples, get a baseline for free! In *ICLR Workshop on Deep Reinforcement Learning Meets Structured Prediction*, 2019.

[178] H. A. Kramers. Brownian motion in a field of force and the diffusion model of chemical reactions. *Physica*, 7(4):284–304, 1940.

[179] S. Kremsner, A. Steinicke, and M. Szölgyenyi. A deep neural network algorithm for semilinear elliptic PDEs with applications in insurance mathematics. *arXiv preprint arXiv:2010.15757*, 2020.

[180] N. V. Krylov. *Nonlinear elliptic and parabolic equations of the second order*, volume 7. Springer, 1987.

[181] H. Kunita. Stochastic differential equations and stochastic flows of diffeomorphisms. In *Ecole d'été de probabilités de Saint-Flour XII-1982*, pages 143–303. Springer, 1984.

[182] H. Kushner and P. G. Dupuis. *Numerical methods for stochastic control problems in continuous time*, volume 24. Springer Science & Business Media, 2013.

[183] B. Kutschan. Tangent cones to tensor train varieties. *Linear Algebra and its Applications*, 544:370–390, 2018.

[184] I. E. Lagaris, A. Likas, and D. I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks*, 9(5):987–1000, 1998.

[185] I. E. Lagaris, A. C. Likas, and D. G. Papageorgiou. Neural-network methods for boundary value problems with irregular boundaries. *IEEE Transactions on Neural Networks*, 11(5):1041–1049, 2000.

[186] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[187] K. Law, A. Stuart, and K. Zygalakis. Data assimilation. *Cham, Switzerland: Springer*, 214, 2015.

[188] W. Lee, H. Yu, and H. Yang. Reparameterization gradient for non-differentiable models. In *Advances in Neural Information Processing Systems*, 2018.

[189] B. Leimkuhler and C. Matthews. *Molecular Dynamics*. Springer, 2016.

[190] T. Lelievre and G. Stoltz. Partial differential equations and stochastic methods in molecular dynamics. *Acta Numerica*, 25:681, 2016.

[191] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993.

[192] B. Li, T. Bengtsson, and P. Bickel. Curse-of-dimensionality revisited: Collapse of importance sampling in very high-dimensional systems. *Tech Reports, Department of Statistics, UC Berkeley*, 696:1–18, 2005.

[193] Q. Li, B. Lin, and W. Ren. Computing committor functions for the study of rare events using deep learning. *The Journal of Chemical Physics*, 151(5):054112, 2019.

[194] Y. Li and R. E. Turner. Rényi divergence variational inference. In *Advances in Neural Information Processing Systems*, 2016.

[195] H. C. Lie. Convexity of a stochastic control functional related to importance sampling of Itô diffusions. *arXiv preprint arXiv:1603.05900*, 2016.

[196] F. Liese and I. Vajda. On divergences and informations in statistics and information theory. *IEEE Transactions on Information Theory*, 52(10):4394–4412, 2006.

[197] P.-L. Lions. Optimal control of diffusion processes and Hamilton-Jacobi–Bellman equations part 2: viscosity solutions and uniqueness. *Communications in partial differential equations*, 8(11):1229–1276, 1983.

[198] J. S. Liu. *Monte Carlo strategies in scientific computing*. Springer Science & Business Media, 2008.

[199] M. Loeve. *Probability theory*, volume 1963. Springer, 1963.

[200] F. A. Longstaff and E. S. Schwartz. Valuing american options by simulation: a simple least-squares approach. *The review of financial studies*, 14(1):113–147, 2001.

[201] J. Lu and J. Nolen. Reactive trajectories and the transition path process. *Probability Theory and Related Fields*, 161(1):195–244, 2015.

[202] J. Lu, Z. Shen, H. Yang, and S. Zhang. Deep network approximation for smooth functions. *arXiv preprint arXiv:2001.03040*, 2020.

[203] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang. The expressive power of neural networks: a view from the width. *arXiv preprint arXiv:1709.02540*, 2017.

