

DA-VAE: Plug-in Latent Compression for Diffusion via Detail Alignment

Xin Cai¹³, Zhiyuan You¹, Zhoutong Zhang^{2†}, Tianfan Xue¹³⁴

¹Multimedia Laboratory, The Chinese University of Hong Kong

²Adobe NextCam ³Shanghai AI Laboratory ⁴CPII under InnoHK

caixin025@gmail.com, zhiyuanyou@foxmail.com, zhoutongz@adobe.com, tfxue@ie.cuhk.edu.hk



Figure 1. We propose **Detail-Aligned VAE (DA-VAE)**, a VAE model that increases the compression rate of a pretrained VAE, while requiring only light-weight finetuning of the original diffusion backbone while preserving image quality. Image results are from a finetuned SD3.5 Medium. DA-VAE accelerates the original SD3.5 Medium model by 6.04 times for 2048×2048 image generation.

Abstract

Reducing the token count is crucial for both efficient training and inference of latent diffusion models, especially at high resolution. A common approach is to build high-compression-rate image tokenizers that store more information by allocating more channels per token. However, when trained only with reconstruction objectives, high-

dimensional latent spaces often fail to maintain meaningful structure, which in turn complicates diffusion training. As a result, existing methods introduce additional training targets, such as semantic alignment or selective dropout, to enforce structure in the latent space, but these approaches typically require costly retraining of the diffusion model. Pre-trained diffusion models, however, already exhibit a structured, lower-dimensional latent space; thus, a simpler idea is to expand the latent dimensionality while preserving this

[†] Project lead.

structure. To this end, we propose **Detail-Aligned VAE (DA-VAE)**, a method that increases the compression ratio of a pretrained VAE while requiring only lightweight adaptation for the pretrained diffusion backbone. Specifically, DA-VAE imposes an explicit latent layout: the first C channels are taken directly from the pretrained VAE at a base resolution, and an additional D channels encode extra details that emerge at higher resolutions. We introduce a simple detail-alignment mechanism to encourage the expanded latent space to share the structural properties of the original space defined by the first C channels. Finally, we present a warm-start fine-tuning strategy that enables 1024×1024 image generation with Stable Diffusion 3.5 using only 32×32 tokens, $4\times$ fewer than the original model, within a compute budget of 5 H100-days. We further demonstrate that our method unlocks 2048×2048 image generation with SD3.5, achieving a $6\times$ speedup while preserving image quality. We further validate the effectiveness of our method and design choices quantitatively on ImageNet.

1. Introduction

Recent text-to-image Diffusion Transformers (DiTs) have achieved state-of-the-art image generation quality. Various works therefore aim to improve efficiency for those models from different perspectives, such as quantization [26], few-step distillation [49], and efficient attention mechanisms [43]. Orthogonal to these aspects, another direction to improve efficiency is through token count reduction. Since self-attention’s computational cost is quadratic over number of tokens, a $4\times$ reduction of tokens would result in $16\times$ reduction for its computational cost.

Existing high-compression-ratio tokenizers [8, 11] are often trained from scratch, aiming to squeeze more pixels into a token with more channels. Since these works introduce a new latent space, the downstream diffusion model needs to be trained from scratch as well, which requires a tremendous training cost and a large training set. Adding more to the problem, it is known that high-dimensional features prohibit effective diffusion model training, where one needs to introduce structures to the new latent space, either through semantic alignment or through training time dropout. Those challenges combined make this paradigm difficult to iterate—one needs to first train a tokenizer, balance between reconstruction and alignment/auxiliary tasks [46], then train a generative model from scratch to verify whether the newly introduced latent space is effective for generation.

We propose a different yet simple paradigm to increase the compression ratio of the VAE without complete retraining from scratch. That is, we start with a pretrained diffusion model, and aim to increase tokenizer efficiency by explicitly introducing a scale-space structure over the channel dimension of each token. Specifically, with a tokenizer that encodes an image at resolution $H \times W$ with T tokens,

we aim to increase the dimension of each token, such that those T tokens can represent an image at a higher resolution of $sH \times sW$. To achieve this without complete retraining, we keep the first C channels of each token the same as the latent of the image at base resolution. We then introduce D extra channels to each token, aiming to encode detail information only available at high resolution. With this design, we can then fine-tune the diffusion network trained on original $H \times W$ tokens, as our DA-VAE inherits those tokens.

Still, naively fine-tuning a diffusion denoising network on this new latent space may still fail, since the extra detail channels lack meaningful structure [8, 11], therefore hindering downstream diffusion training. To overcome this, we impose an explicit alignment constraint over the detail channels D , such that they should have similar structures to the pretrained latent channels C . This alignment proves to be crucial for downstream diffusion training.

Based on this observation, our pipeline is designed as follows. Since we re-use the original latent in the first C dimension, we use a warm-start strategy to further speed up fine-tuning. Specifically, we zero initialize the patch embedder for the extra D channels, while using the original patch embedder for the first C channels from the pretrained weights. We further introduce an optimization schedule that penalizes the extra D channels less during the early training steps. We show that, through ablation studies, this recipe yields better generation results given a fixed training budget. We further validate our method by fine-tuning SD3.5M [38] from 512×512 to 1024×1024 image generation while keeping the token number fixed. This results in an overall $\approx 4\times$ speedup compared to naive 1024×1024 image generation, where adaptation only takes 5 H100-days. We further demonstrate 2048×2048 image generation for SD3.5M with a $6\times$ speedup, while the original model cannot reliably generate coherent structures.

In summary, our contributions are:

- A method that improves tokenizer efficiency on the pretrained DiT without costly retraining.
- An explicitly structured latent that supports downstream fine-tuning through detail alignment.
- A fine-tuning recipe that efficiently adapts a pretrained DiT to the structured latent, enabling $2\times$ higher resolution under the same token budget.
- We validate our method’s effectiveness quantitatively on ImageNet and qualitatively by fine-tuning SD3.5, where the adaptation for SD3.5M only takes 5 H100-days.

2. Related Work

Diffusion model acceleration. Diffusion models [21] achieve strong image quality [13, 34, 53] but are computationally expensive due to many function evaluations over large latent grids. Existing acceleration mainly follows three directions. First, improved ODE/SDE solvers and

consistency-style objectives reduce the number of sampling steps [27, 54, 55, 58, 60, 62]. Second, one- or few-step generators distill the full trajectory into a small number of evaluations [15, 16, 29, 35, 47, 48]. Third, per-step efficiency is improved via pruning and token/feature selection, quantization, and optimized attention or execution (e.g., compiler stacks, parallelization, caching) [1, 14, 28, 37, 39, 40, 43, 44, 57]. However, these methods preserve the tokenization scheme and thus still scale quadratically with the number of latent tokens. We instead target *token efficiency*: we design a structured latent representation that enables DiTs to operate with substantially fewer tokens while keeping the backbone and sampler largely unchanged, and our tokenizer can be combined with the above techniques for further speedup.

Image tokenizers for generation. Latent diffusion models replace pixel-space denoising with denoising in a lower-resolution latent space learned by a 2D continuous tokenizer, typically an $8\times$ VAE [33]. This design is widely adopted by subsequent systems, which mainly scale model size and data while keeping a similar tokenizer [2, 9, 10, 12, 13, 24, 30, 32, 63], so the number of latent tokens still grows quadratically with image resolution. To reduce this token burden, recent work proposes more aggressively compressed tokenizers: DC-AEs and follow-ups [8, 11] build deeper encoder-decoder hierarchies that operate at higher spatial downsampling factors, and several 1D tokenizers [5, 6, 22, 45, 50] further lower the number of tokens fed into the diffusion backbone. However, these token-reduction schemes typically require training a new generative model directly on the new latent space [43, 44].

