arXiv:2306.00637v1 [cs.CV] 1 Jun 2023

# Würstchen: Efficient Pretraining of Text-to-Image Models

**Pablo Pernias**[*]
Independent Researcher
Sant Joan d'Alacant, Spain
pablo@pernias.com

**Dominic Rampas**[*]
Technische Hochschule Ingolstadt
Ingolstadt, Germany
and
Wand Technologies Inc.
New York, USA
dominic.rampas@gmail.com

**Marc Aubreville**
Technische Hochschule Ingolstadt
Ingolstadt, Germany
marc.aubreville@thi.de

## Abstract

We introduce Würstchen, a novel technique for text-to-image synthesis that unites competitive performance with unprecedented cost-effectiveness and ease of training on constrained hardware. Building on recent advancements in machine learning, our approach, which utilizes latent diffusion strategies at strong latent image compression rates, significantly reduces the computational burden, typically associated with state-of-the-art models, while preserving, if not enhancing, the quality of generated images. Würstchen achieves notable speed improvements at inference time, thereby rendering real-time applications more viable. One of the key advantages of our method lies in its modest training requirements of only 9,200 GPU hours, slashing the usual costs significantly without compromising the end performance. In a comparison against the state-of-the-art, we found the approach to yield strong competitiveness. This paper opens the door to a new line of research that prioritizes both performance and computational accessibility, hence democratizing the use of sophisticated AI technologies. Through Würstchen, we demonstrate a compelling stride forward in the realm of text-to-image synthesis, offering an innovative path to explore in future research.

## 1 Introduction

State-of-the-art diffusion models [Ho et al., 2020, Saharia et al., 2022, Ramesh et al., 2022] have advanced the field of image synthesis considerably, achieving remarkable results that closely approximate photorealism. However, these foundation models, while impressive in their capabilities, carry a significant drawback: they are computationally demanding. For instance, Stable Diffusion 1.4, one of the most notable models in the field, used 150,000 GPU hours for training. Against this backdrop, we propose a novel approach, named "Würstchen", which drastically reduces the computational demands while maintaining competitive performance. Our method is based on a novel architecture that elegantly distributes the task of image synthesis across three distinct stages, thereby making the learning process more manageable and computationally efficient.

[*]Equal contributors

Preprint. Under review.


Figure 1: Text-conditional generations using Würstchen.

The approach uses three distinct stages for image synthesis (see Figure 2): initially, a text-conditional latent diffusion model is used to create a latent image of reduced resolution (Stage C), which is then decoded by another model into a vector-quantized latent space of higher resolution (Stage B). Finally, the quantized latent image is decoded to yield the full-resolution output image (Stage A).

Training is performed in reverse order to the inference (Figure 3): The initial training is carried out on Stage A and employs a Vector-quantized Generative Adversarial Network (VQGAN) to create a discretized latent space. As shown in earlier work, this compact representation facilitates learning and inference speed [Rombach et al., 2022, Chang et al., 2023, Rampas et al., 2023]. In the next phase Stage B is trained, which acts as a further compression stage, employing an encoder that projects images into an even more condensed space and a decoder that tries to reconstruct VQGAN latents from the encoded image. We employ a token predictor based on the Paella [Rampas et al., 2023] model for this task, which is conditioned on the representation of the encoded image, as it comes with the benefits of a low required number of sampling steps (which is especially beneficial to computational efficiency due to the comparatively highly resolved latent space) [Rampas et al., 2023], simple implementation and training. Finally, for the construction of Stage C, the aforementioned image encoder is employed to project images into the condensed latent space where a text-conditional latent diffusion model [Rombach et al., 2022] is trained. The significant reduction in space dimensions in Stage C allows for more efficient training of the diffusion model, considerably reducing both the computational resources required and the time taken for the process.

Our proposed Würstchen model thus introduces a thoughtfully designed approach to address the high computational burden of current state-of-the-art models, providing a significant leap forward in text-to-image synthesis. With this approach we are able to train a 1B parameter Stage C text-conditional diffusion model within approximately 9,200 GPU hours, resembling a 16x reduction in computation compared to the amount Stable Diffusion 1.4 used for training (150,000 GPU hours), while showing similar fidelity both visually and numerically. Throughout this paper, we provide a comprehensive evaluation of Würstchen's efficacy, demonstrating its potential to democratize the deployment & training of high-quality image synthesis models.

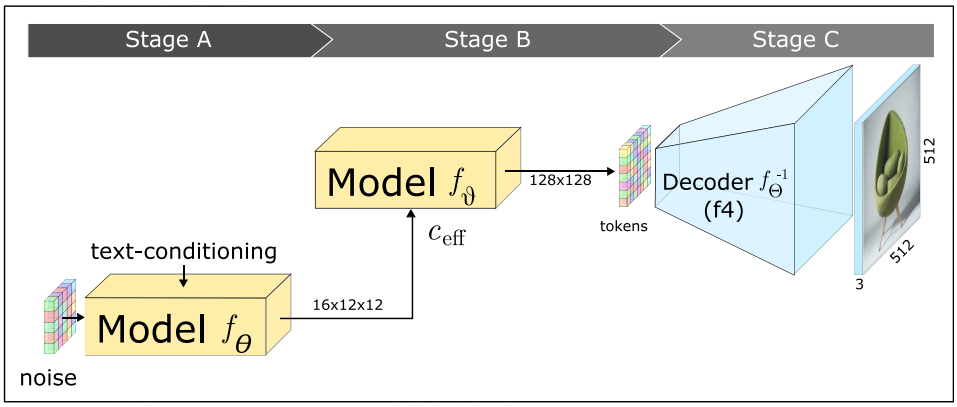


Figure 2: Inference architecture for text-conditional image generation.