[204] C. J. Maddison, A. Mnih, and Y. W. Teh. The concrete distribution: a continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017.

[205] V. Maiorov and A. Pinkus. Lower bounds for approximation by MLP neural networks. *Neurocomputing*, 25(1-3):81–91, 1999.

[206] X.-L. Meng and W. H. Wong. Simulating ratios of normalizing constants via a simple identity: a theoretical exploration. *Statistica Sinica*:831–860, 1996.

[207] M. Mider, P. A. Jenkins, M. Pollock, G. O. Roberts, and M. Sørensen. Simulating bridges using confluent diffusions. *arXiv preprint arXiv:1903.10184*, 2019.

[208] G. N. Milstein and M. V. Tretyakov. *Stochastic numerics for mathematical physics*. Springer Science & Business Media, 2013.

[209] F. C. Mitroi. Estimating the normalized Jensen functional. *J. Math. Inequal*, 5(4):507–521, 2011.

[210] A. Mnih and K. Gregor. Neural variational inference and learning in belief networks. In *International Conference on Machine Learning*, 2014.

[211] A. Mnih and D. J. Rezende. Variational inference for Monte Carlo objectives. In *International Conference on Machine Learning*, 2016.

[212] S. Mohamed, M. Rosca, M. Figurnov, and A. Mnih. Monte Carlo gradient estimation in machine learning. *arXiv preprint arXiv:1906.10652*, 2019.

[213] T. Müller, B. McWilliams, F. Rousselle, M. Gross, and J. Novák. Neural importance sampling. *arXiv preprint arXiv:1808.03856*, 2018.

[214] C. Naesseth, F. J. R. Ruiz, S. Linderman, and D. M. Blei. Reparameterization gradients through acceptance-rejection methods. In *Artificial Intelligence and Statistics*, 2017.

[215] C. A. Naesseth, F. Lindsten, and D. M. Blei. Markovian score climbing: variational inference with KL$(p|q)$. *arXiv preprint arXiv:2003.10374*, 2020.

[216] N. Nüsken and L. Richter. Interpolating between BSDEs and PINNs: deep learning for elliptic and parabolic boundary value problems. *arXiv preprint arXiv:2112.03749*, 2021.

[217] N. Nüsken and L. Richter. Solving high-dimensional Hamilton–Jacobi–Bellman PDEs using neural networks: perspectives from the theory of controlled diffusions and measures on path space. *Partial Differential Equations and Applications*, 2(4):1–48, 2021.

[218] A. M. Oberman. Convergent difference schemes for degenerate elliptic and parabolic equations: Hamilton–Jacobi equations and free boundary problems. *SIAM Journal on Numerical Analysis*, 44(2):879–895, 2006.

[219] B. Øksendal. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

[220] I. V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[221] I. V. Oseledets and E. E. Tyrtyshnikov. Breaking the curse of dimensionality, or how to use SVD in many dimensions. *SIAM Journal on Scientific Computing*, 31(5):3744–3759, 2009.

[222] M. Oster, L. Sallandt, and R. Schneider. Approximating the stationary Hamilton-Jacobi-Bellman equation by hierarchical tensor products. *arXiv preprint arXiv:1911.00279*, 2019.

[223] A. Owen and Y. Zhou. Safe and effective importance sampling. *Journal of the American Statistical Association*, 95(449):135–143, 2000.

[224] A. B. Owen. Monte Carlo theory, methods and examples. *Monte Carlo Theory, Methods and Examples. Art Owen*, 2013.

[225] G. Pagès. *Numerical probability: An introduction with applications to finance*. Springer, 2018.

[226] J. W. Paisley, D. M. Blei, and M. I. Jordan. Variational Bayesian inference with stochastic search. In *International Conference on Machine Learning*, 2012.

[227] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *arXiv preprint arXiv:1912.02762*, 2019.

[228] É. Pardoux. Backward stochastic differential equations and viscosity solutions of systems of semilinear parabolic and elliptic PDEs of second order. In *Stochastic Analysis and Related Topics VI*, pages 79–127. Springer, 1998.

