DC-SAE: Deep Compression Semantic Autoencoder for Faster Diffusion Convergence

Xu Huang[1*] **Ye Huang**[1*] **Zijun Liao**[1*] **Yuwei Niu**[1] **Xiaojie Li**[1]
Menghan Zhou[2] **De Wen Soh**[2] **Xiaotong Li**[1] **Daquan Zhou**[1†]

[1]Peking University
[2]Singapore University of Technology and Design
[*]Equal contribution. [†]Corresponding author.

Abstract

High-compression tokenizers are essential for scaling latent image generative models. However, aggressive compression creates a fundamental tradeoff between reconstruction fidelity and generation efficiency: high compression image encoder always increases the learning difficulty of diffusion training, resulting in slow model convergence. Recent representation autoencoders speed up the diffusion training by improving the latent feature's expressive capability by replacing VAE encoders with pretrained semantic encoders, yet they are typically limited to moderate compression and lose pixel-level details necessary for faithful reconstruction. To achieve both high compression and fast diffusion training, we propose DC-SAE, a Decoupled Compact Semantic Autoencoder designed for high-compression image generation with accelerated diffusion model convergence. DC-SAE consists of two key components: (1) a macro-level architecture design that leverages semantic encoders to enable higher compression ratios, and (2) a pixel-level encoder that preserves low-level details, ensuring high-fidelity image reconstruction. We empirically demonstrate that DC-SAE performs strongly on image generation tasks, achieving both compact latent representations and efficient training dynamics. Specifically, on the ImageNet dataset with 512×512 resolution, DC-SAE achieves $32\times$ spatial compression, with **29.79** PSNR and **3.37** gFID, substantially outperforming the previous state-of-the-art high-compression tokenizer baselines DC-AE by 13.5% and 54.9% on PSNR and gFID, respectively, maintaining comparable throughput and faster diffusion model training convergence. Beyond class-conditional generation, a 1.6B-parameter DiT using DC-SAE achieves **0.84** on GenEval and **86.007** on DPG-Bench for text-to-image generation at 1024×1024 resolution. Code is available at `https://github.com/DAGroup-PKU/DCSAE`.

1 Introduction

High-compression visual tokenizers such as DC-AE [6, 7] are crucial because they substantially reduce the latent sequence length and computational cost of all latent-based generative models. However, despite their high reconstruction quality, the resulting latent spaces are often less favorable for diffusion model training in terms of convergence speed: although high-compression tokenizers reduce the cost of each forward pass, diffusion models typically require more training steps to converge, which can ultimately reduce the benefits for the overall training cost.

We are therefore motivated to ask: how can we design a tokenizer that achieves both a high input compression ratio and fast diffusion model convergence? To explore this question, we draw inspiration from recent representation autoencoders (RAEs) [30, 50, 52, 37], which have been shown to enable faster diffusion model training convergence than conventional variational autoencoders (VAEs). However, naively increasing the compression ratio of pretrained semantic encoders such as DINOv2

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

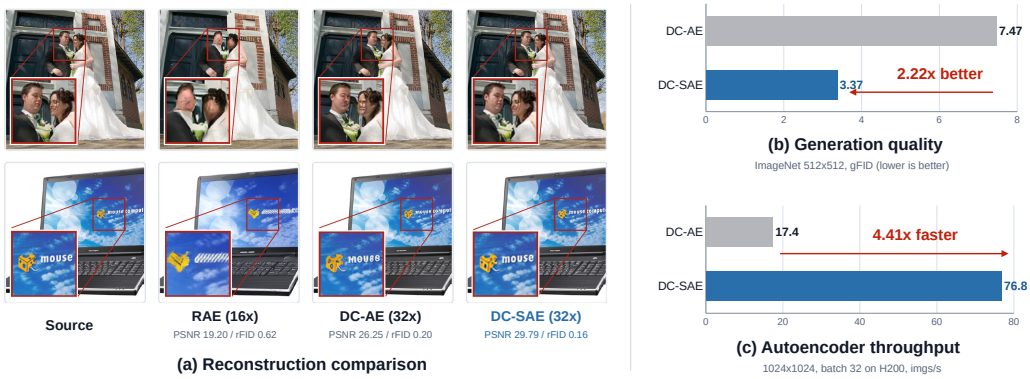

Figure 1: Overview of DC-SAE under $32\times$ spatial compression. (a) DC-SAE preserves reconstruction details missed by RAE and improves over DC-AE. (b) On ImageNet 512×512, DC-SAE achieves better gFID than DC-AE. (c) DC-SAE also provides substantially higher autoencoder throughput.

[30] to a deep-compression regime, e.g., $32\times$, substantially degrades the reconstruction quality of RAEs, rendering them impractical for high-fidelity generation.

