SSDD: Single-Step Diffusion Decoder for Efficient Image Tokenization

Théophane Vallaeys[1,2] ✉, Jakob Verbeek[1], and Matthieu Cord[2,3]

[1] Meta Fundamental AI Research (FAIR), France
[2] Sorbonne University, France
[3] Valeo.ai, France
https://github.com/facebookresearch/SSDD

Abstract. Tokenizers are a key component of state-of-the-art generative image models, extracting the most important features from the signal while reducing data dimension and redundancy. Most current image tokenizers are based on KL-regularized variational autoencoders (KL-VAE), trained with reconstruction, perceptual and adversarial losses. Diffusion decoders have been proposed as a more principled alternative to model the distribution over images conditioned on the latent. However, matching the performance of KL-VAE still required adversarial losses, as well as a higher decoding time due to iterative sampling. To address these limitations, we introduce a new pixel diffusion decoder architecture for improved scaling and training stability, benefiting from transformer components and GAN-free training. We use distillation to replicate the performance of the diffusion decoder in an efficient single-step decoder. This makes SSDD the first GAN-free single-step diffusion decoder to reach state-of-the-art continuous tokenization, with higher reconstruction quality and faster sampling than KL-VAE. In particular, SSDD improves reconstruction FID from 0.87 to 0.46 with 1.4× higher throughput and preserves generation quality of DiTs with 3.8× faster sampling. As such, SSDD can be used as a drop-in replacement for KL-VAE, and for building higher-quality and faster generative models.

Keywords: Image tokenization · Diffusion models · Image reconstruction · Generative models

1 Introduction

Current state-of-the-art image generation models, whether they are based on diffusion [63], flow matching [21] or autoregressive modeling [22, 79], rely on tokenizers as a key component to reduce the high-dimensional visual signal to compact latent representations. This allows for efficient training and inference of these models in the latent space, before decoding the generated latent representations back to the original signal representation in pixel-space. Beyond efficiency, tokenization has also been shown to improve generative performance [22, 63].

✉ Corresponding author: webalorn@meta.com

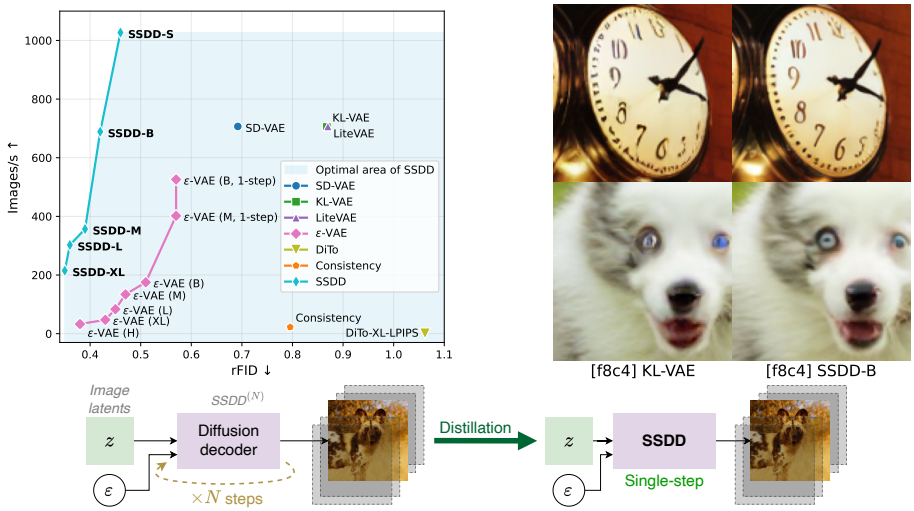

Fig. 1: Left: Decoding speed-quality Pareto-front for different state-of-the-art f8c4 feedforward and diffusion decoders. **Right:** Zoomed-in patches from reconstructions obained with KL-VAE and our SSDD tokenizer with similar throughput. **Bottom:** High-level overview of our SSDD decoder. First we train a multi-step decoder, generating diverse high-quality samples, and then distill it into a single-step decoder. Both multi-step and distilled single-step decoders can generate diverse decodings of a given latent z by using the input noise ε to sample from the distribution over images given the latent.

The quality and compression ratio of the tokenizer directly affect the generative quality of latent models. Image tokenization models either produce discrete tokens —directly compatible with, e.g., discrete text models such as LLMs—, or continuous tokens —which can be modeled with methods such as diffusion and flow matching. In this work we focus on *continuous* tokenization, as used in most state-of-the-art models for images and videos.

For generative modeling, tokenizers are usually trained as autoencoders. They use an encoder $E : x \mapsto z$ to compress an image $x \in \mathbb{R}^{3\times H\times W}$ into a latent representation $z \in \mathbb{R}^{c\times \frac{H}{f}\times \frac{W}{f}}$, where f is the spatial downsampling factor and c is the channel dimension of the latent. The decoder $D : z \mapsto \hat{x}$ recovers an estimate $\hat{x}$ of the original signal x. As the encoding process is lossy when using typical configurations with $3f^2 > c$, the decoding process can be seen as a generative task conditioned on the latent representation $z = E(x)$. Reconstruction quality can be quantified in different ways:

1. **Distortion:** expressed as $\mathbb{E}_x \Delta(x, \hat{x})$, where $\hat{x} = D(E(x))$, and Δ is a measure of the distortion *between the original and reconstructed images*. Distortion can be quantified by low-level similarity metrics such as MSE, MAE, PSNR and SSIM, or by high-level metrics based on features found in trained deep neural networks, such as LPIPS [93] and DreamSim [23].

2. **Distribution shift:** expressed as $d(P_x, P_{\hat{x}})$, with d being a measure of the divergence *between the original and reconstructed image distributions*. Metrics commonly used to evaluate distribution shift of generative models include Fréchet Inception Distance (FID) [27], as well as Density and Coverage [51].

As shown by Blau and Michaeli [8], there exists a fundamental trade-off between these two different notions of reconstruction quality (referred to as *Distortion* and *Perception* in [8]) when the signal is compressed sufficiently. While Distortion can be minimized by a deterministic decoder, Distribution shift is optimized using a generative decoder. See Appendix A for more detail and an illustrative example.

Common tokenizers such as the KL-VAE [63] are trained using an L1 reconstruction loss, LPIPS [93], and a GAN discriminator [25], with an added KL-regularization on the latent space. L1 and LPIPS losses directly optimize for distortion, while the GAN loss aims to decrease distribution shift. However, the diversity of these deterministic decoders, which affects the distribution shift, is limited by the expressivity of the latent z. In particular, for a given image x and its encoding $z = E(x)$ there is only a single decoding possible. Additionally, stable scaling of GAN losses is non-trivial [9]. Diffusion auto-encoders [17,94] have recently been proposed as an alternative to directly model the conditional data distribution and optimizing for distributional shift. Diffusion decoders naturally model the distribution of missing signal from the latents [7], by being able to sample multiple decodings for a given latent, allowing higher compression ratios and focusing the task of latent modeling on higher-level features. While these models have shown improved reconstruction and generation results at a given model size, they still suffer from (i) slow sampling, due to the iterative denoising process and (ii) the reliance on a GAN loss to achieve state-of-the-art results.

In this paper we introduce SSDD, a new diffusion autoencoder that overcomes the limitations of previous diffusion decoders. In particular, we propose an efficient and scalable alternative to the simple convolutional U-Net architectures for diffusion decoders, based on the U-ViT [32,33] and achieve superior results at all tested model scales. We experimentally analyze the key design choices to train our model, which is the first GAN-free single-step diffusion decoder to reach state-of-the-art continuous tokenization, improving the scaling and stability of the training without any diverging runs across model scales. To avoid multiple denoising steps while decoding, we propose a single-step distillation method for SSDD with minimal quality impact. See Figure 1 for an illustration of the SSDD decoder training and sample results.

Taken together, our contributions make SSDD both the highest quality and fastest diffusion decoder, surpassing KL-VAE and previous diffusion auto-encoders on quality and reconstruction speed at the same downsampling factor. In the f8c4 setting (8× downsampling, 4 latent channels) our smallest SSDD-S improves reconstruction FID over KL-VAE from 0.87 to 0.46 with 1.4× higher decoding throughput. At matched throughput, SSDD-B reaches 0.42 rFID. At matched parameter count, SSDD-M reaches 0.39 rFID. As our architecture differs substantially from KL-VAE, decoding time is the most meaningful efficiency metric, and SSDD improves on all fronts. By leveraging an SSDD auto-encoder

with higher compression rate (f16c4), we increase the DiT sampling speed of latents by 3.8× while preserving sample quality when decoded using our model.