[229] E. Pardoux and S. Peng. Adapted solution of a backward stochastic differential equation. *Systems & Control Letters*, 14(1):55–61, 1990.

[230] G. A. Pavliotis. *Stochastic processes and applications: diffusion processes, the Fokker-Planck and Langevin equations*, volume 60. Springer, 2014.

[231] J. W. T. Peters and M. Welling. Probabilistic binary neural networks. *arXiv preprint arXiv:1809.03368*, 2018.

[232] P. Petersen and F. Voigtlaender. Optimal approximation of piecewise smooth functions using deep ReLU neural networks. *Neural Networks*, 108:296–330, 2018.

[233] H. Peyrl, F. Herzog, and H. P. Geering. Numerical solution of the Hamilton-Jacobi-Bellman equation for stochastic optimal control problems. In *Proc. 2005 WSEAS International Conference on Dynamical Systems and Control*, pages 489–497, 2005.

[234] H. Pham. *Continuous-time stochastic control and optimization with financial applications*, volume 61. Springer Science & Business Media, 2009.

[235] H. Pham, H. Pham, and X. Warin. Neural networks-based backward scheme for fully nonlinear PDEs. *arXiv preprint arXiv:1908.00412*, 2019.

[236] A. Pinkus. Approximation theory of the MLP model in neural networks. *Acta numerica*, 8:143–195, 1999.

[237] B. Polyak and P. Shcherbakov. Why does Monte Carlo fail to work properly in high-dimensional optimization problems? *Journal of Optimization Theory and Applications*, 173(2):612–627, 2017.

[238] W. B. Powell. From reinforcement learning to optimal control: a unified framework for sequential decisions. *arXiv preprint arXiv:1912.03513*, 2019.

[239] A. Quarteroni and A. Valli. *Numerical approximation of partial differential equations*, volume 23. Springer Science & Business Media, 2008.

[240] F. Ragone, J. Wouters, and F. Bouchet. Computation of extreme heat waves in climate models using a large deviation algorithm. *Proceedings of the National Academy of Sciences*, 115(1):24–29, 2018. DOI: 10.1073/pnas.1712645115.

[241] M. Raissi. Forward-backward stochastic neural networks: deep learning of high-dimensional partial differential equations. *arXiv preprint arXiv:1804.07010*, 2018.

[242] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[243] R. Ranganath, J. Altosaar, D. Tran, and D. M. Blei. Operator variational inference. In *Advances in Neural Information Processing Systems*, 2016.

[244] R. Ranganath, S. Gerrish, and D. M. Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, 2014.

[245] K. Rawlik, M. Toussaint, and S. Vijayakumar. On stochastic optimal control and reinforcement learning by approximate inference. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[246] S. Reich. Data assimilation: the Schrödinger perspective. *Acta Numerica*, 28:635–711, 2019.

[247] R.-D. Reiss. *Approximate distributions of order statistics: with applications to nonparametric statistics*. Springer science & business media, 2012.

[248] D. Revuz and M. Yor. *Continuous martingales and Brownian motion*, volume 293. Springer Science & Business Media, 2013.

[249] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, 2014.

[250] L. Richter, A. Boustati, N. Nüsken, F. Ruiz, and O. D. Akyildiz. VarGrad: A low-variance gradient estimator for variational inference. *Advances in Neural Information Processing Systems*, 33, 2020.

[251] L. Richter, L. Sallandt, and N. Nüsken. Solving high-dimensional parabolic PDEs using the tensor train format. *International Conference on Machine Learning*, 2021.

[252] H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

[253] C. Robert and G. Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[254] G. M. Rotskoff and E. Vanden-Eijnden. Learning with rare data: using active importance sampling to optimize objectives dominated by rare events. *arXiv preprint arXiv:2008.06334*, 2020.

[255] R. Y. Rubinstein and D. P. Kroese. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media, 2013.