Through a detailed analysis, we find that a potential root cause of the poor reconstruction performance of RAE-like models is their lack of low-level visual information, as semantic encoders primarily capture high-level abstract representations of input images.

We therefore propose a novel dual-stream encoder design, consisting of a semantic encoder for extracting high-level features and an unconstrained pixel-level encoder for preserving low-level visual information. The high-level semantic features capture the major components of visual inputs and map visually similar objects to similar latent representations, yielding a more compact and structured latent space that accelerates diffusion model convergence. Meanwhile, the low-level pixel features compensate for the missing fine-grained details in semantic representations, enabling high-quality reconstruction even under a high compression ratio. The new scheme was termed DC-SAE. DC-SAE defines a new family of autoencoders that enjoys both a high-compression ratio and faster diffusion model convergence. To further improve the reconstruction quality of DC-SAE, we introduce a new module termed Spatial DeMerger. It expands the compact fused latent only on the decoder side, and thus keeps the generator-facing sequence short while providing denser spatial conditioning for high-fidelity reconstruction.

We conduct extensive experiments to validate the effectiveness of DC-SAE. Under the challenging $32\times$ compression regime, DC-SAE consistently outperforms the previous high-compression tokenizer DC-AE in both reconstruction and generation quality. Specifically, DC-SAE achieves 29.79 PSNR and 3.37 gFID, compared with 26.25 PSNR and 7.47 gFID obtained by DC-AE. Beyond quality improvements, DC-SAE also delivers substantially higher end-to-end throughput. As shown in Tab. 7, DC-SAE achieves $4.64\times$, $4.75\times$, and $4.41\times$ higher throughput than DC-AE at 256×256, 512×512, and 1024×1024 resolutions, respectively. These results demonstrate that DC-SAE simultaneously improves reconstruction fidelity, generative performance, and computational efficiency under aggressive latent compression. We also extend the evaluation to text-to-image generation with a 1.6B-parameter DiT and a Qwen3-1.7B text encoder. At 1024×1024 resolution with classifier-free guidance (CFG), this model achieves 0.84 on GenEval and 86.007 on DPG-Bench, demonstrating the applicability of DC-SAE beyond class-conditional ImageNet generation.

We further provide an in-depth analysis of the proposed semantic–pixel dual-stream design. Under the $32\times$ compression setting, DiT trained on the joint semantic–pixel latent space converges substantially faster than models trained on either semantic-only or pixel-only latents. Visualization results Fig. 5 further reveal that the pixel branch complements the semantic branch by recovering fine-grained local appearance details that are largely absent from high-level semantic representations. Together, these findings suggest that combining semantic compactness with pixel-level fidelity is key to building high-compression tokenizers that remain both reconstruction-friendly and diffusion-friendly. Our contributions are summarized as follows:

- **High-compression semantic encoders.** We propose a simple, yet extremely effective way to adapt the semantic autoencoders from $16\times$ to $32\times$ compression, without re-training them. This makes it

possible to utilize the pre-trained semantic encoders such as series works of DINOs [4, 30, 38] and CLIPs [34, 50].

- **Fast diffusion model training convergence.** We propose a novel dual-stream encoder mechanism that achieves $32\times$ latent compression while maintaining intrinsically fast convergence for diffusion model training. Unlike prior methods that rely on explicit and often complex latent-space constraints—such as VFM alignment [44], self-supervised token losses [21], or channel masking [7]—our approach improves convergence through the architectural design of the tokenizer itself. Moreover, it is complementary to these existing techniques and can potentially benefit from incorporating them further.
- **State-of-the-art deep compression autoencoder.** By combining these designs, we successfully train DC-SAE, which establishes a new state of the art in both reconstruction quality and diffusion model training efficiency. On ImageNet, DC-SAE improves PSNR by 13.5% and FID by 54.9%, while achieving a $4.41\times$ speedup over the previous state-of-the-art deep-compression method.

2 Related work

2.1 Representation encoders and unified tokenizers for generation

Pretrained visual representation encoders have recently been adopted as alternatives to conventional VAE encoders for latent generation. Self-supervised and vision-language models such as DINOv2, SigLIP, CLIP, and MAE provide semantically structured features that are useful for recognition and alignment [30, 50, 34, 13]. Representation autoencoders (RAEs) further show that replacing the reconstruction-only VAE encoder with a pretrained representation encoder can produce semantically meaningful latents and accelerate diffusion transformer training [52]. However, these encoders are not originally designed for pixel-faithful reconstruction or generative tokenization. Their objectives often discard texture, precise local geometry, color statistics, and high-frequency details that are weakly relevant for semantic understanding but crucial for reconstruction. Complementary efforts explore semantic priors and latent-space regularization to improve latent representations and downstream generation [8, 44, 48, 24, 31].