Our main contributions are the following:

1. We propose a novel architecture for text-to-image synthesis that substantially reduces computational demands while achieving state-of-the-art performance. This approach introduces an efficient pipeline following a three-stage paradigm, namely a text-conditioned diffusion model (Stage C), an image encoder/decoder (Stage B), and a VQGAN (Stage A).
2. Our architecture enables the training of a 1B parameter Stage C diffusion model with a significantly reduced compute budget. This level of efficiency is achieved without sacrificing the quality of the synthesized images.
3. We provide comprehensive experimental validation of the model's efficacy, opening the door to further research in the field of efficient, high-quality generative models in presenting a compelling paradigm that simultaneously prioritizes both performance and computational feasibility.

4. We are publicly releasing the source code and the entire suite of model weights.

## 2 Related Work

### 2.1 Conditional Image Generation

The field of image generation guided by text prompts has undergone significant progression in recent years. Initial approaches predominantly leveraged Generative Adversarial Networks (GANs) [Reed et al., 2016, Zhang et al., 2017]. More recently, however, a paradigm shift in the field of image generation towards diffusion models [Sohl-Dickstein et al., 2015, Ho et al., 2020] has occurred. These approaches, in some cases, have not only met but even exceeded the performance of GANs in both conditional and unconditional image generation [Dhariwal and Nichol, 2021]. Diffusion models put forth a score-based scheme that gradually eliminates pertubations (e.g., noise) from a target image, with the training objective framed as a reweighted variational lower-bound. Next to diffusion models, another dominant choice for training text-to-image models are transformers. In their early stages, transformer-based models utilized an autoregressive approach, leading to a significant slowdown in inference due to the requirement for each token to be sampled individually. Current strategies, however, employ a bidirectional transformer [Ding et al., 2022, Chang et al., 2022, Chang et al., 2023] to address the challenges that traditional autoregressive models present. As a result, image generation can be executed using fewer steps, while also benefiting from a global context during the generative phase. Other recent work has shown that convolution-based approaches for image generation can yield similar results [Rampas et al., 2023].

### 2.2 Compressed Latent Spaces

The majority of approaches in the visual modality of generative models use some way to train at a smaller space, followed by upscaling to high resolutions, as training at large pixel resolutions can become exponentially more expensive with the size of images. For text-conditional image generation, there are two established categories of approaches: encoder-based and upsampler-based. Latent diffusion models [Rombach et al., 2022], DALL-E [Ramesh et al., 2021], CogView [Ding et al., 2021, Ding et al., 2022], MUSE [Chang et al., 2023] belong to the first category and employ a two-stage training process. Initially, an autoencoder [Rumelhart et al., 1985] is trained to provide a lower-dimensional, yet perceptually equivalent, representation of the data. This representation forms the basis for the subsequent training of a diffusion or a transformer model. Eventually, generated latent representations can be decoded with the decoder branch of the autoencoder to the pixel space. The result is a significant reduction in computational complexity for the diffusion / sampling process and efficient image decoding from the latent space using a single network pass. On the contrary, upsampler-based methods generate images at low resolution in the pixel space and use subsequent models for upscaling the images to higher resolution. UnClip [Ramesh et al., 2022] and Imagen [Saharia et al., 2022] both generate images at 64x64 and upscale using two models to 256 and 1024 pixels. The former model is the largest in terms of parameter count, while the latter models are smaller due to working at higher resolution and only being responsible for upscaling.

### 2.3 Conditional Guidance

The conditional guidance of models in text-based scenarios is typically facilitated through the encoding of textual prompts via a pretrained language model. Two major categories of text encoders are prevalently employed: contrastive text encoders and uni-modal text encoders. Contrastive Language-Image Pretraining (CLIP) [Radford et al., 2021] is a representative of the contrastive multimodal models that strives to align text descriptions and images bearing semantic resemblance within a common latent space. A host of image generation methodologies have adopted a frozen CLIP model as their exclusive conditioning method in recent literature. The hierarchical DALL-E 2 by Ramesh *et al*. [Ramesh et al., 2022] specifically harnesses CLIP image embeddings as input for their diffusion model, while a 'prior' performs the conversion of CLIP text embeddings to image embeddings. Stable Diffusion [Rombach et al., 2022], on the other hand, makes use of un-pooled CLIP text embeddings to condition its latent diffusion model. In contrast, the works of Saharia *et al*. [Saharia et al., 2022], Liu *et al*. [Liu et al., 2022] and Chang *et al*. [Chang et al., 2023] leverage a large, uni-modal language model such as T5 [Raffel et al., 2020] or ByT5 [Xue et al., 2022] that

can encode textual prompts with notable accuracy, leading to image generations of superior precision in terms of composition, style, and layout.

# 3 Method

Our method comprises three stages, all implemented as deep neural networks. For image generation, we first generate a latent image at a strong compression ratio using a text-conditional latent diffusion model (Stage C). Subsequently, this representation is transformed to an upsampled and quantized latent space by the means of a secondary model which is tasked for this reconstruction (Stage B). Finally, the tokens that comprise the latent image in this intermediate resolution are decoded to yield the output image (Stage A). The training of this architecture is performed in reverse order, starting with Stage A, then following up with Stage B and finally Stage C (see Figure 3).

## 3.1 Stage A and B

It is a known and well-studied technique to reduce the computational burden by compressing data into a smaller representation[Rombach et al., 2022, Chang et al., 2022]. Our approach follows this paradigm, too, and makes use of Stage A & B to achieve a notably higher compression than usual. Let $H \times W \times C$ be the dimensions of images. A spatial compression maps images to a latent representation with a resolution of $h \times w \times z$ with $h = H/f, w = W/f$, where $f$ defines the compression rate. Common approaches for modelling image synthesis use a one-stage compression between f4 and f16 [Esser et al., 2021, Chang et al., 2023, Rombach et al., 2022], with higher factors usually resulting in worse reconstructions. Our Stage A consists of a f4 VQGAN [Esser et al., 2021] with parameters $\Theta$ and initially encodes images $\mathbf{x} \in \mathbb{R}^{3\times512\times512}$ into $128 \times 128$ discrete tokens from a learnt codebook of size 8,192.

$$\mathbf{x}_q = f_\Theta(\mathbf{x})$$

The network is trained as described by Esser *et al*. and tries to reconstruct the image based on the quantized latents, so that:

$$f_\Theta^{-1}\left(f_\Theta\left(\mathbf{x}\right)\right) = f_\Theta^{-1}\left(\mathbf{x}_q\right) \approx \mathbf{x}$$

where $f_\Theta^{-1}$ resembles the decoder part of the VQGAN.