A complementary direction focuses less on compression and more on learning *semantically aligned* latent spaces that improve the trade-off between reconstruction and generation [4, 36, 46, 52, 59]. Prior work shows that shaping the latent geometry to better match semantic structure can yield more robust generation. However, these methods mainly focus on improving the global semantic structure of the latent space, and pay less attention to preserving structured fine-grained details that are critical for high-resolution image synthesis.

Our method is orthogonal to both deep-compression and semantic-alignment tokenizers. We instead target a *token-efficient* latent space that remains compatible with an existing pretrained DiT: rather than discarding the original tokenizer, we introduce a structured base-detail composition and explicitly align the new latent representation to the original VAE space, allowing us to reduce tokens while keeping the DiT backbone and training objective largely unchanged.

Efficient autoencoder adaptation. A related line of work studies how to upgrade or replace the autoencoder while reusing as much of the generative backbone as possible. Previous work such as [9, 31] adopts a stronger tokenizer and adapts the DiT on top of it for high-resolution gen-

eration, but the retraining pipeline is still computationally expensive. Concurrent work DC-Gen [11, 18] adapts a pretrained DiT to a new, more compressed latent space. However, this target space differs from the original VAE latent space, and the mismatch between them is non-trivial to compensate. In contrast, our method keeps the original VAE latent space as a reference: we introduce a structured latent space and train a compressor that reduces the number of tokens in an aligned latent space, so that the pretrained DiT can be reused with minimal modification.

3. Method

State-of-the-art text-to-image diffusion models [24, 32, 33] typically compress latent spaces to reduce computational cost. To describe the compression ability of a tokenizer, three metrics are often used: feature down-sampling rate f , number of channels per token C , and the patch size p of the downstream diffusion model. For an image of size $H \times W$, the latent dimension of all tokens is $(H \times W \times C)/(f^2 \times p^2)$. To increase token efficiency, previous works [43] have shown that increasing the downsampling ratio f is both efficient and friendly to downstream generation model training than increasing the patch size p . Moreover, simply increasing the downsampling ratio f limits the reconstruction abilities of the tokenizer, and previous tokenizers [8, 11] therefore increase the channel number C to compensate for that.

However, increasing C naively brings many challenges to the downstream diffusion training. As discussed in [11, 46], training a diffusion model on wide-channel tokens is unstable, and semantic alignment or auxiliary tasks are often needed to increase convergence of diffusion. This process usually requires retraining both tokenizer and the diffusion model from scratch, which is prohibitively expensive, introducing significant training cost and data collection cost.

We therefore propose DA-VAE together with a fine-tuning recipe that can both reduce the token size (larger downsampling ratio f) and generate friendly tokens for diffusion training. Core to our method is an explicitly structured latent space with an alignment strategy, and a warm-start diffusion fine-tuning recipe that adapts to the structured latent within a reasonable compute budget.

3.1. Structured Latent and Alignment

We describe the designs and training of DA-VAE, which increases a pretrained VAE’s spatial compression rate f and number of channels C with a structured latent space.

Structured latent space. We start with a pretrained VAE encoder E that encodes an image \mathbf{I} of resolution $H \times W$ into latent space z , whose dimensions are $C \times \frac{H}{f} \times \frac{W}{f}$. To improve its efficiency, we encode a higher resolution image \mathbf{I}_{hr} of size $sH \times sW$ into latent z_{hr} of size $C' \times \frac{H}{f} \times \frac{W}{f}$, where $C' > C$. In our experiment, we set $s = 2$.

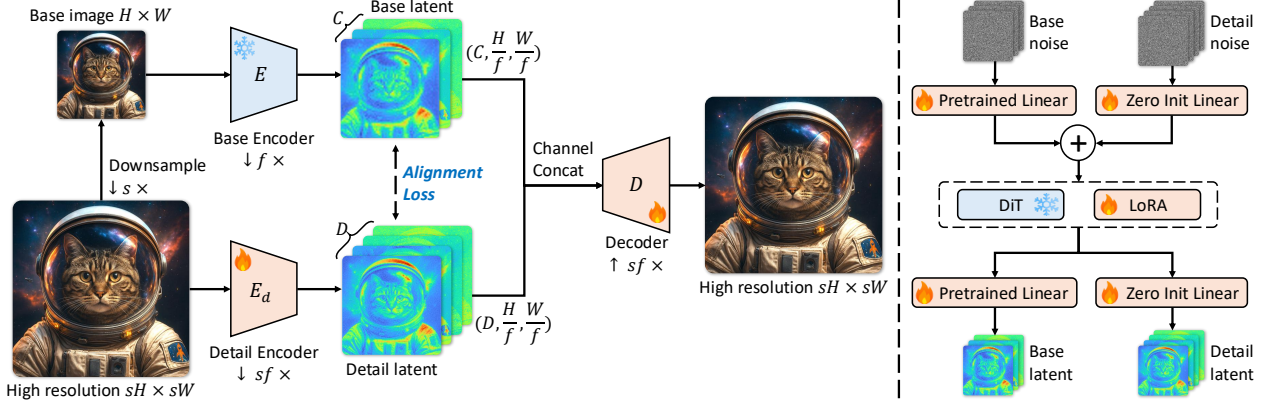


Figure 2. **Overview of our method.** Left: **Illustration of our Detail-Aligned VAE (DA-VAE)**, which encodes a high-resolution image using the same number of visual tokens as the base image. Right: **Zero initialization** of the linear layer for detail latent. At the beginning of training, the model keep pretrained diffusion model capability of generating images at the base resolution.

Our structured latent is designed such that $C' = C + D$, where the first C channels are exactly the latent of I . We augment the latent with an additional D -channel detail branch, encoded from a separate encoder E_{hr} using I_{hr} .

Specifically, our structured latent z_{hr} is composed of two parts concatenated over the channel dimension:

$$z_{hr} = [z, z_d] \in \mathbb{R}^{(C+D) \times \frac{H}{f} \times \frac{W}{f}}, \quad (1)$$

where

$$\begin{aligned} z &= E(I) \in \mathbb{R}^{C \times \frac{H}{f} \times \frac{W}{f}}, \\ z_d &= E_d(I_{hr}) \in \mathbb{R}^{D \times \frac{H}{f} \times \frac{W}{f}}. \end{aligned} \quad (2)$$

To decode the structured latent into image, we use a single decoder D such that $D(z_{hr})$ reconstructs I_{hr} . Throughout our experiments, we keep E fixed to its pretrained weights, and only optimize parameters of E_d and D . Fig. 2 shows this design.

Latent alignment. Naively adding more channels to the existing latent makes diffusion training difficult. When the VAE encoder is trained only with a reconstruction loss, the extra detail channels z_d tend to absorb noisy residuals rather than forming a meaningful semantic structure. As shown in the right column of Fig. 3, features derived from z_d are poorly organized and weakly correlated with either the original latent z or the underlying labels, which makes them hard for the downstream diffusion model to exploit.