2 Related work

Image tokenizers. Tokenization is the first stage in state-of-the-art generative image models. Tokenizers for diffusion models compress images into continuous low-dimension representations [57, 63], using a convolutional autoencoder [28] trained with L1, KL, LPIPS [93] and GAN [25] losses, which we refer to as KL-VAE. Discrete autoencoders are used as tokenizers for discrete autoregressive modeling [11, 22, 54, 62, 76, 87, 88], replacing the KL regularization with quantization. Advances have been made on multiple aspects of those models, including efficiency [65], semantic alignment [84], architecture and training [13, 82], spatial downsampling [15], quantization methods [35, 39, 50, 89, 96], and 1D-tokenization [14, 91]. However, most tokenizers rely on deterministic decoders optimized for image-to-image distortion, but may be suboptimal in terms of distribution shift [8].

Diffusion decoders. Diffusion decoders [34, 44, 56, 61, 71], usually based on U-Net models [64], are trained using a diffusion loss. They have been applied in a compression setting [7, 10, 31, 83] in place or in addition to deterministic decoders to improve quality by modeling the distribution of realistic reconstructions. Some variants also use diffusion decoders as an additional stage within a KL-VAE [4, 58], further compressing the encoding space. Recent work has shown that these models are scalable and easy to train when only using the diffusion loss [17], and, at the cost of adding additional LPIPS and GAN losses, can be competitive to similarly-sized VAEs [6, 68, 94]. Diffusion decoders still suffer from slow multi-step sampling and the reliance on hard-to-scale adversarial training [9]. Our work aims to alleviate these limitations.

Diffusion and flow image modeling. Diffusion and flow matching models estimate the reverse process of a forward process that interpolates between data and pure Gaussian noise [29, 74, 75]. They were first shown by [20] to outperform previous methods such as GANs [25] and VAEs [37]. Diffusion models, often operating in a compressed latent space for efficiency [63], underlie current state-of-the-art models for text-to-image [5, 53, 59, 63], image-to-image [66], and video generation [1, 24, 26, 60]. Ongoing research efforts aim to improve various aspects of these models, including training, sampling speed and model architecture [16, 36, 38, 52, 57]. In particular, flow matching [2, 43, 45] reformulates the diffusion as a velocity field estimation between noise and data, yielding improvements and faster sampling in generative image models [21]. Pixel-space diffusion has seen progress with new training architectural improvements [18, 32, 33]. As an alternative to diffusion, autoregressive models have seen fast development both for discrete and continuous modeling [11, 22, 41, 54, 78, 79, 90]. Recent approaches have also been mixing both methods to improve generation [12, 40, 77, 97]. Our work leverages continuous flow matching for generative modeling in pixel space.

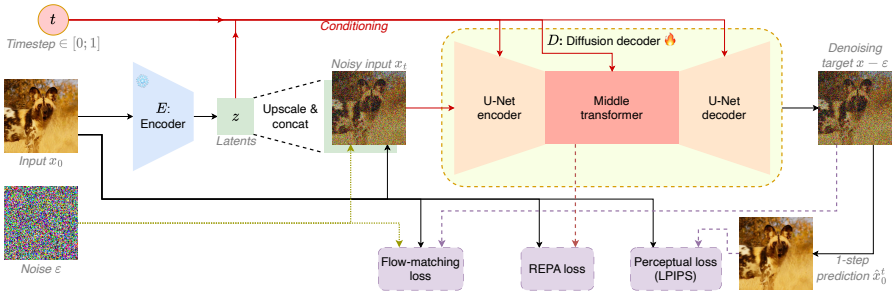

Fig. 2: Training of SSDD tokenizer. Input image x_0 is mapped to latents z by the (possibly frozen) encoder E. Noise $\epsilon \sim \mathcal{N}(0, 1)$ is sampled and added to x_0 to form the noisy input x_t. The decoder D learns to denoise x_t conditioned on denoising time t (via AdaNorm) and latent z (by using it as input to the decoder and AdaNorm layers). The model is trained with flow-matching generative loss, REPA features alignment loss, and LPIPS perceptual loss.

Few-step inference. The main drawback of diffusion models is the iterative inference process which requires multiple, often several tens or hundreds, forward passes through the model. To address this, few- or single-step sampling methods have been developed, through distillation or trajectory alignment. Consistency models [48, 72, 73] distill pre-trained diffusion model capabilities into new few-step tailored generator models. Other methods fine-tune the diffusion model using a teacher-student setup for fast sampling [46, 47, 49, 67, 85, 86, 95]. We leverage this approach and study its impact on decoding performance of our models.

3 Single-step diffusion decoder method

We introduce here the main components of our method SSDD combining several key innovations to make diffusion decoders both efficient and practical. First, we introduce a hybrid U-Net–transformer architecture that leverages convolutional inductive biases while scaling effectively with transformer blocks (Section 3.1). Second, we propose a GAN-free training scheme based on flow matching, perceptual alignment, and feature regularization, ensuring stable training without adversarial losses (Section 3.2). Third, we design a single-step distillation process that transfers the behavior of multi-step diffusion to a fast one-step decoder, achieving both quality and speed (Section 3.3). Finally, we show that SSDD can operate with shared encoders, enabling compatibility with existing VAEs and reducing training costs (Section 3.4). Together, these components make SSDD the first diffusion decoder optimized for single-step reconstruction, without compromising generative performance.

3.1 A scalable pixel-space diffusion decoder architecture

Existing pixel-space diffusion decoders use the convolutional U-Net architecture [94] that was found to be successful in early pixel-space diffusion models [20].

Transformer architectures, however, have proven to be more effective and scalable for latent modeling [57]. We aim to combine the strengths of convolutional U-Nets to model features in pixel-space with the generative capabilities of diffusion transformers. We base our architecture on the U-ViT [32] with several modifications. See Figure 2 for an overview.

U-Net with latent transformer. Based on the U-ViT architecture, we modify a U-Net [20, 64] using four levels of two convolutional ResNet blocks and three convolution downsampling / upsampling layers by removing all attention layers and replacing the middle attention block by a transformer operating on tokens representing 8×8 pixel patches. Each transformer block uses the GEGLU activation function [70], a $4\times$ multiplicative factor for the hidden MLP dimension, and multi-head attention [80].

Time and position embedding. We embed time using adaptive group-normalization (AdaGN) [20] as the second normalization layer in ResNet blocks. To enforce a local inductive bias, we learn a relative positional embedding table [69] for each self-attention layer, with visual tokens only attending to tokens up to a distance of 8 on the width and height axes. This results in a fixed 17×17 attention window and relative positional embedding table.

Conditioning. Following DiTo [17] and ε-VAE [94], we condition our model by upsampling the latent feature grid z from $(c, \frac{H}{f}, \frac{W}{f})$ to (c, H, W), and concatenating it with the noised image along the channel dimension in our model. As our model is relatively deep and narrow compared to prior work, we improve conditioning by adding a second mechanism using adaptive normalization. We replace the first GroupNorm of ResNet blocks with AdaGN, and the first LayerNorm of transformer blocks by AdaLN [57].

Scaling. To study a wide range of decoding capacities, we scale our diffusion decoders from 13.4M (SSDD-S) to 153.8M parameters (SSDD-XL) by increasing the number of channels and of transformer layers. For most of our experiments we consider SSDD-M (48M parameters), which has a similar size as the convolutional decoder of KL-VAE [63] (47.2M parameters). More details about model parametrization can be found in Appendix B.

3.2 GAN-free training

Flow matching. We train our model using the optimal transport flow-matching loss [43] with $\sigma_{min} = 0$. Specifically, we use $x_t = (1-t)x + t\varepsilon$, and our decoder $D(x_t|t, z)$ is trained to predict the velocity field $\nu = x - \varepsilon$ with the L2 loss. This defines our main training loss $\mathcal{L}_{\text{FM}} = \| x - \varepsilon - D(x_t|t, z) \|^2$. Following [21], we sample t during training using the logit-normal distribution [3] with location $m = 0$ and scale $s = 1$.