[256] F. J. R. Ruiz and M. K. Titsias. A contrastive divergence for combining variational inference and MCMC. In *International Conference on Machine Learning*, 2019.

[257] F. J. R. Ruiz, M. K. Titsias, and D. M. Blei. The generalized reparameterization gradient. In *Advances in Neural Information Processing Systems*, 2016.

[258] R. Salakhutdinov and I. Murray. On the quantitative analysis of deep belief networks. In *Proceedings of the 25th international conference on Machine learning*, pages 872–879, 2008.

[259] T. Salimans and D. A. Knowles. On using control variates with stochastic approximation for variational Bayes and its connection to stochastic linear regression. *arXiv preprint arXiv:1401.1022*, 2014.

[260] D. Sanz-Alonso. Importance sampling and necessary sample size: an information theory approach. *SIAM/ASA Journal on Uncertainty Quantification*, 6(2):867–879, 2018.

[261] I. Sason. On improved bounds for probability metrics and f-divergences. *arXiv preprint arXiv:1403.7164*, 2014.

[262] I. Sason. Tight bounds for symmetric divergence measures and a new inequality relating f-divergences. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2015.

[263] E. Schrödinger. *Über die Umkehrung der Naturgesetze*. Verlag der Akademie der Wissenschaften in Kommission bei Walter De Gruyter, 1931.

[264] C. Schütte and W. Huisinga. *Biomolecular conformations can be identified as metastable sets of molecular dynamics*. Elsevier, 2003.

[265] C. Schütte and M. Sarich. *Metastability and Markov State Models in Molecular Dynamics*, volume 24. American Mathematical Soc., 2013.

[266] C. Schwab and J. Zech. Deep learning in high dimension: neural network expression rates for generalized polynomial chaos expansions in UQ. *Analysis and Applications*, 17(01):19–55, 2019.

[267] O. Shayer, D. Levi, and E. Fetaya. Learning discrete weights using the local reparameterization trick. In *International Conference on Learning Representations*, 2018.

[268] W. Sickel and T. Ullrich. Tensor products of Sobolev-Besov spaces and applications to approximation from the hyperbolic cross. *Journal of Approximation Theory*, 161(2):748–786, 2009.

[269] A. H. Siddiqi and S. Nanda. *Functional analysis with applications*. Springer, 1986.

[270] D. Siegmund. Importance sampling in the Monte Carlo study of sequential tests. *The Annals of Statistics*:673–684, 1976.

[271] J. Sirignano and K. Spiliopoulos. DGM: a deep learning algorithm for solving partial differential equations. *Journal of computational physics*, 375:1339–1364, 2018.

[272] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson. Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12):4629–4640, 2008.

[273] K. Spiliopoulos. Large deviations and importance sampling for systems of slow-fast motion. *Applied Mathematics & Optimization*, 67(1):123–161, 2013.

[274] K. Spiliopoulos. Nonasymptotic performance analysis of importance sampling schemes for small noise diffusions. *Journal of Applied Probability*, 52(3):797–810, 2015.

[275] G. Stoltz, T. Lelievre, and M. Rousset. *Free energy computations: A mathematical perspective*. World Scientific, 2010.

[276] D. W. Stroock and S. S. Varadhan. *Multidimensional diffusion processes*. Springer, 2007.

[277] A.-S. Sznitman. Topics in propagation of chaos. In *Ecole d'été de probabilités de Saint-Flour XIX—1989*, pages 165–251. Springer, 1991.

[278] A.-S. Sznitman. *Brownian motion, obstacles and random media*. Springer Science & Business Media, 1998.

[279] S. Thijssen and H. Kappen. Path integral control and state-dependent feedback. *Physical Review E*, 91(3):032104, 2015.

[280] M. K. Titsias and M. Lázaro-Gredilla. Doubly stochastic variational Bayes for non-conjugate inference. In *International Conference on Machine Learning*, 2014.

[281] N. Touzi. *Optimal stochastic control, stochastic target problems, and backward SDE*, volume 29. Springer Science & Business Media, 2012.