Afterwards, Stage B is learnt in the compressed and discrete VQGAN space to reconstruct images that were encoded with an additional model which utilizes an inherent higher compression ratio (see Figure 3). We make use of the large (L) configuration of an EfficientNet2 stem [Tan and Le, 2020] to encode images, and task the Stage B model to reconstruct the representation of the same image in the VQGAN space of Stage A. The EfficientNet2 $e_\phi$ takes in images $x \in \mathbb{R}^{3\times384\times384}$ and embeds them into a space of $\mathbb{R}^{1280\times12\times12}$.

We use simple bicubic interpolation for the resizing of the images from $512\times512$ to $384\times384$. On top of that representation, we add a $1 \times 1$ convolutional head that normalizes and projects the embeddings to $c_{\text{eff}} \in \mathbb{R}^{16\times12\times12}$. This compressed representation of the images is given to the Stage B decoder as conditioning to guide the decoding process. We formulated this learning process in a typical noising/denoising framework and decided to use the architecture of Paella [Rampas et al., 2023] for that. The approach works on quantized tokens and is hence perfectly suitable for this task. Image tokens $\mathbf{x}_q$ are noised by random token replacement with other tokens from the VQGAN codebook based on random timesteps. The noised representation $\tilde{\mathbf{x}}_{q,t}$, together with the EfficientNet embeddings $\mathbf{c}_{\text{eff}}$, text conditioning $c_{\text{text}}$ and the timestep $t$ are given to the model.

$$\bar{\mathbf{x}}_{q,0} = f_\vartheta(\tilde{\mathbf{x}}_{q,t}, \mathbf{c}_{\text{eff}}, \mathbf{c}_{\text{text}}, t)$$

Its task is to predict the original tokens. Sampling is executed in an iterative fashion given new EfficientNet embeddings. After training, images $\mathbf{x} \in \mathbb{R}^{3\times512\times512}$ can be decoded from a latent space of $\mathbb{R}^{16\times12\times12}$, resulting in a total spatial compression of **f42**.

Figure 4 shows depictions of images and their corresponding reconstructions. Because the EfficientNet encoder was trained on ImageNet data, which does not capture the broad distribution of images present in large text-image datasets, the model is initialized from a pretrained checkpoint, but also updated during the training of Stage B. We use Cross-Attention [Vaswani et al., 2017] for conditioning and project both $\mathbf{c}_{\text{eff}}$ (flattened) and $\mathbf{c}_{\text{text}}$ to the same dimension in each block of the