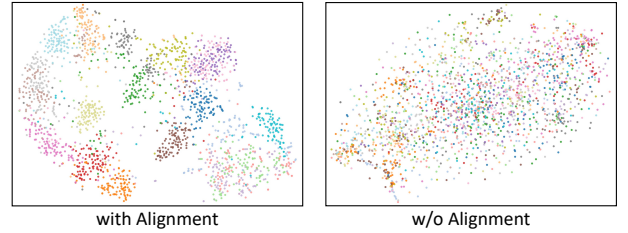
To regularize these channels, inspired by recent semantic-alignment works, we introduce a latent alignment loss that encourages z_d to be consistent with the pretrained latent z . Specifically, we minimize

$$\mathcal{L}_{\text{align}} = \|\text{Proj}(z_d) - z\|^2, \quad (3)$$

where $\text{Proj}(\cdot) : \mathbb{R}^{D \times H \times W} \rightarrow \mathbb{R}^{C \times H \times W}$ is a parameter-free channel-wise grouped reduction, defined as:

$$\text{Proj}(z_r)[i, h, w] = \frac{1}{r} \sum_{j=1}^r z_d[ir + j, h, w], \quad (4)$$

(a) t-SNE of Detail Latent on ImageNet Based on VA-VAE



(b) First PCA component of Detail Latent Based on SD3-VAE

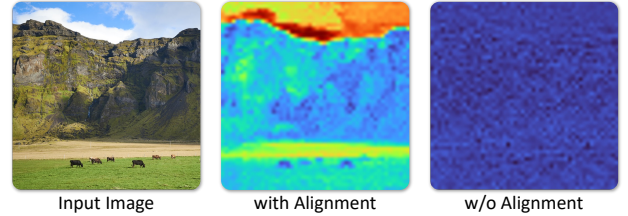


Figure 3. **Effect of the proposed latent alignment loss** on the learned detail feature z_d . In each pair, the right column shows training with only reconstruction loss (no alignment), and the left column shows training with our alignment loss. (a) DA-VAE on VA-VAE: the detail features (points colored by class) become more class-separable and well clustered under alignment, suggesting that z_d inherits the semantic structure of the original latent. (b) DA-VAE on SD3-VAE: alignment encourages the detail branch to capture fine-grained textures while preserving the global image layout, instead of collapsing into noisy residuals.

where $r = D/C$, and i, h, w are indices over channel, height and width. Note that for special cases where $r = 1$, this reduces to a simple L_2 norm between z_d and z .

With the alignment loss, the additional latent z_d learns a structure that closely mirrors z rather than drifting to arbitrary residuals, as illustrated in the left column of Fig. 3. This makes the enriched latent representation much more amenable to downstream diffusion training.

Objective for VAE. In addition to alignment, we adopt the standard reconstruction losses for VAE training, *i.e.*, perceptual loss, L_1 loss, adversarial loss and KL regulariza-

tion:

$$\mathcal{L}_{\text{rec}} = \lambda_L \text{LPIPS} + \lambda_1 L_1 + \lambda_{\text{adv}} \mathcal{L}_{\text{adv}} + \lambda_{KL} L_{KL}. \quad (5)$$

The full training loss is therefore given by:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda_{\text{align}} \mathcal{L}_{\text{align}}. \quad (6)$$

We empirically show that alignment introduces slight degradation in terms of reconstruction, yet greatly boosts generation, as shown by our experiments in Tab. 2.

3.2. Warm Start for Diffusion Fine-tuning

To adapt the pretrained DiT to the new latent space, we introduce a zero-init strategy and a gradual loss scheduling such that fine-tuning the diffusion model can be warm-started effectively from pretrained weights.

A zero-init strategy. As shown in right part of Fig. 2, the Diffusion Transformer uses a patch embedder P to map image latents into the DiT high-dimensional space, i.e.,

$$P(\cdot) : \mathbb{R}^{C \times \frac{H}{F} \times \frac{W}{F}} \rightarrow \mathbb{R}^{L \times \frac{H}{F_P} \times \frac{W}{F_P}}. \quad (7)$$

At the end of the network, an output layer O maps the DiT features back to the image latent space,

$$O(\cdot) : \mathbb{R}^{L \times \frac{H}{F_P} \times \frac{W}{F_P}} \rightarrow \mathbb{R}^{C \times \frac{H}{F} \times \frac{W}{F}}. \quad (8)$$

To accommodate our new latent space with more channels, we introduce an additional patch embedder P' and output layer O' :

$$\begin{aligned} P'(\cdot) : \mathbb{R}^{D \times \frac{H}{F} \times \frac{W}{F}} &\rightarrow \mathbb{R}^{L \times \frac{H}{F_P} \times \frac{W}{F_P}}, \\ O'(\cdot) : \mathbb{R}^{L \times \frac{H}{F_P} \times \frac{W}{F_P}} &\rightarrow \mathbb{R}^{D \times \frac{H}{F} \times \frac{W}{F}}. \end{aligned} \quad (9)$$

Under this design, the input to the DiT space is

$$P(z) + P'(z_{\text{hr}}), \quad (10)$$

and the DiT output L is decoded to the latent spaces via

$$\hat{u} = O(L), \quad \hat{u}_{\text{hr}} = O'(L), \quad (11)$$

where \hat{u} and \hat{u}_{hr} are the predictions for the original and high-resolution latents, respectively.

To preserve the pretrained behavior, we keep the original latent z path intact and zero-initialize P' and O' so that their outputs are zero at initialization. In this way, the overall model is exactly equivalent to the pretrained DiT at the beginning of fine-tuning, so the learned priors are fully preserved and training starts from a valid diffusion model. As illustrated in Fig. 4, this zero initialization leads to a much more stable optimization and significantly faster convergence compared to standard random initialization.

Gradual loss scheduling. Besides the zero-init strategy, we introduce a loss scheduling that allows the diffusion



Figure 4. **Comparison of zero initialization and random initialization.** Benefiting from our zero initialization, the model starts from a well-behaved point and converges faster during training.

fine-tuning process to gradually adapt to the extra channels. Specifically, we introduce a cosine-annealed weighting strategy applied to the diffusion training loss. Let $\hat{\mathbf{u}}_{hr} = [\hat{u}, \hat{u}_d]$ denote the DiT’s prediction on the structured latent $[z_b \parallel z_r]$, and $\mathbf{u}_{hr} = [\mathbf{u}, \mathbf{u}_d]$ are the target of v parameterization. We introduce a loss scheduling weight as:

$$w(n) = \begin{cases} \frac{1 - \cos(\pi n / N_{\text{warm}})}{2}, & n < N_{\text{warm}} \\ 1, & n \geq N_{\text{warm}} \end{cases} \quad (12)$$

where N_{warm} is a hyper-parameter for the number of warm-up steps, and n is the current training step. We apply this weight only to \hat{u}_d , that is:

$$\mathcal{L}_{\text{DiT}}(n) = \frac{1}{|B| + w(n)|R|} (\|\hat{\mathbf{u}} - \mathbf{u}\|_2^2 + w(n)\|\hat{\mathbf{u}}_d - \mathbf{u}_d\|_2^2) \quad (13)$$

At early iterations ($w(n) \approx 0$), gradients are dominated by the base latent channels, ensuring stable alignment with the pretrained backbone. As training proceeds, the diffusion model is gradually forced to model the extra detail channels z_d . This scheduling effectively regularizes the pretrained model to gradually adapt to the new latents.

End-to-end fine-tuning. With the above recipes, we fine-tune all blocks of a pretrained DiT, together with both patch embedders P , P' and output layers O and O' . For large model SD3.5 [13, 38], we use LoRA on all attention and FFN layers, but still optimize all parameters of P , P' , O and O' .

4. Experiments

We evaluate our method on both ImageNet and general text-to-image generation. On ImageNet, we show both qualitative and quantitative results by fine-tuning a base model to generate 512×512 images. To show the importance of individual components of our method, we perform ablation studies with quantitative results. For general text-to-image experiments, we fine-tune over SD3.5 Medium with LoRA and report qualitative and quantitative results, starting from a base resolution of 512×512 to generate 1024×1024 images. We also show qualitative results by fine-tuning from 1024×1024 to generate 2048×2048 images.

| Method | Reconstruction | | Token-nums (# of latent tokens) | Training Epoches | FID-50k ↓ | | Inception Score ↑ |
|---|---------------------------|-------------|------------------------------------|------------------------|----------------------------|----------------------------|------------------------------|
| | AutoEncoders | rFID | | | w/o CFG | w/ CFG | |
| DiT-XL [†] [30] REPA [†] [51] | SD-VAE (f8c4p2) | 0.48 | 32 × 32 | 2400 200 | 12.04 - | 3.04 2.08 | 255.3 274.6 |
| DiT-XL [†] [30] DC-Gen-DiT-XL [†] [19] | DC-AE (f32c32p1) | 0.66 | 16 × 16 | 2400 80 | 9.56 8.21 | 2.84 2.22 | 117.5 122.5 |
| LightningDiT-XL* [46] LightningDiT-XL [46] | VA-VAE (f16c32p2) | 0.50 | 16 × 16 | 80 80 | 21.79 11.31 | 3.98 3.12 | 229.7 254.5 |
| Ours | DA-VAE (f32c128p1) | 0.47 | 16 × 16 | 25 80 | 6.04 4.84 | 2.07 1.68 | 277.6 314.3 |

Table 1. Comparison of our method with other methods on ImageNet 512 × 512 in training efficiency and performance. [†] denotes that we directly re-use the numbers from the corresponding publication.