Perceptual and regularization losses. As noted by [17], perceptual loss is important to guide the generation toward perceptually correct areas. We therefore add the LPIPS [93] loss $\mathcal{L}_{\text{LPIPS}} = \text{LPIPS}(x, \hat{x}_0)$, with $\hat{x}_0 = x_t + tD(x_t|t, z)$ the

single-step prediction of x_0 from x_t and z. As the middle part of our model acts as an implicit latent-space diffusion transformer, we further stabilize and accelerate our training by adding the REPA loss [92], denoted as $\mathcal{L}_{\text{REPA}}$. We use DINOv2-B [55] reference features for REPA, which are aligned with the tokens extracted from the 4th layer of our transformer, through a 2-layer MLP. The FM loss trains the decoder to predict the clean image trajectory, while LPIPS aligns perceptual features and REPA stabilizes transformer representations. We combine them in our final loss as $\mathcal{L} = \mathcal{L}_{\text{FM}} + \lambda_{\text{LPIPS}}\mathcal{L}_{\text{LPIPS}} + \lambda_{\text{REPA}}\mathcal{L}_{\text{REPA}}$, with $\lambda_{\text{LPIPS}} = 0.5$ and $\lambda_{\text{REPA}} = 0.25$.

Multi-scale adaptation. While autoencoders have been observed to generalize to higher resolutions than the one used for training [65], best performance is often reached when training or fine-tuning at specific higher target resolutions. To alleviate the need of training from scratch at multiple resolutions and reduce the required training compute, we use a two-stage training method, based on the procedure from [65]. During the first stage, we train a model on 128×128 crops. We additionally reduce the divergence between the features distribution of the training and fine-tuning images by adding random resizing to the data augmentation, from which we extract a 128×128 crop. During the second and third stages, we fine-tune and distill the first-stage weights at the target resolution (either 128×128 or 256×256). Given the increased cost of training at higher resolutions, we use the second model (optimized for 256×256) to evaluate on larger images (e.g. 512×512 or 1024×1024), which we observe to generalize sufficiently well.

3.3 Single-step diffusion decoder method

A key limitation of diffusion decoders is their iterative nature, which limits throughput in downstream generative models. We propose to align the behavior of a multi-step diffusion decoder with the one of a single-step generator, without sacrificing quality.

Sampling behavior and non-straight velocity dynamics. We first analyze the effect of the number of sampling steps N. As previously observed by [94], reconstruction quality is non-monotonic: we find that one-step denoising yields the best low-level distortion metrics (e.g., PSNR), while perceptual metrics (such as LPIPS and rFID) improve up to an intermediate number of steps before degrading. See Appendix C for corresponding experimental results quantifying these effects.

This behavior is rooted in the interaction between the flow-matching objective and the LPIPS loss. Let x_0 be the original image, $\nu = x_0 - \varepsilon$ the true velocity, and $\hat{\nu} = D(x_t, z)$ the predicted velocity. We define the composite reconstruction loss as

$$\mathcal{L}_\lambda(\hat{\nu}|\nu) = \|\hat{\nu} - \nu\|^2 + \lambda L(\hat{x}_0, x_0),$$

where L is the LPIPS loss and $\hat{x}_0 = x_t + t\hat{\nu}$. The gradient of this loss with respect to $\hat{\nu}$ is:

$$\nabla_{\hat{\nu}}\mathcal{L}_\lambda = 2(\hat{\nu} - \nu) + \lambda t \nabla L \tag{1}$$

$$= 2(\hat{\nu} - (\nu - \frac{\lambda t}{2}\nabla L)) \tag{2}$$

$$= \nabla_{\hat{\nu}}\mathcal{L}_0(\hat{\nu}|\nu - \frac{\lambda t}{2}\nabla L). \tag{3}$$

This reveals that optimizing $\mathcal{L}_\lambda$ is equivalent to learning a velocity field $\nu - \frac{\lambda t}{2}\nabla L$, which is *no longer straight*: $(\nabla_{\hat{\nu}} L)L(\hat{x}_0, x_0)$ is not necessarily aligned with ν, and its impact increases with the noise level t. Consequently, LPIPS-regularized Flow-Matching decoders learn non-straight trajectories that shift during training. While this hinders denoising with many steps, it provides a "sweet spot" for high-quality, few-step reconstruction. We discuss this behavior further in Appendix C. As we target diversity and quality for generative applications over reconstruction fidelity, we start with large denoising steps before adding a few denoising steps to refine details.

We adopt the scheduler from [68], yielding the timesteps schedule $t_i = (\frac{N-i+1}{N})^\rho$ with $\rho = 2$ and $N = 8$ steps for all our SSDD models, to balance realism (*low distributional shift*) and fidelity (*low distortion*).

One-step sampling behavior alignment by distillation. To address the key limitations of diffusion decoders, *i.e.* their slow multi-step inference process, we align the behavior of a single-step generation with that of multi-step through a lightweight distillation strategy. Specifically, we use a frozen copy of our decoder D_{ref} as a teacher to produce multi-step reconstructions $\hat{x}_{\text{ref}}$, and fine-tune the student decoder D to reproduce this behavior in a single step. Unlike [47] that solely rely on an L2 regression loss, we preserve the full training objective during distillation, with flow-matching and LPIPS terms being computed against teacher outputs. Each iteration, we sample $z \sim E(x)$, $\varepsilon \sim \mathcal{N}(0; I)$, $\hat{x}_{\text{ref}} = \text{Sample}_{\text{steps}=N}(D_{\text{ref}}, \varepsilon, z)$.

We fine-tune D using $t=1$, the same z, ε values and the altered loss $\mathcal{L}^{\text{distill}}$, with $\mathcal{L}^{\text{distill}}_{\text{FM}} = \| \hat{x}_{\text{ref}} - \varepsilon - D(x_t|t, z) \|^2$ and $\mathcal{L}^{\text{distill}}_{\text{LPIPS}} = \text{LPIPS}(x, \hat{x}_0)$. This allows the distilled decoder to mimic the perceptual and generative properties of the teacher, while operating in a single step and thus achieving much higher efficiency. To our knowledge, this is the first work showing that the behavior of multi-step diffusion decoders can be aligned to a single-step generator without a strong quality drop, making SSDD directly usable in generative pipelines.

3.4 Shared encoders

As our focus is on the decoder design, we use the standard KL-regularized convolutional design from [63] for the encoder. We refer to encoders by their patch size f controlling the spatial downsampling, and output channel dimension c. Unless specified otherwise, we use the f8c4 encoder in our experiments — compressing

RGB image patches of size 8×8 into 4 channels, achieving a compression factor of $8 \times 8 \times 3/4 = 48$.

Shared encoder. Previous works exploring diffusion autoencoders train a specific encoder with each decoder. While in principle this allows to reach optimal results, it creates a distinct latent space associated with each encoder. In particular, this requires training a separate generative diffusion model associated with each model. Despite the encoder performance being dependent on the image resolution, we find that we can train a near-optimal multi-resolution encoder with a simple procedure. We train an encoder together with a SSDD-M decoder, benefiting from multi-scale data augmentation, learning to encode features at different resolutions. We then freeze this encoder and use it to train decoders with a different number of parameters. Unless specified otherwise, our decoders all rely on a single shared encoder for each combination of downsampling factor (f) and channel count (c).

Pretrained encoders. SSDD can also be directly trained to decode images from any existing encoder mapping an input image to a grid-shaped latent representation. in additional experiments in Appendix D we explore this using various pretrained encoders, including KL-VAE [63], DiTo [17], and DC-AE [15].

4 Experimental validation

4.1 Experimental setup

We train and evaluate SSDD models of various sizes from SSDD-S to SSDD-H on ImageNet-1k [19], and consider generalization of results to COCO [42]. For details on model parametrization and training, see Appendix B. We use SSDD$^{(N)}$ to refer to models using N sampling steps, and plain SSDD for single-step distilled models.

Evaluation. We evaluate the quality of our models on the 50k images of the ImageNet evaluation set. For reconstruction, we assess the *distribution shift* using the reconstruction Fréchet Inception Distance (rFID) [27]. We additionally evaluate the following distortion metrics: PSNR and SSIM [81] for low-level similarity, and LPIPS [93] and DreamSim [23] as high-level perceptual metrics. To evaluate the impact of our model on generation quality, we train Diffusion Transformer models [57] on ImageNet-1k at 256×256 resolution using the original model configuration. Following [57], we train DiT-XL/2 models for 400k steps and evaluate the generation FID (gFID) without classifier-free guidance (CFG) [30] or, if specified, with a CFG of 1.375. We evaluate the throughput on a single H200 GPU, using the torch-compiled model for 1,000 iterations and a batch size of 128.