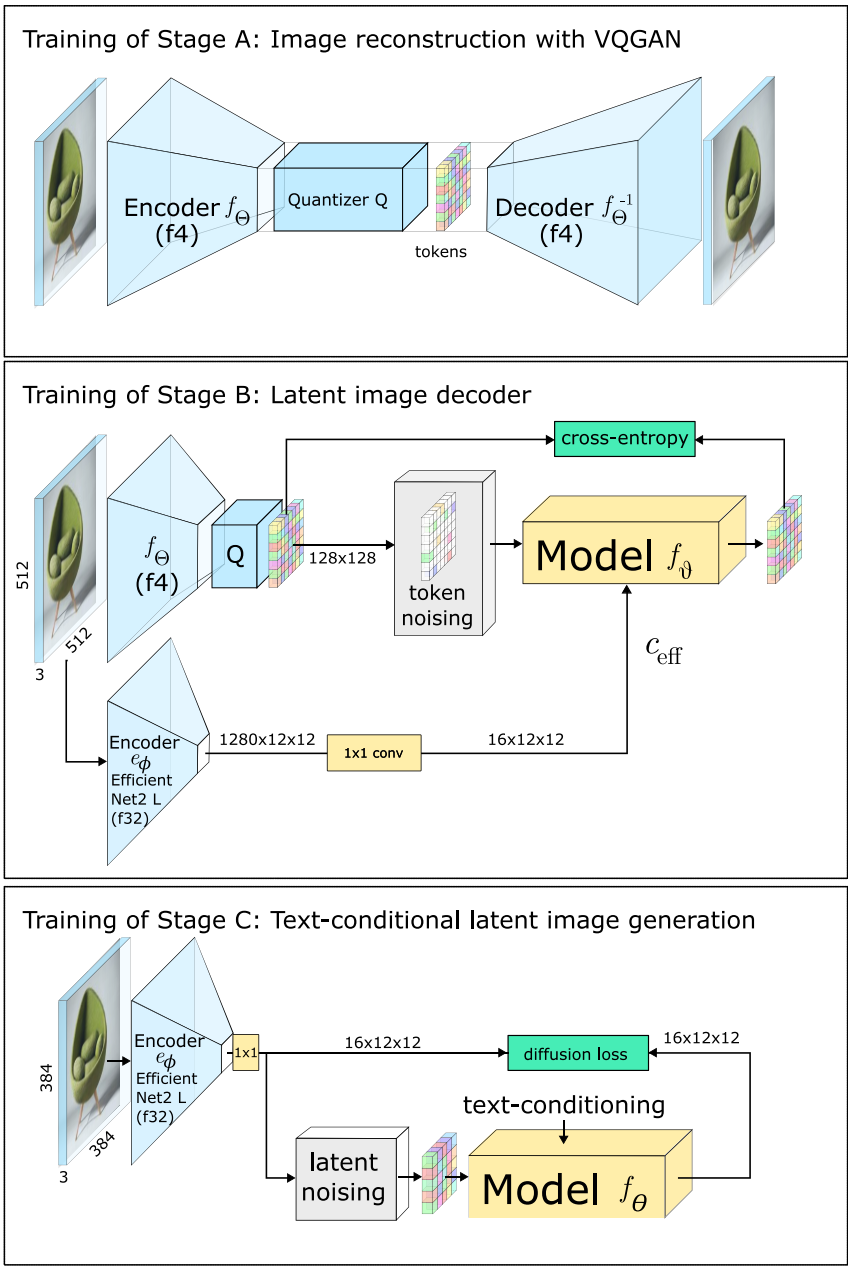


Figure 3: Training objectives of our model. Initially a VQGAN-based autoencoder is trained. Secondly, Stage B is trained as a latent image decoder, decoding an EfficientNet latent image to the original VQGAN latent space. Finally, Stage C is trained as a text-conditional latent diffusion model at a compression rate of f42.

model and concatenate them. We refer to [Rampas et al., 2023] for more details on the training and sampling. Furthermore, during training Stage B, we intermittently add noise to the EfficientNet embeddings, to teach the model to understand non-perfect embeddings, which is likely to be the case when generating these embeddings with Stage C. Lastly, we also randomly drop $\mathbf{c}_{\text{eff}}$ and $\mathbf{c}_{\text{text}}$ to be able to sample with classifier-free-guidance [Ho and Salimans, 2022] during sampling.

### 3.2 Stage C

After Stage A and Stage B are trained, training of the text-conditional last stage can be started. We follow a standard diffusion process, applied in the latent space of the finetuned EfficientNet encoder. Images are encoded into their latent representation $\mathbf{x}_{\text{eff}} = \mathbf{c}_{\text{eff}}$, which now become the target, instead of the conditioning. The latents are noised by using the following forward diffusion formula:

$$\mathbf{x}_{\text{eff},t} = \sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_{\text{eff}} + \sqrt{1 - \bar{\alpha}_t} \cdot \epsilon$$

where $\epsilon$ represents noise from a zero mean unit variance normal distribution. We use a cosine schedule [Nichol and Dhariwal, 2021] to generate $\bar{\alpha}_t$ and use continuous timesteps. The diffusion model takes in the noised embeddings $\mathbf{x}_{\text{eff},t}$, the text conditioning $\mathbf{c}_{\text{text}}$ and the timestep $t$. The model returns the prediction for the noise in the following form:

$$\bar{\epsilon} = \frac{\mathbf{x}_{\text{eff},t} - \mathbf{a}}{\mid 1 - \mathbf{b} \mid + 1e^{-5}}$$

where $\mathbf{a}$ and $\mathbf{b}$ result from:

$$\mathbf{a}, \mathbf{b} = f_\theta(\mathbf{x}_{\text{eff},t}, \mathbf{c}_{\text{text}}, t)$$

We decided to formulate the objective as such, since it made the training more stable. We hypothesize this occurs because the model parameters are initialized to predict $\mathbf{0}$ at the beginning, enlarging the difference to timesteps with a lot of noise. By reformulating to the $\mathbf{a}$ & $\mathbf{b}$ objective, the model initially returns the input, making the loss small for very noised inputs. We use the standard mean-squared-error loss between the predicted noise and the ground truth noise. Additionally, we employ the p2 loss weighting [Choi et al., 2022]:

$$p_2(t) \cdot \mid\mid \epsilon - \bar{\epsilon} \mid\mid^2$$

where $p_2(t)$ is defined as $\frac{1-\bar{\alpha}_t}{1+\bar{\alpha}_t}$, making higher noise levels contribute more to the loss. Text conditioning $\mathbf{c}_{\text{text}}$ are dropped randomly for 5% of the time and replaced with a null-label in order to use classifier-free-guidance [Ho and Salimans, 2022]

### 3.3 Image Generation (Sampling)

Sampling starts at Stage C from initial random noise $\mathbf{x}_{\text{eff},T_C} = \mathcal{N}(0, \mathbf{I})$. We use the DDPM [Ho et al., 2020] algorithm to sample the EfficientNet latents conditioned on text-embeddings. To do so, we run the following operation for $T_C$ steps:

$$\hat{\mathbf{x}}_{\text{eff},t-1} = \frac{1}{\sqrt{\alpha_t}} \cdot (\hat{\mathbf{x}}_{\text{eff},t} - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \bar{\epsilon}) + \sqrt{(1 - \alpha_t) \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t}} \epsilon$$

We denote the outcome as $\bar{\mathbf{x}}_{\text{eff}}$ which is of shape $16 \times 12 \times 12$. This output is flattened to a shape of $144 \times 16$ and given as conditioning, along with the same text embeddings used to sample $\bar{\mathbf{x}}_{\text{eff}}$, to Stage B. This Stage operates at the $128 \times 128$ VQGAN latent space. We initialize $\mathbf{x}_{q,T_B}$ to random tokens drawn from the VQGAN codebook. We sample $\bar{\mathbf{x}}_q$ by iteratively predicting all tokens for $T_B$ steps.

$$\mathbf{x}_{q,t-1} = f_\vartheta(\mathbf{x}_{q,t}, \mathbf{c}_{\text{eff}}, \mathbf{c}_{\text{text}}, t)$$

and subsequently renoising a ratio of the tokens back to their original noise. Finally $\bar{\mathbf{x}}_q$ will be projected back to the pixel space using the decoder $f_\Theta^{-1}$ of the VQGAN (Stage A):

$$\bar{\mathbf{x}} = f_\Theta^{-1}(\bar{\mathbf{x}}_q)$$

A depiction of the sampling pipeline can be seen in Figure 2.