Another related line of work pursues unified tokenizers for both visual understanding and generation [33, 26, 39, 47, 28]. TokenFlow and UniTok improve discrete token spaces through decoupled or multi-codebook designs [33, 26], while UniLiP and UniFlow adapt semantic features for reconstruction and generation via self-distillation or pixel-level decoding mechanisms [39, 47]. LatentUM further studies unified multimodal modeling in a shared semantic latent space [17]. Recent continuous or general-purpose visual tokenizers, such as UniFluid, Ming-UniVision, and AToken, also explore unified representations for visual understanding and generation across images, videos, or multimodal tasks [9, 16, 25].

The closest work to ours is SVG, which augments frozen DINO features with a lightweight residual branch to recover missing perceptual details for latent diffusion [37]. While SVG shows that VFM features can serve as a generative latent space, our focus is different: we study how to convert a frozen semantic encoder into a high-compression visual tokenizer. Under deep compression, reconstruction becomes especially challenging because the compact semantic latent has limited spatial capacity. DC-SAE therefore introduces an unconstrained pixel encoder to complement the frozen semantic branch with local appearance information, without relying on SVG-style self-distillation or semantic regularization. Despite this simple design, DC-SAE improves reconstruction over SVG and recent regularized representation-tokenizer methods such as UniLiP [39], while retaining the fast generation convergence of semantic latents.

2.2 Deep-compression visual tokenizers

Visual tokenizers compress images into compact latent representations for efficient generative modeling [35]. As image generation scales to higher resolutions and longer videos, deep-compression tokenizers have become increasingly important because the latent sequence length directly determines the cost of diffusion or transformer-based generators. DC-AE and DC-AE 1.5 push continuous autoencoders to much higher spatial compression ratios through residual autoencoding, decoupled high-resolution adaptation, and structured latent spaces [6, 7]. TexTok uses text descriptions to provide high-level semantic information during tokenization, allowing image tokens to focus more on

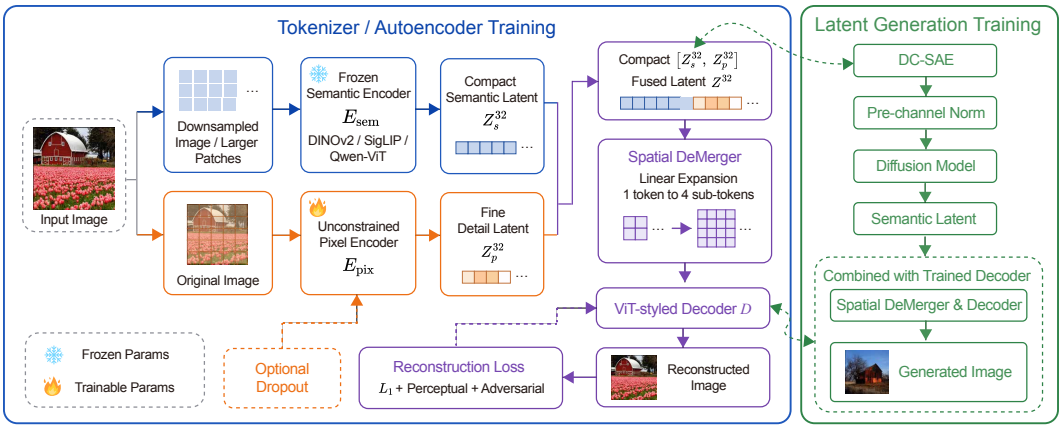

Figure 2: Pipeline of DC-SAE. We first adapt a frozen semantic encoder to a 32× token grid. Since the resulting semantic autoencoder suffers from poor reconstruction under aggressive compression, we introduce an unconstrained pixel branch to preserve local image details. The semantic and pixel branches are fused and decoded by a ViT decoder with Spatial Demerger.

fine visual details under small token budgets [49]. MAETok shows that masked modeling can improve latent-space structure for diffusion models without relying on variational constraints [5]. TC-AE studies deep compression from the perspective of token capacity, using staged token compression and self-supervised token structuring to reduce representation collapse [21]. Recent image and video tokenizers further explore high-compression design from different perspectives, including efficient video autoencoding in H3AE, neural image/video tokenization in Cosmos Tokenizer, diffusion-guided decoding in DGAE, and high-fidelity discrete tokenization in WeTok [43, 29, 23, 53]. Other recent works, such as FlexTok, GloTok, and TokBench, study adaptive token allocation, global semantic structure, and tokenizer evaluation for visual generation [2, 51, 42].