[282] G. Tucker, A. Mnih, C. J. Maddison, and J. Sohl-Dickstein. REBAR: low-variance, unbiased gradient estimates for discrete latent variable models. In *International Conference on Learning Representations*, 2017.

[283] B. Tzen and M. Raginsky. Neural stochastic differential equations: deep latent Gaussian models in the diffusion limit. *arXiv preprint arXiv:1905.09883*, 2019.

[284] B. Tzen and M. Raginsky. Theoretical guarantees for sampling and inference in generative models with latent diffusions. *arXiv preprint arXiv:1903.01608*, 2019.

[285] A. S. Üstünel and M. Zakai. *Transformation of measure on Wiener space*. Springer Science & Business Media, 2013.

[286] A. S. Üstünel. *An introduction to analysis on Wiener space*. Springer, 2006.

[287] R. van Handel. Probability in high dimension. Technical report, Princeton University, 2014.

[288] R. van der Meer, C. Oosterlee, and A. Borovykh. Optimally weighted loss functions for solving PDEs with neural networks. *arXiv preprint arXiv:2002.06269*, 2020.

[289] R. Van Handel. Stochastic calculus, filtering, and stochastic control. 14, 2007.

[290] E. Vanden-Eijnden and J. Weare. Rare event simulation of small noise diffusions. *Communications on Pure and Applied Mathematics*, 65(12):1770–1803, 2012.

[291] M. J. Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

[292] D. Wang, H. Liu, and Q. Liu. Variational inference with tail-adaptive f-divergence. In *Advances in Neural Information Processing Systems*, 2018.

[293] S. Wang, Y. Teng, and P. Perdikaris. Understanding and mitigating gradient pathologies in physics-informed neural networks. *arXiv preprint arXiv:2001.04536*, 2020.

[294] S. Wang, H. Wang, and P. Perdikaris. On the eigenvector bias of Fourier feature networks: from regression to solving multi-scale PDEs with physics-informed neural networks. *arXiv preprint arXiv:2012.10047*, 2020.

[295] S. Wang, X. Yu, and P. Perdikaris. When and why PINNs fail to train: a neural tangent kernel perspective. *arXiv preprint arXiv:2007.14527*, 2020.

[296] E Weinan and B. Yu. The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1):1–12, 2018.

[297] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992.

[298] E. Wong and B. Hajek. *Stochastic processes in engineering systems*. Springer Science & Business Media, 2012.

[299] M. Xu, M. Quiroz, R. Kohn, and S. A. Sisson. Variance reduction properties of the reparameterization trick. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2711–2720. PMLR, 2019.

[300] J. Yang and H. J. Kushner. A Monte Carlo method for sensitivity analysis and parametric optimization of nonlinear stochastic systems. *SIAM journal on control and optimization*, 29(5):1216–1249, 1991.

[301] D. Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94:103–114, 2017.

[302] M. Yin, Y. Yue, and M. Zhou. ARSM: augment-REINFORCE-swap-merge estimator for gradient back-propagation through categorical variables. In *International Conference on Machine Learning*, 2019.

[303] M. Yin and M. Zhou. ARM: augment-REINFORCE-merge gradient for stochastic binary networks. In *International Conference on Learning Representations*, 2019.

[304] J. Yong and X. Y. Zhou. *Stochastic controls: Hamiltonian systems and HJB equations*, volume 43. Springer Science & Business Media, 1999.

[305] C. Zhang, J. Bütepage, H. Kjellström, and S. Mandt. Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):2008–2026, 2018.

[306] J. Zhang. A numerical scheme for BSDEs. *The annals of applied probability*, 14(1):459–488, 2004.

[307] J. Zhang. *Backward stochastic differential equations*. Springer, 2017.

[308] W. Zhang, H. Wang, C. Hartmann, M. Weber, and C. Schütte. Applications of the cross-entropy method to importance sampling and optimal control of diffusions. *SIAM Journal on Scientific Computing*, 36(6):A2654–A2672, 2014.