Figure 5. Qualitative samples from our model trained at 512 × 512 resolution on ImageNet.

| Autoencoder | Reconstruction | | | | Generation FID-10k ↓ |
|---------------------------|----------------|--------------|-------------|-------------|-------------------------|
| | rFID ↓ | PSNR ↑ | LPIPS ↓ | SSIM ↑ | |
| SD-VAE (f8c4p4) | 0.48 | 29.22 | 0.13 | 0.79 | 58.17 |
| DC-AE (f32c32p1) | 0.66 | 27.78 | 0.16 | 0.74 | 35.97 |
| VA-VAE (f16c32p2) | 0.50 | 28.43 | 0.13 | 0.78 | 44.65 |
| DA-VAE (f32c128p1) | 0.47 | 28.53 | 0.12 | 0.78 | 31.51 |

Table 2. Performance comparison of different autoencoders. All generation models were trained from scratch.

ImageNet experiment details. We use the pretrained VA-VAE and LightningDiT-XL from [46] as our base model, which is trained to generate images at the resolution of 256 × 256. VA-VAE uses a spatial compression factor $f = 16$ and $C = 32$ latent channels. Our structural latent adds additional $D = 96$ channels to the latent space, resulting in a total of 128 channels for each token. LightningDiT-XL is a DiT-XL/1 [30] diffusion model trained on the latent space of VA-VAE for 256 × 256 image generation, with a patch size $p = 1$. DA-VAE is trained for 100k steps with a batch size of 1024 and the DiT backbone is fully fine-tuned for 25 epochs with a batch size of 640 on 8 H100 GPUs, following our proposed recipe. We set $N_{warm} = 10k$ steps for our loss scheduling strategy as described in Sec. 3.2. Other

details can be found in the supplementary material.

Text-to-image generation details. For text-to-image generation, we conduct quantitative and qualitative experiments based on Stable Diffusion 3.5 Medium (SD3.5M) [13], fine-tuning it to generate at a resolution of 1024 × 1024 with a base resolution of 512 × 512. The SD3.5M uses a VAE with $f = 8$ compression ratio and $C = 16$ channels in the latent space. Our structural latent adds an additional $D = 16$ channels to the latent space, resulting in a total of 32 channels per token. SD3.5 uses a MMDiT-X diffusion backbone with 2.5B parameters and patch size of $p = 2$. We train DA-VAE for 10k steps with a batch size of 32 on SAM dataset [23] and fine-tune the SD3.5M backbone for 20k steps with a batch size of 128 on a synthetic dataset generated from the base model using the prompts from DiffusionDB [42]. During fine-tuning, we use our proposed recipe and set $N_{warm} = 5k$ steps. Due to compute constraints, we apply LoRA with a rank of 256 for all blocks in the DiT backbone, except for the patch embedder and the output layer. For quantitative evaluation, we use the MJHQ-30K [25] dataset and report CLIP-Score [61] and GenEval [17] results. See supplementary material for other details.



Figure 6. Comparison between our method (1024×1024) and Stable Diffusion 3.5 (1024×1024 by 512×512 upsample).



Figure 7. Comparison between our method and Stable Diffusion 3.5 on a 2048×2048 resolution. Zoom in for a better view.

4.1. ImageNet results

Autoencoder evaluation. We first evaluate the reconstruction and generation performance of DA-VAE compared to the base VA-VAE and other baselines. For reconstruction, we report rFID [20], PSNR, LPIPS [56] and SSIM [41] between the original images and the reconstructed images on the ImageNet validation set. For generation, we train a DiT-XL for 20k steps **from scratch** with same token budget on the latent space of each VAE and report the FID-10k on ImageNet 512×512 . For fair comparisons, we do not apply any classifier-free guidance during sampling. We summarize the results in Tab. 2. Our DA-VAE achieves a better trade-off between reconstruction and generation compared to other autoencoder baselines, demonstrating the effective-

ness of our design. Note that our structured latent itself allows faster diffusion training compared to baselines under the same token count, despite being high dimensional.

Diffusion fine-tuning evaluation. We evaluate the effectiveness of our structured latent and adaptation strategy by comparing class-to-image generation results of our method against state-of-the-art methods at 512×512 resolution.

Since we start with VA-VAE with LightningDiT-XL of patch size $p = 1$, we compare with fine-tuning LightningDiT-XL with $p = 2$, using the adaptation strategy proposed by DC-Gen [19]. As a performance reference, we also train LightningDiT-XL with $p = 2$ from scratch, annotated by an asterisk (*) in Tab. 1. We also compare with other autoencoders for 512×512 generation that uses

| Method | AutoEncoders | Token-nums (# of latent tokens) | Params (B) | Throughput (img/s) | FID ↓ | CLIP Score ↑ | GenEval ↑ |
|--------------------------------|--------------------------|------------------------------------|---------------|-----------------------|--------------|-----------------|-------------|
| PixArt-Σ | — | 64 × 64 | 0.6 | 0.40 | 6.15 | 28.26 | 0.54 |
| Hunyuan-DiT | — | 64 × 64 | 1.5 | 0.05 | 6.54 | 28.19 | 0.63 |
| SANA-1.5 | DC-AE(f32c32p1) | 32 × 32 | 4.8 | 0.26 | 5.99 | 29.23 | 0.80 |
| FLUX-dev | FLUX-VAE(f8c16p2) | 64 × 64 | 12 | 0.04 | 10.15 | 27.47 | 0.67 |
| SD3-medium | SD3-VAE (f8c16p2) | 64 × 64 | 2.0 | 0.36 | 11.92 | 27.83 | 0.62 |
| SD3.5-medium | SD3-VAE (f8c16p2) | 64 × 64 | 2.5 | 0.25 | 10.31 | 29.74 | 0.63 |
| SD3.5-medium [†] | SD3-VAE (f8c16p2) | 32 × 32 | 2.5 | 1.03 | 12.04 | 30.17 | 0.63 |
| Ours (SD3.5-M + DA-VAE) | DA-VAE (f16c32p2) | 32 × 32 | 2.5 | 1.03 | 10.91 | 31.91 | 0.64 |

Table 3. **Comparison of our method with SOTA approaches in efficiency and performance.** The FID and CLIP Score are reported on MJHQ-30K (1024 × 1024). The Throughput is tested on one A100 GPU with BF16 Precision with batch=10.

| Alignment | Reconstruction | | | | Generation |
|-------------------------------|----------------|--------|---------|--------|------------|
| loss weight λ_{align} | rFID ↓ | PSNR ↑ | LPIPS ↓ | SSIM ↑ | FID-10k ↓ |
| 0.0 | 0.59 | 29.23 | 0.11 | 0.80 | 16.37 |
| 0.1 | 0.55 | 28.70 | 0.12 | 0.79 | 9.58 |
| 0.5 | 0.47 | 28.53 | 0.12 | 0.78 | 9.27 |
| 1.0 | 0.63 | 27.90 | 0.14 | 0.76 | 9.23 |

Table 4. **Ablation on alignment-loss weight.** Increasing λ_{align} slightly degrades reconstruction (higher rFID / LPIPS, lower PSNR / SSIM) but improves generation quality (lower gFID), with the best trade-off at a moderate weight.

| Method | Alignment | Zero Init | Weight Scheduler | FID-10k ↓ |
|----------------------|-----------|-----------|------------------|-------------|
| Ours (full) | ✓ | ✓ | ✓ | 9.27 |
| w/o alignment | ✗ | ✓ | ✓ | 16.37 |
| w/o zero init | ✓ | ✗ | ✓ | 29.73 |
| w/o weight scheduler | ✓ | ✓ | ✗ | 9.80 |

Table 5. **Ablation on three components.** Our full model enables all three (✓); each ablation disables exactly one component (✗).

the same or even more tokens than ours, where our strategy shows better generation performance. In addition to quantitative results, we show qualitative results with rich details and complex structures on ImageNet in Fig. 5, demonstrating the effectiveness of our method.