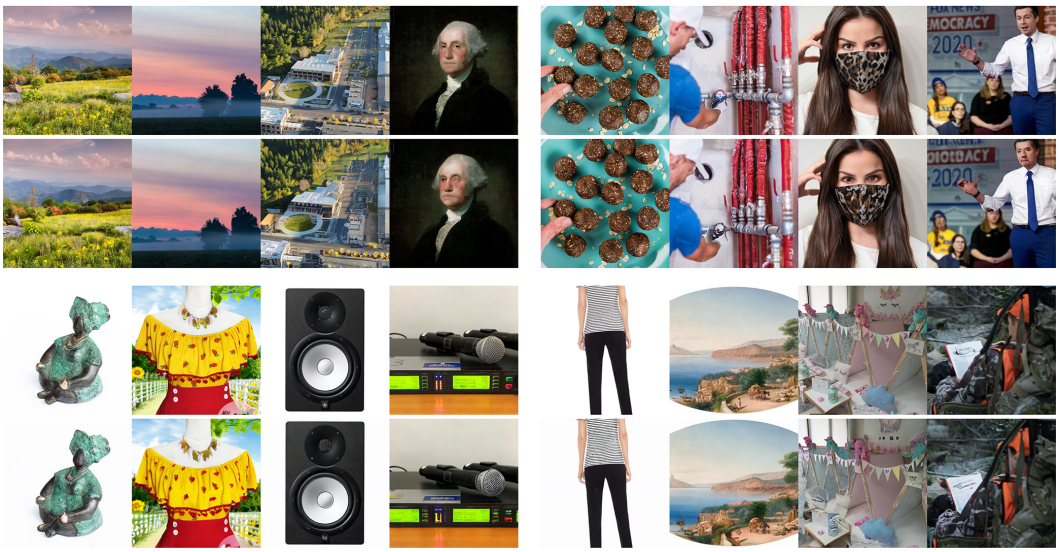

Figure 4: Reconstruction samples using Stage B using a total compression factor of f42.

### 3.4 Model Decisions

Many choices were required when setting up and training the different stages. One of the most important decision had to be made about the image encoder. Theoretically, any visual model could be used for that, but three things should be kept in mind: the training objective the model was trained with, the parameter count and the embedding dimension. We hypothesize that it is beneficial to use an encoder that already has a good feature representation of a wide variety of images. Furthermore, having a small and parameter efficient model makes training of Stage B & C faster. Finally, the feature dimension of the encoder network is vital. If it is excessively small, it may fail to capture sufficient image details; conversely, if it is overly large, it may unnecessarily increase computational requirements and extend training duration. Moreover, the type of model for Stage A & B also resembles a choice to be made. We decided to use the architecture of Paella for Stage B, due to its ability to handle quantized data and requiring very little number of inference steps to sample images [Rampas et al., 2023]. The latter attribute is crucial to maintain a low-latency pipeline, as sampling at a resolution of $128 \times 128$ could be computationally demanding if many steps are necessary. However, in theory a diffusion model could be used, too. A different architecture is needed for Stage C since it requires a model capable of handling continuous data, unlike the one used for Stage B. Hence we decided to use a latent diffusion model [Rombach et al., 2022]. While diffusion models require more inference steps, this demand is rendered more feasible within the context of a denser latent space.

## 4 Experiments

### 4.1 Text-to-Image Training

To demonstrate Würstchen's capabilities on text-to-image generation, we trained a 18M parameter Stage A, a 600M parameter Stage B and a 1B parameter Stage C. We employed an EfficientNet2-Large stem [Tan and Le, 2020] in the training. Stage B and C are both conditioned on un-pooled CLIP-H [Ilharco et al., 2021] text-embeddings. All models are optimized using AdamW [Loshchilov and Hutter, 2019] with a learning rate of $1e^{-4}$ using a linear warm-up schedule for 10k steps. Stage B & C were trained for 0.25M and 0.8M steps using a batch size of 384 and 1280, respectively. All stages were trained on subsets of the improved-asesthetic LAION-5B [Schuhmann et al., 2022] dataset.

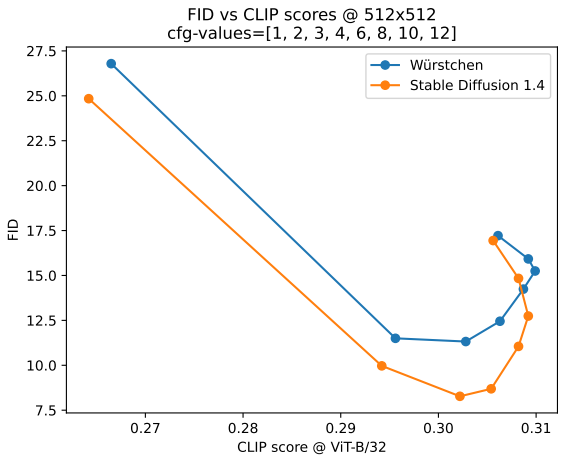


Figure 5: Pareto curves for FID and CLIP scores comparing Würstchen to Stable Diffusion 1.4. We observe the two models to be on par in terms of the CLIP score, but Stable Diffusion 1.4 achieving higher fidelity for the COCO dataset. We hypothesize the inferior performance on FID to be highly affected by Stage B, as reconstructions lack details.