Baselines. We evaluate our model against the standard KL-VAE from [63] trained on ImageNet, and the commonly-used stronger performing ft-EMA VAE [57], which we refer to as SD-VAE. We also include LiteVAE [65] and VA-VAE [84], and compare against existing diffusion decoders DiTo [17] and ε-VAE [94]. We evaluate the publicly-available weights of SD-VAE (f8c4) and the KL-VAE (f8c4, f16c16, f32c64), and reproduce DiTo training from their

Table 1: Comparison with state-of-the-art models on ImageNet 256×256. Using a downsampling factor f of 8 or 16, and $c=4$ or $c=16$ latent channels. For each metric, we highlight the **first** and second best results. *#D: decoder parameter count. N: number of decoding steps. †: numbers reported from original papers. ‡: SD-VAE is the ft-EMA fine-tuned variant of KL-VAE on a larger dataset [57].*

	Method	#D	N	Ims./sec.	rFID↓	LPIPS↓	DreamSim↓	PSNR↑	SSIM↑
f8c4	SD-VAE‡	47.2M	1	707	0.69	0.061	0.042	25.52	0.78
	KL-VAE [63]	47.2M	1	705	0.87	0.065	0.046	24.11	0.75
	LiteVAE [63]†	53.3M	1	707	0.87	-	-	26.02	0.74
	ε-VAE (B, 1-step) [94]†	20.6M	1	526	0.57	-	-	-	-
	ε-VAE (M) [94]†	49.3M	3	134	0.47	-	-	27.65	0.84
	ε-VAE (H) [94]†	355.6M	3	33	0.38	-	-	**29.49**	**0.85**
	DiTo-XL-LPIPS [17]	592.2M	50	1	1.06	0.077	0.055	23.53	0.72
	Consistency decoder [6]†	625.1M	2	23	0.80	0.065	0.044	23.71	0.73
	SSDD-S	13.4M	1	**1027**	0.46	0.060	0.039	24.08	0.74
	SSDD-B	20.2M	1	689	0.42	0.058	0.036	24.18	0.75
	SSDD-M	48.0M	1	357	0.39	0.055	0.034	24.38	0.75
	SSDD-L	85.2M	1	302	0.36	0.053	0.032	24.49	0.76
	SSDD-XL	153.8M	1	215	**0.35**	**0.052**	**0.031**	24.60	0.76
f16c16	KL-VAE [63]	36.4M	1	**1196**	0.82	0.066	0.048	23.88	0.74
	VA-VAE [84]†	39.5M	1	1178	0.55	0.132	-	**26.10**	0.72
	FlowMo (continuous) [68]†	-	25	-	0.65	0.054	-	26.61	0.791
	SSDD-S	13.4M	1	985	0.45	0.061	0.038	23.75	0.73
	SSDD-B	20.2M	1	686	0.41	0.059	0.035	23.86	0.73
	SSDD-M	48.0M	1	368	0.40	0.056	0.033	24.08	0.74
	SSDD-L	85.2M	1	315	**0.34**	0.053	**0.030**	24.20	0.74
	SSDD-XL	153.8M	1	213	0.36	**0.052**	**0.030**	24.31	**0.75**
f16c4	KL-VAE [63]	36.4M	1	**1176**	2.93	-	-	20.57	0.66
	ε-VAE (M) [94]†	49.3M	3	134	1.91	-	-	21.27	0.69
	ε-VAE (H) [94]†	355.6M	3	33	1.35	-	-	**22.60**	**0.71**
	SSDD-S	13.4M	1	951	2.29	0.186	0.121	15.59	0.45
	SSDD-B	20.2M	1	663	1.77	0.177	0.111	16.06	0.46
	SSDD-M	48.0M	1	360	1.23	0.161	0.096	17.13	0.49
	SSDD-L	85.2M	1	305	0.96	0.155	0.089	17.26	0.50
	SSDD-XL	153.8M	1	216	**0.89**	**0.148**	**0.083**	17.72	0.51

codebase. For other baselines, we use available published results by lack of public code or weights.

4.2 Image reconstruction and generation

Fast image reconstruction. We compare reconstruction quality of our SSDD models against other baselines in Table 1 using three encoder configurations. All methods use the same encoder architecture, and we use single-step distilled models for SSDD. Our method outperforms all existing in terms of rFID, LPIPS and DreamSim, across all encoder configurations, using smaller models and a single sampling step. We also benefit from a significant throughput increase compared to multi-step models, while maintaining high reconstruction quality. In particular, for the **f8c4** encoder configuration, our SSDD-S model **outperforms all single-step decoding methods both in speed and perceptual quality**, including single-

step sampling from ε-VAE. Scaling the number of parameters to 345.9M, our SSDD-H obtains similar or better rFID compared to ε-VAE-H, while benefiting from a 3× increase in decoding speed. In Figure 1 we compare deterministic and generative reconstruction methods in a speed-rFID plot, showing that SSDD is Pareto optimal across the board.

Switching to **f16c16** and **f16c4** encoders, the standard KL-VAE decoders are deeper and narrower, reducing their parameter count with an associated drop in quality. Since our models keep similar size and speed across the different encoder configurations, SSDD consistently outperforms existing baselines with a higher compression ratio (**f16c4**), while maintaining similarly high throughput. Using these deeper encoders, the gap of performance between deterministic and diffusion decoders widens. This reveals the impact of generative-focused reconstruction for heavily compressed representations. We also observe an increased gap in sample quality between smaller and larger models, associated with their respective modeling capacity. Larger models bring larger quality gains, as the conditional image distribution modeling task complexity increases. Small models (SSDD-S) lack those generation capabilities, but achieve fast, high-quality reconstruction at lower compression rates. SSDD-M strikes a balance by achieving good reconstruction quality at all scales with a parameter count similar to standard KL-VAE models.

While SSDD improves over prior state-of-the-art autoencoders in terms of rFID, LPIPS and DreamSim, it yields somewhat worse results in terms of low-level distortion metrics (SSIM, PSNR). We believe, however, that these metrics are not representative for the perceived reconstruction quality — as illustrated in the qualitative examples in Figure 1, Figure S6, Figure S7 and Figure S5 — and should be de-emphasized for evaluation of generation-oriented decoders.

Scaling across model and image size. We scale the input resolution from 128 to 512 and model size from **S** to **H** in Table 2, using the f8c4 encoder configuration. Our SSDD models use single-step distilled decoders fine-tuned at either 128×128 (evaluations at 128), or 256×256 (evaluations at 256 and 512). Overall, SSDD yields significantly better results than existing baselines, even with smaller models, in particular at 128 and 512 resolution.

Impact of spatial downsampling. We conduct evaluations of reconstruction quality on both SSDD and KL-VAE across encoders with different spatial downsampling rates, but using the same total latent space size. In particular, we consider three settings where $f^2/c = 16$. In Figure 3 we show reconstruction quality as a function of the downsampling factor. We also visualize reconstructions at the lowest and highest spatial downsampling factors in Figure 3, and for more examples in Figures S6 and S7. For SSDD we observe a modest decrease in quality when increasing f, whereas KL-VAE suffers from a severe degradation in performance for the $f = 32$ setting. This confirms that SSDD architecture functions well for varying levels of spatial downsampling with limited effect on resulting reconstructions.

Table 2: Scaling of models and image resolution on ImageNet. All models use an f8c4 encoder. SSDD instances use the same shared encoder and are distilled into single-step decoders. Decoder with similar parameters count are shown highlighted with the same color. Evaluations at 512×512 or higher resolution are conducted using the same model as for 256×256.