4.2. Text-to-image results

We evaluate the text-to-image generation performance of our method by applying our recipe to Stable Diffusion 3.5 Medium, with a base resolution of 512 × 512 for generating images at 1024 × 1024. We compare to state-of-the-art methods for 1024 × 1024 image generation. As the results summarized in Tab. 3, the SD3.5 Medium fine-tuned with our DA-VAE models achieve similar quantitative results to their respective base models, but with an approximately 4× throughput improvement at 1K resolution. For qualitative comparison, we provide visual examples in Fig. 6, which shows that our method produces images with more complex structures, richer details, and better prompt fidelity than the 512 × 512 SD3.5-M baseline. We also examine scaling to 2048 × 2048. When naively running SD3.5-M at this resolution, the image quality degrades noticeably: large objects become distorted and the scene layout sometimes collapses. In contrast, our DA-VAE-augmented model generates high-quality 2048 × 2048 samples that preserve global structure

and fine details, as illustrated in Fig. 7.

4.3. Ablation studies

In this section, we conduct ablation studies to showcase different effects of the alignment loss weight, and validate our design choices by removing core components of our method one at a time. We conduct all ablation studies on the ImageNet for 512 × 512 class conditional generation based on the VA-VAE and LightningDiT-XL. All VAE models are trained for 50 epochs and all DiT models are trained for 20 epochs for ablation studies. For fair comparisons, we do not apply any classifier-free guidance during sampling.

Impact of alignment loss weights. We first study the effect of the alignment loss weight. As shown in Tab. 4, we vary the weights of the alignment losses λ_{align} in Eq. (6) when training the DA-VAE and report the reconstruction and generation performance of the resulting models. Without alignment ($\lambda_{align} = 0$), the model attains strong reconstruction fidelity but yields poor generation quality due to an unstructured latent space. With a small weight ($\lambda_{align} = 0.1$), generation quality improves substantially while reconstruction remains largely preserved. With a large weight ($\lambda_{align} = 1.0$), generation quality further improves, but reconstruction degrades due to over-regularization. We observe that a moderate weight of 0.5 achieves the best trade-off between reconstruction and generation quality. Therefore, we use $\lambda_{align} = 0.5$ for all other experiments.

Effectiveness of design choices. We then validate the effectiveness of our design choices in our method. As shown in Tab. 5, we conduct ablation studies on the three main components in our method: (a) the alignment loss between the detail latent and the base latent (b) the zero-initialization strategy; and (c) the loss scheduling. Among the three, we find that alignment and zero-init are crucial for effective generation, and the loss scheduling slightly improves results.

5. Conclusion and Limitations

Our method provides a simple and efficient recipe to increase the effective compression ratio of a pretrained VAE while keeping the token count unchanged, enabled by detail alignment. This approach does not require expensive

retraining. We showcase promising results for generic text-to-image tasks, where our method enables higher-resolution generation with the same number of visual tokens. However, our work has several limitations. First, we deliberately choose our current detail alignment loss due to its simplicity; there may be better alternatives. Second, given our compute budget, we have not yet evaluated full fine-tuning on SD3.5 or applied our method to more recent but costly backbones such as FLUX. Finally, as a proof-of-concept, our method currently uses synthetic data for fine-tuning. Therefore, our generated images at 1024×1024 are less photorealistic than SD3.5's native 1024×1024 generation. We leave these directions for future work.

References

- [1] Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, et al. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, 2024. 3
- [2] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A ViT backbone for diffusion models. In *CVPR*, 2023. 3
- [3] Black Forest Labs. Flux. <https://github.com/black-forest-labs/flux>, 2024. Accessed 2025-11-20. 3
- [4] Bowei Chen, Sai Bi, Hao Tan, He Zhang, Tianyuan Zhang, Zhengqi Li, Yuanjun Xiong, Jianming Zhang, and Kai Zhang. Aligning visual foundation encoders to tokenizers for diffusion models. *arXiv preprint arXiv:2509.25162*, 2025. 3
- [5] Hao Chen, Ze Wang, Xiang Li, Ximeng Sun, Fangyi Chen, Jiang Liu, Jindong Wang, Bhiksha Raj, Zicheng Liu, and Emad Barsoum. Softvq-vae: Efficient 1-dimensional continuous tokenizer. *arXiv preprint arXiv:2412.10958*, 2024. 3
- [6] Hao Chen, Yujin Han, Fangyi Chen, Xiang Li, Yidong Wang, Jindong Wang, Ze Wang, Zicheng Liu, Difan Zou, and Bhiksha Raj. Masked autoencoders are effective tokenizers for diffusion models. *arXiv preprint arXiv:2502.03444*, 2025. 3
- [7] Hansheng Chen, Kai Zhang, Hao Tan, Leonidas Guibas, Gordon Wetzstein, and Sai Bi. pi-flow: Policy-based few-step generation via imitation distillation. *arXiv preprint arXiv:2510.14974*, 2025. 3
- [8] Junyu Chen, Han Cai, Junsong Chen, Enze Xie, Shang Yang, Haotian Tang, Muyang Li, Yao Lu, and Song Han. Deep compression autoencoder for efficient high-resolution diffusion models. *arXiv preprint arXiv:2410.10733*, 2024. 2, 3
- [9] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart- σ : Weak-to-strong training of diffusion transformer for 4k text-to-image generation. *arXiv preprint arXiv:2403.04692*, 2024. 3
- [10] Junsong Chen, YU Jincheng, GE Chongjian, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart- α : Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *ICLR*, 2024. 3
- [11] Junyu Chen, Dongyun Zou, Wenkun He, Junsong Chen, Enze Xie, Song Han, and Han Cai. Dc-ae 1.5: Accelerating diffusion model convergence with structured latent space. *arXiv preprint arXiv:2508.00413*, 2025. 2, 3
- [12] Xiaoliang Dai, Ji Hou, Chih-Yao Ma, Sam Tsai, Jialiang Wang, Rui Wang, Peizhao Zhang, Simon Vandenhende, Xiaoofang Wang, Abhimanyu Dubey, et al. Emu: Enhancing image generation models using photogenic needles in a haystack. *arXiv preprint arXiv:2309.15807*, 2023. 3
- [13] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024. 2, 3, 5, 6
- [14] Gongfan Fang, Xinyin Ma, and Xinchao Wang. Structural pruning for diffusion models. In *NeurIPS*, 2024. 3
- [15] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. *arXiv preprint arXiv:2410.12557*, 2024. 3
- [16] Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025. 3
- [17] Dhruva Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In *NeurIPS*, 2023. 6
- [18] Wenkun He, Yuchao Gu, Junyu Chen, Dongyun Zou, Yujun Lin, Zhekai Zhang, Haocheng Xi, Muyang Li, Ligeng Zhu, Jincheng Yu, et al. Dc-gen: Post-training diffusion acceleration with deeply compressed latent space. *arXiv preprint arXiv:2509.25180*, 2025. 3
- [19] Wenkun He, Yuchao Gu, Junyu Chen, Dongyun Zou, Yujun Lin, Zhekai Zhang, Haocheng Xi, Muyang Li, Ligeng Zhu, Jincheng Yu, et al. Dc-gen: Post-training diffusion acceleration with deeply compressed latent space. *arXiv preprint arXiv:2509.25180*, 2025. 6, 7
- [20] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 7
- [21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 2
- [22] Dongwon Kim, Ju He, Qihang Yu, Chenglin Yang, Xiaohui Shen, Suha Kwak, and Liang-Chieh Chen. Democratizing text-to-image masked generative models with compact text-aware one-dimensional tokens. *arXiv preprint arXiv:2501.07730*, 2025. 3
- [23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023. 6
- [24] Black Forest Labs. Flux. *Online*, 2024. 3