### 4.2 Text-to-Image Evaluation

Evaluations of text-to-image models in both supervised and zero-shot settings commonly use the COCO 2014 [Chen et al., 2015] validation set as a reference benchmark [Rombach et al., 2022, Saharia et al., 2022, Chang et al., 2023, Ramesh et al., 2022]. The primary automated metrics employed for performance assessment are the Frechet Inception Distance (FID) [Heusel et al., 2017], which quantifies image fidelity, and the CLIP score [Hessel et al., 2022, Radford et al., 2021], which determines alignment between image and text. In line with prior studies, we provide the FID-30k metric in a zero-shot context, which involves randomly selecting 30K prompts and image pairs from the validation set and comparing the model's generated samples based on these prompts with the reference images from the validation set in the latent space of an independend third model (Inception V3 trained on ImageNet). The same generated images will be used to calculate the CLIP score with the captions. The results can be seen in Table 1. All of the experiments use the standard DDPM [Ho et al., 2020] algorithm to sample latents in Stage C. Stage B uses the sampling as described in [Rampas et al., 2023]. Both stages also make use of classifier-free-guidance [Ho and Salimans, 2022] with guidance scale $w$. We fix the hyperparameters for Stage B sampling to $T_B = 8$ and $w = 2$. To find good sampling parameters for Stage C, we evaluate FID & CLIP score for different classifier-free-guidance weights $w$. We choose to fix $T_C = 60$. Furthermore, we also compare to the most similar model, Stable Diffusion 1.4, in terms of trainable parameters and conditioning. The results are shown in Figure 5 as pareto curves for the COCO [Chen et al., 2015] dataset. We observe similar results for the CLIP scores in both models, however, slightly worse results for FID values. We hypothesize this is partly caused by artifacts and innacuracies produced by the image reconstruction of Stage B. In an attempt to validate this hypothesis, we computed the FID scores between original COCO images and reconstructed images using Stage B only, which gave a score of $\text{FID} = 5.73$, thus highlighting the fact that, as we believed, the quality of the reconstructions is indeed a significant contributor to the FID score, and a clear target for improvement. Furthermore, Figure A.1-A.5 show visual comparisons between Würstchen and Stable Diffusion 1.4. All prompts are non-cherrypicked and all generations use the same seed. The prompts represent a diverse subset of the dalle-mini prompts [Dayma et al., 2021]. Visually, we observe similar fidelity and prompt-alignment and find both models to be on par.

### 4.3 Computational Requirements

Table 1 shows the computational costs for training Würstchen compared to the original StableDiffusion 1.4. Based on the evaluations in Section 4.2, it can be seen that the proposed setup of decoupling high-resolution image projection from the actual text-conditional generation can be leveraged even

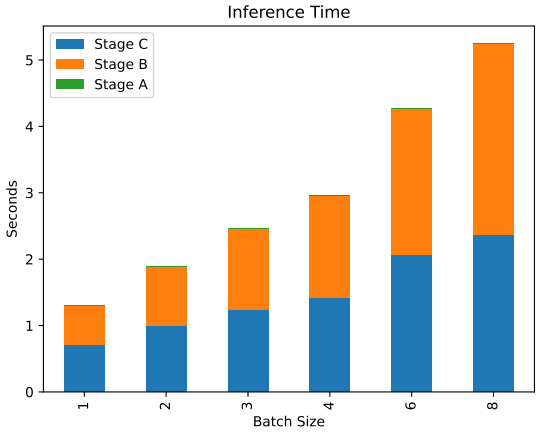


Figure 6: Optimized sampling speeds for different batch sizes.

| Model | Parameters | Sampling Steps | FID-COCO-30k ↓ 256px | FID-COCO-30k ↓ 512px | open source | GPU hours @ A100 |
|---|---|---|---|---|---|---|
| CogView [Ramesh et al., 2021] | 4B | 1024 | 27.1 | | ✓ | – |
| DALL-E [Ramesh et al., 2021] | 12B | 256 | 17.89 | | – | – |
| LDM [Rombach et al., 2022] | 1.45B | 250 | 12.63 | | ✓ | - - |
| GLIDE [Nichol et al., 2021] | 3.5B | 250 | 12.24 | | – | – |
| Make-A-Scene [Gafni et al., 2022] | 4B | 1024 | 11.84 | | – | – |
| Paella [Rampas et al., 2023] | 1B | **12** | 11.07 | | ✓ | 64,000 |
| DALL-E 2 [Ramesh et al., 2022] | 3.5B | 250 | 10.39 | | – | – |
| MUSE-3B [Chang et al., 2023] | 3B | 24 | 7.78 | | – | – |
| Imagen [Saharia et al., 2022] | 2B | 1000 | 7.27 | | – | – |
| Parti [Yu et al., 2022] | 20B | 1024 | **7.23** | | – | – |
| Würstchen (proposed) | 0.99B | 60 | | 11.32 | ✓ | **9,200** |
| Stable Diffusion 1.4 [Rombach et al., 2022] | 0.8B | 50 | | **8.27*** | ✓ | 150,000 |

Table 1: Comparison of the zero-shot Fréchet Inception Distance to other state-of-the-art text-to-image methods on $256 \times 256$ and $512 \times 512$ image resolutions. * own evaluation

more as done in the past [Esser et al., 2021, Saharia et al., 2022, Ramesh et al., 2022], while still staying on-par in terms of quality, fidelity and alignment. Stage C, being the most expensive stage to train, required only 9,200 GPU hours, compared to 150,000 GPU hours[2] for StableDiffusion 1.4, making it a 16x improvement. Moreover, although needing to sample with both Stage A & B to generate the VQGAN latents $\bar{\mathbf{x}}_q$, the total inference is still very fast. Figure 6 shows sampling times for different batch sizes.