Method	#D	128×128 rFID↓	LPIPS↓	256×256 rFID↓	LPIPS↓	512×512 rFID↓	LPIPS↓	1024×1024 rFID↓	LPIPS↓
KL-VAE	47.2M	-	-	0.87	0.065	0.29	0.064	0.20	0.059
ε-VAE (B)	20.6M	1.94	-	0.52	-	0.61	-	-	-
ε-VAE (M)	49.3M	1.58	-	0.47	-	0.53	-	-	-
ε-VAE (L)	89.0M	1.47	-	0.45	-	0.41	-	-	-
ε-VAE (XL)	140.6M	1.34	-	0.43	-	0.39	-	-	-
ε-VAE (H)	355.6M	1	-	0.38	-	0.35	-	-	-
DiTo-XL + LPIPS	592.2M	-	-	1.06	0.077	0.28	0.080	0.15	0.086
Consistency decoder	625.1M	-	-	0.80	0.065	0.30	0.066	0.15	0.061
SSDD-S	13.4M	1.89	0.061	0.46	0.060	0.28	0.062	0.16	0.064
SSDD-B	20.2M	1.48	0.059	0.42	0.058	0.22	0.060	0.13	0.058
SSDD-M	48.0M	1.04	0.056	0.39	0.055	0.20	0.057	0.12	0.055
SSDD-L	85.2M	0.88	0.054	0.36	0.053	**0.19**	0.055	**0.11**	0.053
SSDD-XL	153.8M	**0.81**	**0.052**	**0.35**	**0.052**	0.20	**0.053**	**0.11**	**0.051**

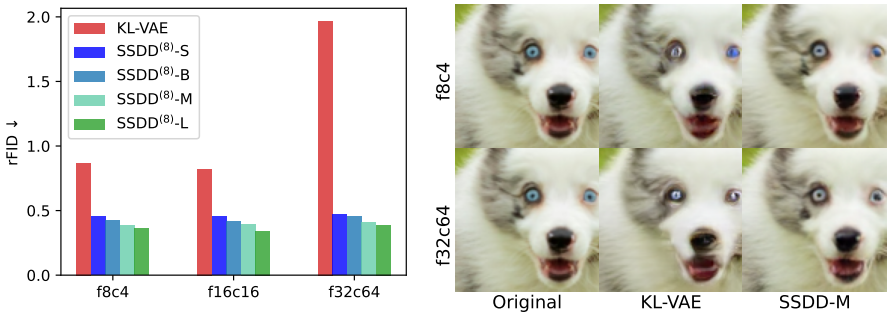

Fig. 3: Evolution of rFID and qualitative comparison when increasing spatial downsampling. Evaluated on ImageNet 256×256 with a constant compression ratio by adjusting c. Images patches are high-frequency features extracted from a larger 256×256 image.

Application on image generation. We conduct image generation experiments by training DiT-XL models [57] on ImageNet 256×256. In Table 3 we report results in terms of rFID and generation speed. We observe that the increased generative capabilities of the SSDD decoders results in higher sample quality across all setting compared to KL-VAE and DiTo, with and without CFG. While our smallest model SSDD-S outperforms all baselines, larger decoders progressively increase the image quality at all encoder configurations. Additionally, we show that SSDD can significantly increase the throughput for similar generation quality.

Table 3: Evaluation of autoencoders on image generation on ImageNet 256×256**.** Generation speed in images/second includes sampling the DiT-XL/2 model as well as pixel decoding.

	Autoencoder	No CFG gFID↓	No CFG Images/s	With CFG gFID↓	With CFG Images/s
f8c4	KL-VAE [63]	19.71	7.05	7.63	3.54
	DiTo-XL-LPIPS [17]	22.47	0.87	10.83	0.77
	SSDD-S	17.77	**7.07**	8.02	**3.55**
	SSDD-B	17.46	7.05	7.85	3.54
	SSDD-M	17.03	6.98	7.63	3.52
	SSDD-L	16.86	6.96	7.48	3.52
	SSDD-XL	**16.79**	6.89	**7.46**	3.50
f16c4	SSDD-S	20.28	**27.55**	9.83	**13.98**
	SSDD-B	18.89	27.20	8.98	13.89
	SSDD-M	17.24	26.29	7.99	13.65
	SSDD-L	16.27	25.95	7.34	13.55
	SSDD-XL	**16.08**	25.04	**7.12**	13.30
f16c16	KL-VAE [63]	31.96	**27.79**	26.42	**14.04**
	SSDD-S	25.84	27.57	20.51	13.98
	SSDD-B	25.59	27.24	20.28	13.90
	SSDD-M	25.20	26.34	19.91	13.66
	SSDD-L	25.03	26.02	19.74	13.57
	SSDD-XL	**25.01**	25.07	**19.68**	13.31

3.8×

Using a f16c4 encoder, the image is compressed into a 4× smaller embedding compared to the f8c4 setting. Using a larger decoder (SSDD-L) with DiT-XL/2, we can generate images 3.8× faster than with the same DiT based on KL-VAE with an f8c4 encoder, with no loss in quality both with and without CFG.

Ablation of our decoder design choices. In Table 4, we evaluate the impact of each of our design choices. We start from a DiTo-S model as a baseline: we reproduce the DiTo architecture from [17], and reduce the base channel number to 64, yielding a 48.3M parameters (similar to SSDD-M) decoder with an f8c4 encoder, which we refer to as DiTo-S. We train this model on 128×128 images, using the same optimizer and parameters as SSDD. We then progressively add components of our method. Each one improves the generative capabilities of the model (rFID), with some increasing the image-to-image distortion. In particular, our improved architecture and its regularization with the REPA loss bring the most improvements of 1.16 and 0.44 rFID respectively. Using a KL-encoder instead of the layer norm of [17] has a positive impact on performance, and allows us to use a standard encoder as commonly used to train generative models [21,57]. It focuses our work around the *decoder* component, without requiring specific properties from the encoder. Additionally, the use of a shared encoder (Section 3.4) and shared pre-training (stage 1 in Section 3.2), which improve training efficiency, do not hurt performance. Finally, only the distillation has a slight but negative impact on rFID, but brings the model from 8-steps to 1-step sampling. To demonstrate that we can achieve optimal results with a GAN-free method, we also evaluate the impact of adding a GAN loss. We follow the parametrization of

Table 4: Ablation of design choices starting from DiTo as baseline. We indicate the number N of sampling steps used for each model, directly affecting decoding speed.

Ablation	rFID↓		PSNR↑		DreamSim↓		N
DiTo-S-LPIPS *(#D=48.3M)*	3.17		23.10		0.107		24
+ SSDD-M decoder *(#D=48.0M)*	2.01	*-1.16*	23.38	*+0.28*	0.060	*-0.048*	24
+ REPA loss	1.58	*-0.44*	22.82	*-0.56*	0.104	*+0.045*	24
+ replace z-norm by KL	1.32	*-0.26*	23.02	*+0.20*	0.098	*-0.006*	24
+ logit-normal sampling	1.30	*-0.02*	23.13	*+0.11*	0.101	*+0.003*	16
+ t-spacing sampler	1.25	*-0.05*	23.43	*+0.30*	0.097	*-0.004*	8
+ EMA	1.17	*-0.08*	23.38	*-0.05*	0.097	*±0.000*	8
+ shared encoder (§)	1.07	*-0.10*	23.58	*+0.20*	0.090	*+0.030*	8
+ shared pre-training **(SSDD$^{(8)}$-M)**	**0.97**	*-0.10*	23.04	*-0.54*	0.087	*-0.003*	8
+ distillation **(SSDD-M)**	1.04	*+0.07*	23.28	*+0.24*	**0.084**	*-0.002*	1
(§) + GAN loss	*1.07*	*±0.00*	*23.58*	*±0.00*	*0.089*	*-0.001*	8

the *adversarial denoising trajectory matching* from [94], and use it in combination with the setup where we added the shared encoder, shown three lines above and indicated by §. We do not observe any significant impact on rFID and other metrics. A more detailed analysis of the impact of the GAN is included in Appendix E. We further isolate the contribution of each ablation component into essential, standard-recipe and pipeline-only groups in Appendix D, and report additional ablations on the GAN loss applied to the distilled single-step model and on L2-only distillation in Appendix D (Table S5). Appendix D additionally compares our two-stage pipeline to single-stage MeanFlow training.

Evaluation on COCO images. To assess the generalization of our model to other sets of natural images, we follow [94] by evaluating SSDD and comparable baselines on the 5k images from the COCO-2017 evaluation set [42]. Results are reported in Table 5. We observe that SSDD generalizes better than existing baselines. In particular, SSDD-B yields better reconstruction than ε-VAE (H), while it was only surpassed by SSDD-L and SSDD-XL on ImageNet in Table 2. This shows that, despite a relatively small size and high decoding speed, our models provide high reconstruction fidelity at various model scales and data.

5 Conclusion

We introduce SSDD, a diffusion autoencoder that leverages flow-matching, perceptual alignment and REPA regularization to reach state-of-the-art reconstruction and high generation quality without relying on adversarial training. Through a lightweight distillation strategy, we show that multi-step diffusion behavior can be compressed into a single-step decoder, yielding up to 10× faster decoding than ε-VAE (H) at similar rFID. On ImageNet, SSDD consistently outperforms KL-VAE and recent diffusion decoders across reconstruction (rFID, LPIPS, DreamSim) and downstream generation (gFID with DiT), with particularly strong gains under high-compression settings. These results establish

Table 5: Generalization of auto-encoders to new datasets. All models use an f8c4 encoder and are trained at 256×256 resolution on ImageNet, except for the consistency decoder. Evaluation conducted on the COCO-2017 5K validation set.