- [25] Daiqing Li, Aleks Kamko, Ehsan Akhgari, Ali Sabet, Linmiao Xu, and Suhail Doshi. Playground v2. 5: Three insights towards enhancing aesthetic quality in text-to-image generation. *arXiv preprint arXiv:2402.17245*, 2024. 6
- [26] Muyang Li, Yujun Lin, Zhekai Zhang, Tianle Cai, Xiuyu Li, Junxian Guo, Enze Xie, Chenlin Meng, Jun-Yan Zhu, and Song Han. Svdquant: Absorbing outliers by low-rank components for 4-bit diffusion models. *arXiv preprint arXiv:2411.05007*, 2024. 2
- [27] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023. 3
- [28] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *CVPR*, 2024. 3
- [29] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *CVPR*, 2023. 3
- [30] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 3, 6
- [31] Xiangyu Peng, Zangwei Zheng, Chenhui Shen, Tom Young, Xinying Guo, Binluo Wang, Hang Xu, Hongxin Liu, Mingyan Jiang, Wenjun Li, et al. Open-sora 2.0: Training a commercial-level video generation model in \$200 k. *arXiv preprint arXiv:2503.09642*, 2025. 3
- [32] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 3
- [33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 3
- [34] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022. 2
- [35] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *ICLR*, 2022. 3
- [36] Minglei Shi, Haolin Wang, Wenzhao Zheng, Ziyang Yuan, Xiaoshi Wu, Xintao Wang, Pengfei Wan, Jie Zhou, and Jiwen Lu. Latent diffusion model without variational autoencoder. *arXiv preprint arXiv:2510.15301*, 2025. 3
- [37] Andy Shih, Suneel Belkale, Stefano Ermon, Dorsa Sadigh, and Nima Anari. Parallel sampling of diffusion models. In *NeurIPS*, 2024. 3
- [38] Stability AI. Sd3.5. <https://github.com/Stability-AI/sd3.5>, 2024. 2, 5, 1
- [39] Zhiwei Tang, Jiasheng Tang, Hao Luo, Fan Wang, and Tsung-Hui Chang. Accelerating parallel sampling of diffusion models. In *ICML*, 2024. 3
- [40] Jiannan Wang, Jiarui Fang, Aoyu Li, and PengCheng Yang. Pipefusion: Displaced patch pipeline parallelism for inference of diffusion transformer models. *arXiv preprint arXiv:2405.14430*, 2024. 3
- [41] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 2004. 7
- [42] Zijie J Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models. In *ACL*, 2023. 6
- [43] Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024. 2, 3
- [44] Enze Xie, Junsong Chen, Yuyang Zhao, Jincheng Yu, Ligeng Zhu, Yujun Lin, Zhekai Zhang, Muyang Li, Junyu Chen, Han Cai, et al. Sana 1.5: Efficient scaling of training-time and inference-time compute in linear diffusion transformer. *arXiv preprint arXiv:2501.18427*, 2025. 3
- [45] Qingsong Xie, Zhao Zhang, Zhe Huang, Yanhao Zhang, Haonan Lu, and Zhenyu Yang. Layton: Latent consistency tokenizer for 1024-pixel image reconstruction and generation by 256 tokens. *arXiv preprint arXiv:2503.08377*, 2025. 3
- [46] Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. Generation: Taming optimization dilemma in latent diffusion models. In *CVPR*, 2025. 2, 3, 6, 1
- [47] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. *arXiv preprint arXiv:2405.14867*, 2024. 3
- [48] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, 2024. 3
- [49] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, 2024. 2
- [50] Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. In *NeurIPS*, 2024. 3
- [51] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024. 6
- [52] Zhengrong Yue, Haiyu Zhang, Xiangyu Zeng, Boyu Chen, Chenting Wang, Shaobin Zhuang, Lu Dong, KunPeng Du, Yi Wang, Limin Wang, et al. Uniflow: A unified pixel flow tokenizer for visual understanding and generation. *arXiv preprint arXiv:2510.10575*, 2025. 3
- [53] Jinjin Zhang, Qiuyu Huang, Junjie Liu, Xiefan Guo, and Di Huang. Diffusion-4k: Ultra-high-resolution image synthesis with latent diffusion models. In *CVPR*, 2025. 2
- [54] Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. In *ICLR*, 2023. 3

- [55] Qinsheng Zhang, Molei Tao, and Yongxin Chen. gddim: Generalized denoising diffusion implicit models. In *ICLR*, 2023. 3
- [56] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 7
- [57] Tianchen Zhao, Tongcheng Fang, Enshu Liu, Wan Rui, Widyadewi Soedarmadji, Shiyao Li, Zinan Lin, Guohao Dai, Shengen Yan, Huazhong Yang, et al. Vedit-q: Efficient and accurate quantization of diffusion transformers for image and video generation. *arXiv preprint arXiv:2406.02540*, 2024. 3
- [58] Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. In *NeurIPS*, 2024. 3
- [59] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025. 3
- [60] Kaiwen Zheng, Cheng Lu, Jianfei Chen, and Jun Zhu. Dpm-solver-v3: Improved diffusion ode solver with empirical model statistics. In *NeurIPS*, 2023. 3
- [61] SUN Zhengwentai. clip-score: CLIP Score for PyTorch. <https://github.com/taited/clip-score>, 2023. Version 0.2.1. 6
- [62] Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. *arXiv preprint arXiv:2503.07565*, 2025. 3
- [63] Zixin Zhu, Xuelu Feng, Dongdong Chen, Jianmin Bao, Le Wang, Yinpeng Chen, Lu Yuan, and Gang Hua. Designing a better asymmetric vqgan for stablediffusion. *arXiv preprint arXiv:2306.04632*, 2023. 3

DA-VAE: Plug-in Latent Compression for Diffusion via Detail Alignment

Supplementary Material

This supplementary material provides implementation details and additional analyses. In particular, it

- Sec. S1 summarizes the training and sampling hyperparameters used in all experiments;
- Sec. S2 describes how to instantiate DA-VAE on top of a pretrained VAE tokenizer, using SD3-VAE as a concrete example;
- Sec. S3 and S4 verify that the decoder and the diffusion backbone actually make use of the extra detail latent channels, rather than ignoring them.
- Sec. S5 presents additional qualitative results for DA-VAE enhanced Stable Diffusion 3.5 Medium.

S1. Training and sampling hyperparameters

Tab. S1 lists the optimization and sampling configurations used in all our experiments.

For ImageNet class-to-image experiments with LightningDiT-XL, we largely follow the training recipe of Yao et al. [46] and only adjust the learning rate, batch size and loss weights to accommodate our higher-compression DA-VAE latent space. DA-VAE is trained with AdamW and a relatively small KL weight, while the alignment loss λ_{align} is set to a moderate value to balance reconstruction and generation quality.

For SD3.5-M text-to-image experiments, we use a smaller batch size during DA-VAE training due to the higher input resolution and slightly different loss weights ($\lambda_L, \lambda_1, \lambda_{\text{adv}}, \lambda_{\text{KL}}, \lambda_{\text{align}}$) to stabilize high-resolution reconstruction.