## 5 Discussion

Several constraints persist in our current experimental setup. A principal issue arises when the model attempts to sample images in Stage B at resolutions higher than its training capacity, leading to the generation of repetitive patterns. This could potentially result from the model's inability to interpret larger images appropriately. We attribute this issue to the conditioning mechanism, where the EfficientNet embeddings are injected via cross-attention, causing them to be flattened, thereby losing their two-dimensional positional bias. This might account for the model's difficulties in generalizing to varying resolutions during inference. Figure 7 provides examples of sampling at different resolutions. Furthermore, the current design of Würstchen suffers from common limitations that are characteristic of models conditioned solely on CLIP text-embeddings, such as StableDiffusion [Rombach et al., 2022], which includes challenges in rendering text and compositional difficulties with complex scenes. However, the relatively inexpensive computational demands of this model open up possibilities for iterating on the model design at faster pace. Moreover, training of Stage B using the current architecture turned out to be difficult to train due to instabilities. As a result,

[2]As reported in the model card at `https://huggingface.co/CompVis/stable-diffusion-v-1-4-original`

Figure 7: Failure cases of Stage B: Decoding at resolutions unseen during training (in this case: 512×768) represents a great challenge for the model and results in repetitive patterns. We hypothesize the reason can be found in the conditioning mechanism used to inject the EfficientNet embeddings.

we had to stop training early. After training Stage B for 250k steps, the model already performed good for the amount of compression it has to decode, however still shows flaws in finer details of images. Figure 4 shows examples. We leave this open for future work to iterate on the stability of the training mechanism. On the other hand, training Stage C behaved significantly more stable and did not encounter issues during training or sampling.

We anticipate that Stage B could see significant enhancements in terms of image reconstruction quality and its ability to handle images beyond the training resolution. Adjustments may be made within the conditioning mechanism, with positional embeddings on the cross-attention potentially improving the aforementioned issue. Alternative conditioning mechanisms, such as Modulated / Adaptive Layer Normalization [Chen et al., 2019, Perez et al., 2017] or simple concatenation, could also prove effective. Moreover, if the EfficientNet latents could be quantized, the possibility of implementing a sampling mechanism working on quantized latent spaces (such as [Chang et al., 2023, Rampas et al., 2023]) for Stage A would arise, which could further reduce computational demands. Conversely, training Stage B using Latent Diffusion Models [Rombach et al., 2022] might also be increased in efficiency by minimizing the number of inference steps by approaches like distillation or consistency models [Song et al., 2023]. It should be noted that the current design of Stage B also functions as an upsampler, encoding $384 \times 384$ images via EfficientNet and decoding into $512 \times 512$ images. It is conceivable that this ratio might be increased, allowing Stage B to serve as both a decoder and upsampler. Furthermore, considerable effort has been dedicated to enhancing the efficiency of text-to-image model training through tactics such as pre-calculating embeddings and using lower precision number formats. Other than mixed precision training [Micikevicius et al., 2018], we have not implemented any specific accelerative strategies, suggesting the potential for even greater computational efficiency and reduced resource requirements. Finally, the paradigm of further decoupling large-scale conditional training from high-resolution constraints could also be applied to the field of conditional video generation. Such an approach could yield even more significant accelerations in training & processing speed than for images.

## 6 Conclusion

In this work we presented our text-conditional image generation model Würstchen, which employs a three stage process of decoupling text-conditional image generation from high resolution spaces. The

proposed process enables to train large scale models efficiently, substantially reducing computational requirements, while at the same time providing high-fidelity images. Our trained model achieved comparable performance to models trained using significantly more computational resources, illustrating the viability of this approach and suggesting potential efficient scalability to even larger model parameters. We hope our work can serve as a starting point for further research into a more sustainable and computationally more efficient domain of generative AI and open up more possibilities into training, finetuning & deploying large-scale models on consumer hardware. We provide all of our source code, including training-, and inference scripts and trained models on GitHub [3].

## Acknowledgements

The authors wish to express their thanks to Stability AI Inc. for providing generous computational resources for our experiments and LAION gemeinnütziger e.V. for dataset access and support.

## References

[Chang et al., 2023] Chang, H. et al. (2023). Muse: Text-to-image generation via masked generative transformers. *arXiv:2301.00704*.

[Chang et al., 2022] Chang, H., Zhang, H., Jiang, L., Liu, C., and Freeman, W. T. (2022). MaskGIT: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325.

[Chen et al., 2019] Chen, T., Lucic, M., Houlsby, N., and Gelly, S. (2019). On self modulation for generative adversarial networks.

[Chen et al., 2015] Chen, X., Fang, H., Lin, T.-Y., Vedantam, R., Gupta, S., Dollár, P., and Zitnick, C. L. (2015). Microsoft COCO captions: Data collection and evaluation server. *arXiv:1504.00325*.

[Choi et al., 2022] Choi, J., Lee, J., Shin, C., Kim, S., Kim, H., and Yoon, S. (2022). Perception prioritized training of diffusion models.

[Dayma et al., 2021] Dayma, B., Patil, S., Cuenca, P., Saifullah, K., Abraham, T., Lê Khac, P., Melas, L., and Ghosh, R. (2021). Dall·e mini.

[Dhariwal and Nichol, 2021] Dhariwal, P. and Nichol, A. (2021). Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794.

[Ding et al., 2021] Ding, M., Yang, Z., Hong, W., Zheng, W., Zhou, C., Yin, D., Lin, J., Zou, X., Shao, Z., Yang, H., et al. (2021). Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34:19822–19835.

[Ding et al., 2022] Ding, M., Zheng, W., Hong, W., and Tang, J. (2022). Cogview2: Faster and better text-to-image generation via hierarchical transformers. *arXiv:2204.14217*.

[Esser et al., 2021] Esser, P., Rombach, R., and Ommer, B. (2021). Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12873–12883.

[Gafni et al., 2022] Gafni, O., Polyak, A., Ashual, O., Sheynin, S., Parikh, D., and Taigman, Y. (2022). Make-a-scene: Scene-based text-to-image generation with human priors. *arXiv:2203.13131*.

[Hessel et al., 2022] Hessel, J., Holtzman, A., Forbes, M., Bras, R. L., and Choi, Y. (2022). Clipscore: A reference-free evaluation metric for image captioning.

[Heusel et al., 2017] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). GANs trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, 30.

[Ho et al., 2020] Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851.

[Ho and Salimans, 2022] Ho, J. and Salimans, T. (2022). Classifier-free diffusion guidance. *arXiv:2207.12598*.