Method	#D	COCO 256×256 rFID↓	COCO 256×256 LPIPS↓	COCO 512×512 rFID↓	COCO 512×512 LPIPS↓
KL-VAE [63]	47.2M	4.65	0.063	2.68	0.064
ε-VAE (M) [94]	49.3M	3.98	-	-	-
ε-VAE (H) [94]	355.6M	3.65	-	-	-
DiTo-XL + LPIPS [17]	592.2M	5.95	0.076	2.93	0.079
Consistency decoder [6]	625.1M	4.53	0.063	2.65	0.065
SSDD-S	13.4M	3.77	0.059	2.51	0.062
SSDD-B	20.2M	3.62	0.057	2.34	0.059
SSDD-M	48.0M	3.41	0.054	2.24	0.056
SSDD-L	85.2M	<u>3.29</u>	<u>0.052</u>	<u>2.15</u>	<u>0.054</u>
SSDD-XL	153.8M	**3.25**	**0.050**	**2.13**	**0.053**

SSDD as the first GAN-free single-step diffusion decoder to reach state-of-the-art continuous tokenization, combining speed, stability, and generative fidelity, and providing a scalable foundation for future large-scale generative modeling.

Acknowledgement

This work has been supported by chair VISA DEEP (ANR-20-CHIA-0022) and Cluster PostGenAI@Paris (ANR-23-IACL-0007, FRANCE 2030).

References

1. Agarwal, N., Ali, A., Bala, M., Balaji, Y., Barker, E., Cai, T., Chattopadhyay, P., Chen, Y., Cui, Y., Ding, Y., Dworakowski, D., Fan, J., Fenzi, M., Ferroni, F., Fidler, S., Fox, D., Ge, S., Ge, Y., Gu, J., Gururani, S., He, E., Huang, J., Huffman, J., Jannaty, P., Jin, J., Kim, S.W., Klár, G., Lam, G., Lan, S., Leal-Taixe, L., Li, A., Li, Z., Lin, C.H., Lin, T.Y., Ling, H., Liu, M.Y., Liu, X., Luo, A., Ma, Q., Mao, H., Mo, K., Mousavian, A., Nah, S., Niverty, S., Page, D., Paschalidou, D., Patel, Z., Pavao, L., Ramezanali, M., Reda, F., Ren, X., Sabavat, V.R.N., Schmerling, E., Shi, S., Stefaniak, B., Tang, S., Tchapmi, L., Tredak, P., Tseng, W.C., Varghese, J., Wang, H., Wang, H., Wang, H., Wang, T.C., Wei, F., Wei, X., Wu, J.Z., Xu, J., Yang, W., Yen-Chen, L., Zeng, X., Zeng, Y., Zhang, J., Zhang, Q., Zhang, Y., Zhao, Q., Zolkowski, A.: Cosmos world foundation model platform for physical AI. arXiv preprint **2501.03575** (2025), https://arxiv.org/abs/2501.03575
2. Albergo, M.S., Vanden-Eijnden, E.: Building normalizing flows with stochastic interpolants. In: ICLR (2023), https://openreview.net/forum?id=li7qeBbCR1t
3. Atchison, J., Shen, S.M.: Logistic-normal distributions: Some properties and uses. Biometrika **67**(2), 261–272 (1980)
4. Bachmann, R., Allardice, J., Mizrahi, D., Fini, E., Kar, O.F., Amirloo, E., El-Nouby, A., Zamir, A., Dehghan, A.: FlexTok: Resampling images into 1d token sequences of flexible length. In: ICML (2025), https://openreview.net/forum?id=DgdOkUUBzf

5. Baldridge, J., Bauer, J., Bhutani, M., Brichtova, N., Bunner, A., Chan, K., Chen, Y., Dieleman, S., Du, Y., Eaton-Rosen, Z., Fei, H., de Freitas, N., Gao, Y., Gladchenko, E., Colmenarejo, S.G., Guo, M., Haig, A., Hawkins, W., Hu, H., Huang, H., Igwe, T.P., Kaplanis, C., Khodadadeh, S., Kim, Y., Konyushkova, K., Langner, K., Lau, E., Luo, S., Mokrá, S., Nandwani, H., Onoe, Y., van den Oord, A., Parekh, Z., Pont-Tuset, J., Qi, H., Qian, R., Ramachandran, D., Rane, P., Rashwan, A., Razavi, A., Riachi, R., Srinivasan, H., Srinivasan, S., Strudel, R., Uria, B., Wang, O., Wang, S., Waters, A., Wolff, C., Wright, A., Xiao, Z., Xiong, H., Xu, K., van Zee, M., Zhang, J., Zhang, K., Zhou, W., Zolna, K., Aboubakar, O., Akbulut, C., Akerlund, O., Albuquerque, I., Anderson, N., Andreetto, M., Aroyo, L., Bariach, B., Barker, D., Ben, S., Berman, D., Biles, C., Blok, I., Botadra, P., Brennan, J., Brown, K., Buckley, J., Bunel, R., Bursztein, E., Butterfield, C., Caine, B., Carpenter, V., Casagrande, N., Chang, M.W., Chang, S., Chaudhuri, S., Chen, T., Choi, J., Churbanau, D., Clement, N., Cohen, M., Cole, F., Dektiarev, M., Du, V., Dutta, P., Eccles, T., Elue, N., Feden, A., Fruchter, S., Garcia, F., Garg, R.: Imagen 3. arXiv preprint **2408.07009** (2024), https://doi.org/10.48550/arXiv.2408.07009
6. Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo, Y., Manassra, W., Dhariwal, P., Chu, C., Jiao, Y.: Improving image generation with better captions. https://cdn.openai.com/papers/dall-e-3.pdf (2023)
7. Birodkar, V., Barcik, G., Lyon, J., Ioffe, S., Minnen, D., Dillon, J.V.: Sample what you can't compress. arXiv preprint **2409.02529** (2024), https://arxiv.org/abs/2409.02529
8. Blau, Y., Michaeli, T.: The perception-distortion tradeoff. In: CVPR (2018)
9. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: ICLR (2019), https://openreview.net/forum?id=B1xsqj09Fm
10. Careil, M., Muckley, M.J., Verbeek, J., Lathuilière, S.: Towards image compression with perfect realism at ultra-low bitrates. In: ICLR (2024), https://arxiv.org/abs/2310.10325
11. Chang, H., Zhang, H., Jiang, L., Liu, C., Freeman, W.T.: MaskGIT: Masked generative image transformer. In: CVPR (2022)
12. Chen, B., Martí Monsó, D., Du, Y., Simchowitz, M., Tedrake, R., Sitzmann, V.: Diffusion forcing: Next-token prediction meets full-sequence diffusion. In: NeurIPS (2024), https://proceedings.neurips.cc/paper_files/paper/2024/file/2aee1c4159e48407d68fe16ae8e6e49e-Paper-Conference.pdf
13. Chen, H., Han, Y., Chen, F., Li, X., Wang, Y., Wang, J., Wang, Z., Liu, Z., Zou, D., Raj, B.: Masked autoencoders are effective tokenizers for diffusion models. In: ICML (2025), https://openreview.net/forum?id=dzwUOiBlQW
14. Chen, H., Wang, Z., Li, X., Sun, X., Chen, F., Liu, J., Wang, J., Raj, B., Liu, Z., Barsoum, E.: SoftVQ-VAE: Efficient 1-dimensional continuous tokenizer. In: CVPR (2025)
15. Chen, J., Cai, H., Chen, J., Xie, E., Yang, S., Tang, H., Li, M., Han, S.: Deep compression autoencoder for efficient high-resolution diffusion models. In: ICLR (2025), https://openreview.net/forum?id=wH8XXUOUZU
16. Chen, T.: On the importance of noise scheduling for diffusion models. arXiv preprint **2301.10972** (2023), https://arxiv.org/abs/2301.10972
17. Chen, Y., Girdhar, R., Wang, X., Rambhatla, S.S., Misra, I.: Diffusion autoencoders are scalable image tokenizers. arXiv preprint **2501.18593** (2025), https://arxiv.org/abs/2501.18593