During DiT fine-tuning, we apply the gradual loss scheduling described in the main paper: for the first N_{warm} steps (10k for LightningDiT-XL and 5k for SD3.5-M), the detail-latent loss is down-weighted so that optimization is initially dominated by the original base latent, and then gradually forced to model the new detail channels. We maintain an exponential moving average (EMA) of the DiT parameters with decay 0.999 for all runs.

For sampling, we use 250 diffusion steps with a classifier-free guidance scale of 4.0 and the CFG schedule from [46] on ImageNet, and 30 steps with a guidance scale of 2.5 for SD3.5-M, as summarized in Tab. S1.

| Stage | Hyper-parameter | lightningDiT-XL [46], Class-to-image | SD3.5-M [38], Text-to-image |
|-------------------------|--|--------------------------------------|-----------------------------|
| DA-VAE Training | learning rate | 1e-4 | 1e-4 |
| | batch size | 128 | 16 |
| | training steps | 100K | 10K |
| | optimizer | AdamW, betas=[0.5, 0.9] | AdamW, betas=[0.9, 0.999] |
| | loss weights ($\lambda_L, \lambda_1, \lambda_{\text{adv}}, \lambda_{\text{KL}}, \lambda_{\text{align}}$) | (1.0, 1.0, 0.1, 1e-6, 0.5) | (1.0, 2.0, 0.1, 1e-7, 1.0) |
| DiT Fine-Tuning | learning rate | 2e-4 | 1e-4 |
| | Gradual loss scheduling steps | 10K | 5K |
| | batch size | 640 | 128 |
| | training steps | 140K | 10K |
| | optimizer | AdamW, betas=[0.9, 0.95] | AdamW, betas=[0.9, 0.999] |
| | EMA decay | 0.999 | 0.999 |
| Sampling for Generation | # sampling steps | 250 | 30 |
| | CFG / guidance scale | 4.0 | 2.5 |
| | CFG interval start | 0.2 | - |
| | timestep shift | 0.3 | - |

Table S1. Training and sampling hyperparameters for lightningDiT-XL and SD3.5-M.

S2. DA-VAE architecture

Fig. S1 illustrates how we turn a SD3-VAE into DA-VAE. We *borrow* the overall encoder and decoder backbone architectures from SD3-VAE, but remove their original feature-to-latent and latent-to-feature heads and replace them with our own downsampling and upsampling blocks. The resulting encoder E_f and decoder D_f are therefore retrained as part of DA-VAE.

Concretely, the SD3-VAE encoder produces an intermediate feature map $F \in \mathbb{R}^{512 \times H \times W}$. In the original SD3-VAE, a shallow head directly maps F to a latent $z_{\text{sd3}} \in \mathbb{R}^{16 \times H \times W}$. In our design, we discard this head and instead attach a small downsampling module that further reduces the spatial resolution of F while keeping the channel dimension fixed (e.g., a stack of strided 3×3 conv blocks). This yields a more compressed base latent $z_d \in \mathbb{R}^{16 \times (H/s) \times (W/s)}$. Then we concat the

detail latent z_d with the base latent z which is the original SD3-VAE feature of the downsampled base image. Together, these form the structured latent (z, z_d) used by our DiT.

The decoder side is modified in a symmetric way. Instead of feeding the original SD3-VAE latent z_{sd3} into a latent-to-feature stem, we concatenate our base and detail latents along the channel dimension and first apply a lightweight upsampling block (e.g., pixel shuffle) that inverts the encoder’s spatial downsampling. A 3×3 convolution then maps the upsampled latent back to a $512 \times H \times W$ feature map, which is passed through the SD3-VAE decoder backbone D_f for reconstruction.

In summary, DA-VAE keeps the deep convolutional backbone structure of SD3-VAE but replaces its shallow latent heads with our own downsampling/upsampling design, enabling a higher-compression latent space with an explicit separation between base and detail channels. All components, including the reused backbone blocks, are trained end-to-end under our DA-VAE objective.

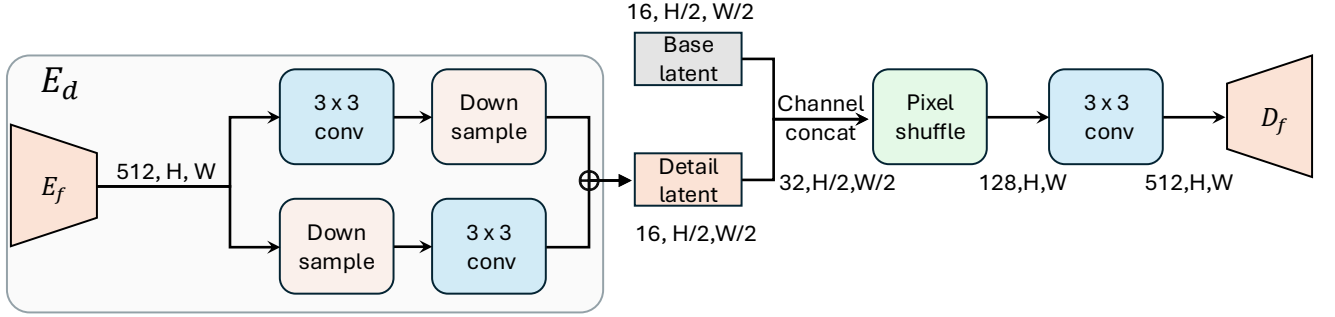
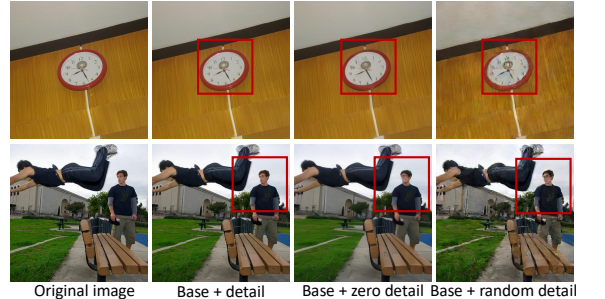


Figure S1. **DA-VAE architecture instantiated on SD3-VAE.** We reuse the convolutional encoder E_f and decoder D_f blocks from SD3-VAE, but remove its original feature-to-latent and latent-to-feature heads. Instead, a lightweight downsampling module maps the shared $512 \times H \times W$ feature map to a more compressed detail latent z_d and a parallel base latent z of the same shape, while a symmetric upsampling module concatenates (z, z_d) , upsamples them back to $512 \times H \times W$, and feeds the result into the reused decoder backbone. This yields a higher-compression latent space with explicit base and detail channels, while keeping most of the VAE architecture intact.

S3. Decoder sensitivity to the detail latent

| Decoder variant | Reconstruction (ImageNet val) | | | |
|-----------------------------|-------------------------------|--------------|-------------|-------------|
| | rFID ↓ | PSNR ↑ | LPIPS ↓ | SSIM ↑ |
| Full (base + detail) | 0.47 | 28.53 | 0.12 | 0.78 |
| Base + random detail | 8.25 | 23.67 | 0.30 | 0.62 |
| Base + zero detail | 2.93 | 24.71 | 0.25 | 0.63 |

(a) Reconstruction metrics on the ImageNet validation set.



(b) Example reconstructions on ImageNet. Best for zoom-in view.

Figure S2. **Ablation on detail channels in the DA-VAE decoder on ImageNet.** (a) Reconstruction metrics for different decoder variants. (b) Visual examples showing that randomizing or zeroing the detail latent either destroys the image or removes fine-grained details such as faces and text. Please zoom in for best view.

We evaluate the sensitivity of the decoder to the detail latent on the ImageNet validation set. Starting from a trained DA-VAE, we fix the base latent z and modify the detail latent z_d in two ways: (i) we add z_d with i.i.d. Gaussian noise in $\mathcal{N}(0, I)$ (*Base + random detail*); and (ii) we set z_d to zero (*Base + zero detail*). The quantitative reconstruction metrics on ImageNet are summarized in Fig. S2a, and representative reconstructions are visualized in Fig. S2b.