[Ilharco et al., 2021] Ilharco, G., Wortsman, M., Carlini, N., Taori, R., Dave, A., Shankar, V., Namkoong, H., Miller, J., Hajishirzi, H., Farhadi, A., and Schmidt, L. (2021). *OpenCLIP*. Zenodo.

[Liu et al., 2022] Liu, R., Garrette, D., Saharia, C., Chan, W., Roberts, A., Narang, S., Blok, I., Mical, R., Norouzi, M., and Constant, N. (2022). Character-aware models improve visual text rendering. *arXiv:2212.10562*.

[3] https://github.com/dome272/wuerstchen

[Loshchilov and Hutter, 2019] Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. *International Conference on Learning Representations (ICLR)*.

[Micikevicius et al., 2018] Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., and Wu, H. (2018). Mixed precision training.

[Nichol and Dhariwal, 2021] Nichol, A. and Dhariwal, P. (2021). Improved denoising diffusion probabilistic models.

[Nichol et al., 2021] Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. (2021). Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv:2112.10741*.

[Perez et al., 2017] Perez, E., Strub, F., de Vries, H., Dumoulin, V., and Courville, A. (2017). Film: Visual reasoning with a general conditioning layer.

[Radford et al., 2021] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

[Raffel et al., 2020] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P. J., et al. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

[Ramesh et al., 2022] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022). Hierarchical text-conditional image generation with CLIP latents. *arXiv:2204.06125*.

[Ramesh et al., 2021] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021). Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR.

[Rampas et al., 2023] Rampas, D., Pernias, P., and Aubreville, M. (2023). A novel sampling scheme for text- and image-conditional image synthesis in quantized latent spaces. *arXiv:2211.07292*.

[Reed et al., 2016] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. (2016). Generative adversarial text to image synthesis. In *International conference on machine learning*, pages 1060–1069. PMLR.

[Rombach et al., 2022] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695.

[Rumelhart et al., 1985] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

[Saharia et al., 2022] Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., et al. (2022). Photorealistic text-to-image diffusion models with deep language understanding. *arXiv:2205.11487*.

[Schuhmann et al., 2022] Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. (2022). Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv:2210.08402*.

[Sohl-Dickstein et al., 2015] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR.

[Song et al., 2023] Song, Y., Dhariwal, P., Chen, M., and Sutskever, I. (2023). Consistency models.

[Tan and Le, 2020] Tan, M. and Le, Q. V. (2020). Efficientnet: Rethinking model scaling for convolutional neural networks.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

[Xue et al., 2022] Xue, L., Barua, A., Constant, N., Al-Rfou, R., Narang, S., Kale, M., Roberts, A., and Raffel, C. (2022). Byt5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

[Yu et al., 2022] Yu, J., Xu, Y., Koh, J. Y., Luong, T., Baid, G., Wang, Z., Vasudevan, V., Ku, A., Yang, Y., Ayan, B. K., Hutchinson, B., Han, W., Parekh, Z., Li, X., Zhang, H., Baldridge, J., and Wu, Y. (2022). Scaling autoregressive models for content-rich text-to-image generation. *arXiv:2206.10789*.

[Zhang et al., 2017] Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., and Metaxas, D. N. (2017). Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915.

# A Appendix


(a) An armchair in the shape of an avocado


(b) A Pikachu shaped hat


(c) An illustration of a baby daikon radish in a tutu walking a dog


(d) A hedgehog using a calculator


(e) A volcano erupting next to the golden gate bridge


(f) Painting of an alien by Claude Monet


(g) View of Saturn from space


(h) Illustration of an astronaut in a space suit playing guitar

Figure A.1: Non-cherry-picked samples from Würstchen and Stable Diffusion (random seed: 42).


(a) The Eiffel tower made of French fries


(b) Underwater cathedral


(c) A portrait of a nightmare creature


(d) A dog is holding a gun


(e) A rat holding a red lightsaber in a white background


(f) A pink elephant on a beach


(g) A small blue book sitting on a large red book.


(h) A red cube on top of a blue cube


(i) A man with an apple instead of a head

Figure A.2: Non-cherry-picked samples from Würstchen and Stable Diffusion (random seed: 42).


(a) A sign that says Hello World.


(b) A storefront with Text to Image written on it.


(c) Lego Arnold Schwarzenegger.


(d) A photo of a confused grizzly bear in calculus class.


(e) A zebra to the right of a fire hydrant.


(f) A train on top of a surfboard.


(g) A cross-section view of a brain.


(h) Five dogs on the street.


(i) Rainbow coloured penguin.

Figure A.3: Non-cherry-picked samples from Würstchen and Stable Diffusion (random seed: 42).


(a) A shark in the desert.


(b) An elephant under the sea.


(c) A red colored car.


(d) A laptop on top of a teddy bear.


(e) An oil painting portrait of the regal Burger King posing with a Whopper.


(f) Photo of a mega Lego space station inside a kid's bedroom.


(g) A magnifying glass over a page of a 1950s batman comic.


(h) Darth Vader playing with raccoon in Mars during sunset.


(i) Watercolor painting of a field of sunflowers

Figure A.4: Non-cherry-picked samples from Würstchen and Stable Diffusion (random seed: 42).


(a) Pencil sketch of a woman's face.


(b) An old woman buying vegetables in the market. Ukiyo-e


(c) The international space station in the style of Pablo Picasso


(d) Drawing of man fighting a robot. Anime style.


(e) A still of SpongeBob in the movie Toy Story.


(f) Abraham Lincoln in the GoldenEye 007 video game for Nintendo64 released in 1997.


(g) Barack Obama as the final boss of the 1993 video game Doom


(h) An 8-bit pixel art fan fiction version of the video game Halo for XBox


(i) A still of a group of wild animals including giraffes and lions in the movie The Matrix from 1999

Figure A.5: Non-cherry-picked samples from Würstchen and Stable Diffusion (random seed: 42).