18. Crowson, K., Baumann, S.A., Birch, A., Abraham, T.M., Kaplan, D.Z., Shippole, E.: Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers. In: ICML (2024), https://proceedings.mlr.press/v235/crowson24a.html
19. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: CVPR (2009)
20. Dhariwal, P., Nichol, A.: Diffusion models beat GANs on image synthesis. In: NeurIPS (2021), https://proceedings.neurips.cc/paper_files/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf
21. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., Podell, D., Dockhorn, T., English, Z., Rombach, R.: Scaling rectified flow transformers for high-resolution image synthesis. In: ICML (2024), https://proceedings.mlr.press/v235/esser24a.html
22. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: CVPR (2021), https://doi.ieeecomputersociety.org/10.1109/CVPR46437.2021.01268
23. Fu, S., Tamir, N., Sundaram, S., Chai, L., Zhang, R., Dekel, T., Isola, P.: DreamSim: Learning new dimensions of human visual similarity using synthetic data. In: NeurIPS (2023), https://arxiv.org/abs/2306.09344
24. Girdhar, R., Singh, M., Brown, A., Duval, Q., Azadi, S., Rambhatla, S.S., Shah, A., Yin, X., Parikh, D., Misra, I.: Factorizing text-to-video generation by explicit image conditioning. In: ECCV (2024), https://doi.org/10.48550/arXiv.2311.10709
25. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS (2014)
26. Gupta, A., Yu, L., Sohn, K., Gu, X., Hahn, M., Li, F.F., Essa, I., Jiang, L., Lezama, J.: Photorealistic video generation with diffusion models. In: ECCV (2024)
27. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In: NeurIPS (2017), https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf
28. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006), https://www.science.org/doi/abs/10.1126/science.1127647
29. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS (2020), https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf
30. Ho, J., Salimans, T.: Classifier-free diffusion guidance. In: NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications (2021), https://openreview.net/forum?id=qw8AKxfYbI
31. Hoogeboom, E., Agustsson, E., Mentzer, F., Versari, L., Toderici, G., Theis, L.: High-fidelity image compression with score-based generative models. arXiv preprint **2305.18231** (2023), https://arxiv.org/abs/2305.18231
32. Hoogeboom, E., Heek, J., Salimans, T.: Simple diffusion: End-to-end diffusion for high resolution images. In: ICML (2023), https://proceedings.mlr.press/v202/hoogeboom23a.html
33. Hoogeboom, E., Mensink, T., Heek, J., Lamerigts, K., Gao, R., Salimans, T.: Simpler diffusion (SiD2): 1.5 FID on ImageNet512 with pixel-space diffusion. In: CVPR (2025), https://arxiv.org/abs/2410.19324
34. Hudson, D.A., Zoran, D., Malinowski, M., Lampinen, A.K., Jaegle, A., McClelland, J.L., Matthey, L., Hill, F., Lerchner, A.: SODA: Bottleneck diffusion models for representation learning. In: CVPR (2024)

35. Huh, M., Cheung, B., Agrawal, P., Isola, P.: Straightening out the straight-through estimator: Overcoming optimization challenges in vector quantized networks. In: ICML (2023), https://proceedings.mlr.press/v202/huh23a.html
36. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. In: NeurIPS (2022), https://proceedings.neurips.cc/paper_files/paper/2022/file/a98846e9d9cc01cfb87eb694d946ce6b-Paper-Conference.pdf
37. Kingma, D., Welling, M.: Auto-encoding variational Bayes. In: ICLR (2014)
38. Kingma, D., Gao, R.: Understanding diffusion objectives as the ELBO with simple data augmentation. In: NeurIPS (2023), https://proceedings.neurips.cc/paper_files/paper/2023/file/ce79fbf9baef726645bc2337abb0ade2-Paper-Conference.pdf
39. Lee, D., Kim, C., Kim, S., Cho, M., Han, W.S.: Autoregressive image generation using residual quantization. In: CVPR (2022)
40. Li, T., Tian, Y., Li, H., Deng, M., He, K.: Autoregressive image generation without vector quantization. In: NeurIPS (2024), https://proceedings.neurips.cc/paper_files/paper/2024/file/66e226469f20625aaebddbe47f0ca997-Paper-Conference.pdf
41. Liang, W., Yu, L., Luo, L., Iyer, S., Dong, N., Zhou, C., Ghosh, G., Lewis, M., Yih, W., Zettlemoyer, L., Lin, X.V.: Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. In: ICLR 2025 Workshop on World Models: Understanding, Modelling and Scaling (2025), https://openreview.net/forum?id=vksqRL2Yyz
42. Lin, T., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.: Microsoft COCO: common objects in context. In: ECCV (2014), https://arxiv.org/abs/1405.0312
43. Lipman, Y., Chen, R.T.Q., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. In: ICLR (2023), https://openreview.net/forum?id=PqvMRDCJT9t
44. Liu, D., Peng, Y., Tang, H., Chen, Y., Han, C., Ge, Z., Jiang, D., Liao, M.: DGAE: Diffusion-guided autoencoder for efficient latent representation learning. arXiv preprint **2506.09644** (2025), https://arxiv.org/abs/2506.09644
45. Liu, X., Gong, C., qiang liu: Flow straight and fast: Learning to generate and transfer data with rectified flow. In: ICLR (2023), https://openreview.net/forum?id=XVjTT1nw5z
46. Liu, X., Zhang, X., Ma, J., Peng, J., qiang liu: InstaFlow: One step is enough for high-quality diffusion-based text-to-image generation. In: ICLR (2024), https://openreview.net/forum?id=1k4yZbbDqX
47. Luhman, E., Luhman, T.: Knowledge distillation in iterative generative models for improved sampling speed. arXiv preprint **2101.02388** (2021), https://arxiv.org/abs/2101.02388
48. Luo, S., Tan, Y., Huang, L., Li, J., Zhao, H.: Latent consistency models: Synthesizing high-resolution images with few-step inference. In: ICLR (2023), https://arxiv.org/abs/2310.04378
49. Meng, C., Rombach, R., Gao, R., Kingma, D., Ermon, S., Ho, J., Salimans, T.: On distillation of guided diffusion models. In: CVPR (2023)
50. Mentzer, F., Minnen, D., Agustsson, E., Tschannen, M.: Finite scalar quantization: VQ-VAE made simple. In: ICLR (2024), https://openreview.net/forum?id=8ishA3LxN8

51. Naeem, M.F., Oh, S.J., Uh, Y., Choi, Y., Yoo, J.: Reliable fidelity and diversity metrics for generative models. In: ICML (2020), https://proceedings.mlr.press/v119/naeem20a.html
52. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: ICML (2021), https://proceedings.mlr.press/v139/nichol21a.html
53. Nichol, A.Q., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., Mcgrew, B., Sutskever, I., Chen, M.: GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In: ICML (2022), https://proceedings.mlr.press/v162/nichol22a.html
54. van den Oord, A., Vinyals, O., Kavukcuoglu, K.: Neural discrete representation learning. In: NeurIPS (2017), https://proceedings.neurips.cc/paper_files/paper/2017/file/7a98af17e63a0ac09ce2e96d03992fbc-Paper.pdf
55. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.Y., Li, S.W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: DINOv2: Learning robust visual features without supervision. Transactions on Machine Learning Research (2024), https://openreview.net/forum?id=a68SUt6zFt
56. Pandey, K., Mukherjee, A., Rai, P., Kumar, A.: DiffuseVAE: Efficient, controllable and high-fidelity generation from low-dimensional latents. Transactions on Machine Learning Research (2022), https://openreview.net/forum?id=ygoNPRiLxw
57. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: ICCV (2023)
58. Pernias, P., Rampas, D., Richter, M.L., Pal, C.J., Aubreville, M.: Würstchen: An efficient architecture for large-scale text-to-image diffusion models. In: ICLR (2024), https://openreview.net/forum?id=gU58d5QeGv
59. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: SDXL: Improving latent diffusion models for high-resolution image synthesis. In: ICLR (2024), https://openreview.net/forum?id=di52zR8xgf
60. Polyak, A., Zohar, A., Brown, A., Tjandra, A., Sinha, A., Lee, A., Vyas, A., Shi, B., Ma, C.Y., Chuang, C.Y., Yan, D., Choudhary, D., Wang, D., Sethi, G., Pang, G., Ma, H., Misra, I., Hou, J., Wang, J., Jagadeesh, K., Li, K., Zhang, L., Singh, M., Williamson, M., Le, M., Yu, M., Singh, M.K., Zhang, P., Vajda, P., Duval, Q., Girdhar, R., Sumbaly, R., Rambhatla, S.S., Tsai, S., Azadi, S., Datta, S., Chen, S., Bell, S., Ramaswamy, S., Sheynin, S., Bhattacharya, S., Motwani, S., Xu, T., Li, T., Hou, T., Hsu, W.N., Yin, X., Dai, X., Taigman, Y., Luo, Y., Liu, Y.C., Wu, Y.C., Zhao, Y., Kirstain, Y., He, Z., He, Z., Pumarola, A., Thabet, A., Sanakoyeu, A., Mallya, A., Guo, B., Araya, B., Kerr, B., Wood, C., Liu, C., Peng, C., Vengertsev, D., Schonfeld, E., Blanchard, E., Juefei-Xu, F., Nord, F., Liang, J., Hoffman, J., Kohler, J., Fire, K., Sivakumar, K., Chen, L., Yu, L., Gao, L., Georgopoulos, M., Moritz, R., Sampson, S.K., Li, S., Parmeggiani, S., Fine, S., Fowler, T., Petrovic, V., Du, Y.: Movie Gen: A cast of media foundation models. arXiv preprint **2410.13720** (2024)
61. Preechakul, K., Chatthee, N., Wizadwongsa, S., Suwajanakorn, S.: Diffusion autoencoders: Toward a meaningful and decodable representation. In: CVPR (2022)
62. Razavi, A., van den Oord, A., Vinyals, O.: Generating diverse high-fidelity images with VQ-VAE-2. In: NeurIPS (2019), https://proceedings.neurips.cc/paper_files/paper/2019/file/5f8e2fa1718d1bbcadf1cd9c7a54fb8c-Paper.pdf
63. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022)

64. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention – MICCAI (2015)
65. Sadat, S., Buhmann, J., Bradley, D., Hilliges, O., Weber, R.M.: LiteVAE: Lightweight and efficient variational autoencoders for latent diffusion models. In: NeurIPS (2024), https://proceedings.neurips.cc/paper_files/paper/2024/file/0735ab358bf9bf0f30af6c871b666ec8-Paper-Conference.pdf
66. Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., Norouzi, M.: Palette: Image-to-image diffusion models. In: SIGGRAPH (2022), https://doi.org/10.1145/3528233.3530757
67. Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. In: ICLR (2022), https://openreview.net/forum?id=TIdIXIpzhoI
68. Sargent, K., Hsu, K., Johnson, J., Fei-Fei, L., Wu, J.: Flow to the mode: Mode-seeking diffusion autoencoders for state-of-the-art image tokenization. In: ICCV (2025), https://arxiv.org/abs/2503.11056
69. Shaw, P., Uszkoreit, J., Vaswani, A.: Self-attention with relative position representations. In: NAACL (2018), https://arxiv.org/abs/1803.02155
70. Shazeer, N.: GLU variants improve transformer. arXiv preprint **2002.05202** (2020), https://arxiv.org/abs/2002.05202
71. Shi, J., Wu, C., Liang, J., Liu, X., Duan, N.: DiVAE: Photorealistic images synthesis with denoising diffusion decoder. arXiv preprint **2206.00386** (2022), https://arxiv.org/abs/2206.00386
72. Song, Y., Dhariwal, P.: Improved techniques for training consistency models. In: ICLR (2024), https://openreview.net/forum?id=WNzy9bRDvG
73. Song, Y., Dhariwal, P., Chen, M., Sutskever, I.: Consistency models. In: ICML (2023), https://proceedings.mlr.press/v202/song23a.html
74. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. In: NeurIPS (2019), https://proceedings.neurips.cc/paper_files/paper/2019/file/3001ef257407d5a371a96dcd947c7d93-Paper.pdf
75. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: ICLR (2021), https://openreview.net/forum?id=PxTIG12RRHS
76. Sun, P., Jiang, Y., Chen, S., Zhang, S., Peng, B., Luo, P., Yuan, Z.: Autoregressive model beats diffusion: Llama for scalable image generation. arXiv preprint **2406.06525** (2024), https://doi.org/10.48550/arXiv.2406.06525
77. Tang, H., Wu, Y., Yang, S., Xie, E., Chen, J., Chen, J., Zhang, Z., Cai, H., Lu, Y., Han, S.: HART: Efficient visual generation with hybrid autoregressive transformer. In: ICLR (2025), https://openreview.net/forum?id=q5s0v4xQe4
78. Team, C.: Chameleon: Mixed-modal early-fusion foundation models. arXiv preprint **2405.09818** (2025), https://arxiv.org/abs/2405.09818
79. Tian, K., Jiang, Y., Yuan, Z., Peng, B., Wang, L.: Visual autoregressive modeling: Scalable image generation via next-scale prediction. In: NeurIPS (2024), https://proceedings.neurips.cc/paper_files/paper/2024/file/9a24e284b187f662681440ba15c416fb-Paper-Conference.pdf
80. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017), https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
81. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. Transactions on Image Processing **13**(4), 600–612 (2004)

82. Yang, J., Li, T., Fan, L., Tian, Y., Wang, Y.: Latent denoising makes good visual tokenizers. arXiv preprint **2507.15856** (2025), https://arxiv.org/abs/2507.15856
83. Yang, R., Mandt, S.: Lossy image compression with conditional diffusion models. In: NeurIPS (2023), https://proceedings.neurips.cc/paper_files/paper/2023/file/ccf6d8b4a1fe9d9c8192f00c713872ea-Paper-Conference.pdf
84. Yao, J., Yang, B., Wang, X.: Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In: CVPR (2025)
85. Yin, T., Gharbi, M., Park, T., Zhang, R., Shechtman, E., Durand, F., Freeman, W.T.: Improved distribution matching distillation for fast image synthesis. In: NeurIPS (2024), https://proceedings.neurips.cc/paper_files/paper/2024/file/54dcf25318f9de5a7a01f0a4125c541e-Paper-Conference.pdf
86. Yin, T., Gharbi, M., Zhang, R., Shechtman, E., Durand, F., Freeman, W.T., Park, T.: One-step diffusion with distribution matching distillation. In: CVPR (2024)
87. Yu, J., Li, X., Koh, J.Y., Zhang, H., Pang, R., Qin, J., Ku, A., Xu, Y., Baldridge, J., Wu, Y.: Vector-quantized image modeling with improved VQGAN. In: ICLR (2022), https://openreview.net/forum?id=pfNyExj7z2
88. Yu, L., Cheng, Y., Sohn, K., Lezama, J., Zhang, H., Chang, H., Hauptmann, A.G., Yang, M.H., Hao, Y., Essa, I., Jiang, L.: MAGVIT: Masked generative video transformer. In: CVPR (2023)
89. Yu, L., Lezama, J., Gundavarapu, N.B., Versari, L., Sohn, K., Minnen, D., Cheng, Y., Gupta, A., Gu, X., Hauptmann, A.G., Gong, B., Yang, M.H., Essa, I., Ross, D.A., Jiang, L.: Language model beats diffusion - tokenizer is key to visual generation. In: ICLR (2024), https://openreview.net/forum?id=gzqrANCF4g
90. Yu, Q., He, J., Deng, X., Shen, X., Chen, L.C.: Randomized autoregressive visual generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 18431–18441 (October 2025)
91. Yu, Q., Weber, M., Deng, X., Shen, X., Cremers, D., Chen, L.C.: An image is worth 32 tokens for reconstruction and generation. In: NeurIPS (2024), https://proceedings.neurips.cc/paper_files/paper/2024/file/e91bf7dfba0477554994c6d64833e9d8-Paper-Conference.pdf
92. Yu, S., Kwak, S., Jang, H., Jeong, J., Huang, J., Shin, J., Xie, S.: Representation alignment for generation: Training diffusion transformers is easier than you think. In: ICLR (2025), https://openreview.net/forum?id=DJSZGGZYVi
93. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
94. Zhao, L., Woo, S., Wan, Z., Li, Y., Zhang, H., Gong, B., Adam, H., Jia, X., Liu, T.: Epsilon-VAE: Denoising as visual decoding. In: ICML (2025), https://openreview.net/forum?id=oK4ot2wAJU
95. Zhao, W., Bai, L., Rao, Y., Zhou, J., Lu, J.: UniPC: A unified predictor-corrector framework for fast sampling of diffusion models. In: NeurIPS (2023), https://proceedings.neurips.cc/paper_files/paper/2023/file/9c2aa1e456ea543997f6927295196381-Paper-Conference.pdf
96. Zhao, Y., Xiong, Y., Kraehenbuehl, P.: Image and video tokenization with binary spherical quantization. In: ICLR (2025), https://openreview.net/forum?id=yGnsH3gQ6U
97. Zhou, C., Yu, L., Babu, A., Tirumala, K., Yasunaga, M., Shamis, L., Kahn, J., Ma, X., Zettlemoyer, L., Levy, O.: Transfusion: Predict the next token and diffuse images with one multi-modal model. In: ICLR (2025), https://openreview.net/forum?id=SI2hI0frk6