Randomizing z_d leads to clearly invalid reconstructions with high rFID, low PSNR, and severe artifacts such as distorted faces and unreadable text, indicating that the decoder cannot simply ignore the detail channels. Zeroing z_d produces structurally plausible but over-smoothed images: edges become soft and fine textures disappear, and details around faces and characters are lost. In contrast, the full model using both z and z_d recovers both global structure and high-frequency details.

These observations confirm that the learned detail latent encodes semantically meaningful fine-grained information. Consequently, during fine-tuning the DiT must also learn to generate z_d correctly; otherwise the final high-resolution samples would lack sharp details even if the base latent is well modeled.

S4. Training dynamics of SD3.5-M fine-tuning

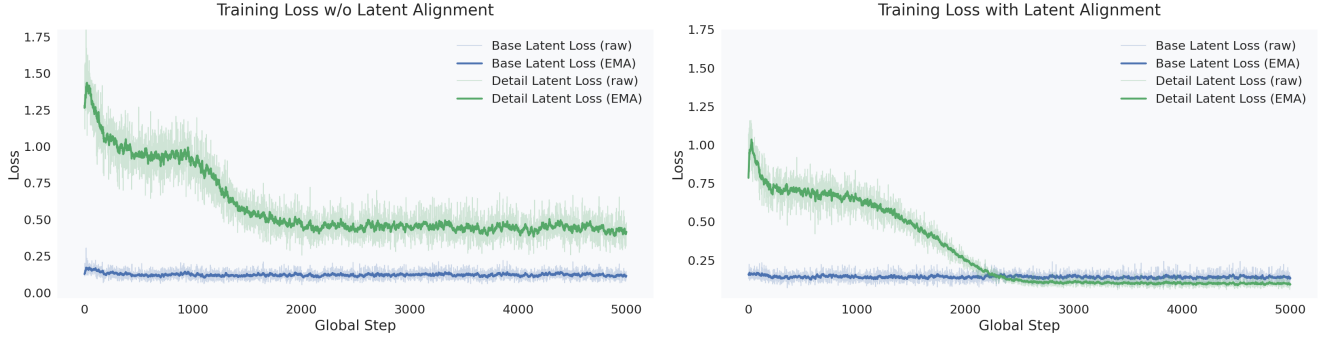


Figure S3. **Training loss curves for SD3.5-M fine-tuning with and without latent alignment.** We plot the unweighted diffusion loss on the base latent (blue) and the detail latent (green) when fine-tuning SD3.5-M with DA-VAE to 1024×1024 resolution, showing both the raw loss (faint) and its exponential moving average (solid). *Left:* without latent alignment, the detail-latent loss decreases slowly and remains significantly higher than the base-latent loss. *Right:* with alignment, optimization is more stable and converges to a lower-loss solution; the detail-latent loss eventually falls below the base-latent loss, indicating that the DiT has learned a well-structured distribution over the extra detail channels and can effectively exploit them.

Fig. S3 visualizes the optimization behaviour when fine-tuning SD3.5-M from 512×512 to 1024×1024 resolution with our DA-VAE. We plot the *unweighted* diffusion loss on the base latent and on the detail latent, i.e., the true per-token MSE before applying the scheduling weight $w(n)$ described in the main paper. For each branch we show both the raw loss and its exponential moving average (EMA).

Two trends are worth noting. First, in both settings the base-latent loss stays relatively low throughout training, while the detail-latent loss starts much higher and gradually decreases. This reflects that fine-tuning primarily teaches the model how to predict the new detail channels, whereas the base latent already carries a well-trained prior from the original SD3.5-M.

Second, comparing the left and right plots shows the effect of latent alignment. Without alignment, the detail-latent loss decays slowly and plateaus at a relatively high value, indicating that the model struggles to capture a coherent distribution over z_d . With alignment, the loss on z_d decreases steadily and eventually falls below the base-latent loss, meaning that the detail channels become easier to model than the base ones. This behaviour is consistent with our quantitative and qualitative results: latent alignment guides the fine-tuning towards a lower-loss solution in which the extra detail latent is well structured and effectively used for high-resolution generation.

S5. Additional qualitative results

To further demonstrate the effectiveness of our method, Fig. S4 presents additional qualitative results of DA-VAE-enhanced Stable Diffusion 3.5 Medium (SD3.5-M) for text-to-image generation. To improve realism, we further fine-tune SD3.5-M with our model for 5K steps on 500K images generated by Flux [3] using prompts collected by [7].



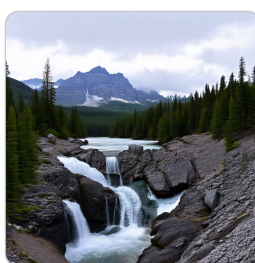
"A crocheted hot air balloon with a unicorn in a basket, hanging against a plain wall with a tree branch"



"A fairytale castle with multiple towers and turrets, surrounded by water with rocks and a bridge leading to it"



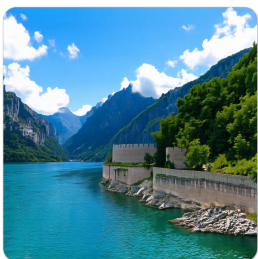
"Woman in bright blue suit serving green soup from a pitcher into jars at an event table with sunflowers"



"Serene mountain landscape with Falls, water cascading over rocks, evergreen forest and distant peaks"



"Cozy window view of fjord and snowy mountain, red house, dark floral curtains, coffee pot on sill"



"A serene view of the Skadar River with the Kotor fortress walls in the foreground"



"A vibrant garden scene potted yellow tulips and boxwood shrubs on a wooden table in front of a greenhouse"



"A picturesque canal scene with half-timbered houses lining the waterway, vibrant flowers in hanging baskets"



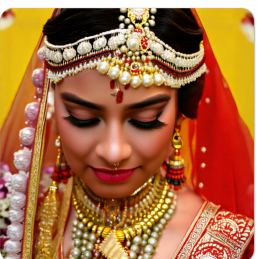
"An outdoor scene featuring a white wooden chair with a blue and white vase holding yellow flowers on its seat"



"Black-and-white rugged landscape, dead tree on rocky cliff above lake and mountains"



"A highly detailed pencil drawing of a bearded man with a prominent mustache"



"Close-up Indian bride, ornate gold jewelry, red sari, realistic"



"Oneeyed Beric Dondelion stayed to fight the White Walkers alone"



"A person in black top hat with silver details and pocket watch, ornate black-and-white face paint"



"A close-up portrait of a person with voluminous, curly, light brown hair styled in loose waves"



"Knickerbocker glory, tall thin glass, icecream, raspberries, couli, blueberries, pistachio, icing sugar"



"A colorful, lively image of a birthday cake, decorated with healthy ingredients such as fruits, nuts, and seeds"



"A stack of fried green tomatoes topped with creamy jalapeño pimiento cheese"



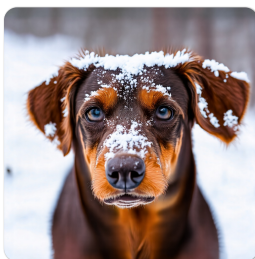
"Tree of life, 3d globe, metallic, epic, cinematic, nature"



"Two dogs dressed in Halloween costumes stand on a porch with pumpkins and a scarecrow"



"Dungeons and dragons, standing in a decayed forest, gray-white eyes, holding a glowing blue druid focus"



"Close-up of a brown dog with blue eyes, snow on its face, in a snowy background, realistic"



"Oil painting kinkde Disney, beautiful award winning Arabian stallion in full costume with lots of tassels"



"A wolf stands alert on a mossy rock, with a backdrop of vibrant red berries and lush green foliage"

Figure S4. Generated examples by our DA-VAE enhanced SD3.5-M. Please zoom in for best view.