These methods reveal a common reconstruction–generation dilemma: increasing latent capacity can improve reconstruction, but may also make the latent distribution harder for downstream generators to learn [7, 21]. Our work addresses this dilemma in the setting of semantic autoencoders. Instead of requiring a single compact latent to preserve both semantics and pixel detail, DC-SAE uses a decoupled design: a compact semantic branch provides the generator-facing representation, while an unconstrained pixel branch supplies local appearance information needed for high-fidelity reconstruction. The two branches are concatenated and expanded by a Spatial DeMerger before decoding, which restores a denser spatial layout without requiring the generator to model a dense latent sequence. This allows us to extend semantic autoencoders beyond the common 16× regime toward higher compression, while preserving the generation efficiency benefits of compact semantic latents.

3 Method

Our goal is to build a high-compression visual tokenizer that preserves the fast convergence property of semantic representations while achieving faithful image reconstruction. Recent semantic autoencoders show that pretrained semantic encoders provide structured latent spaces that are highly effective for diffusion training. However, most existing semantic autoencoders operate at the standard 16× spatial compression ratio. To improve throughput, we first investigate whether such semantic autoencoders can be directly scaled to 32× compression.

3.1 Limitations of Semantic Autoencoders under Deep Compression Regime

Semantic encoders are trained to preserve high-level visual structure, such as object identity, scene layout, and semantic relations. These structures are usually stable under moderate image resizing. Motivated by this observation, we adopt a simple strategy to construct a 32× semantic autoencoder from a standard semantic encoder. Specifically, we resize the input image to half resolution, which is equivalent to using a larger effective patch on the original image. The resized image is then patchified, linearly projected, and fed into the frozen semantic encoder. This directly produces a compact 32× semantic latent.

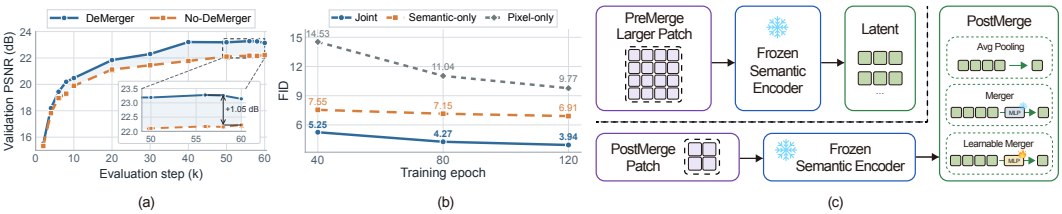

Figure 3: (a) Validation PSNR comparison between DeMerger and No-DeMerger variants. (b) DiT training convergence on joint semantic–pixel, semantic-only, and pixel-only latents under $32\times$ compression. (c) Illustration of pre-merge and post-merge semantic compression for constructing a $32\times$ semantic tokenizer.

The resulting semantic autoencoder is trained with the standard reconstruction objective:

$$\mathcal{L}_{\text{rec}} = \lambda_1 \|\mathbf{x} - \hat{\mathbf{x}}\|_1 + \lambda_{\text{perc}} \mathcal{L}_{\text{perc}}(\mathbf{x}, \hat{\mathbf{x}}) + \lambda_{\text{adv}} \mathcal{L}_{\text{adv}}, \tag{1}$$

where $\mathcal{L}_{\text{perc}}$ denotes perceptual loss and $\mathcal{L}_{\text{adv}}$ denotes the optional adversarial loss.

However, we observe that this direct strategy is insufficient for constructing an effective high-compression tokenizer. While resizing largely preserves the global semantic structure, it substantially degrades the reconstruction fidelity. At 256×256 resolution, the resulting $32\times$ semantic autoencoder achieves only around 15 PSNR. More importantly, the learned latent no longer exhibits the fast diffusion convergence typically expected from pretrained semantic representations. After 80 epochs of DiT training, it reaches only 7.15 gFID.

One may suspect that this failure comes from the specific input-level resizing design rather than semantic compression itself. To rule out this possibility, we further test common MLLM-style post-merge compression strategies, such as average pooling or merging already-extracted semantic tokens. As discussed in Sec. 4.4, these alternatives perform even worse than the resizing-based pre-merge strategy. This suggests that the bottleneck is not merely the choice of compression implementation.

We attribute this failure to the objective mismatch of semantic autoencoding. A pretrained semantic encoder is not trained for invertible reconstruction. It keeps information useful for recognition and representation learning, but discards many pixel-level details, such as color statistics, textures, local boundaries, and high-frequency patterns. When the compression ratio is increased from $16\times$ to $32\times$, this information loss is further amplified. Therefore, a semantic branch alone is insufficient for high-compression visual tokenization.

3.2 Semantic-guided Unconstrained AutoEncoder

The failure of the direct $32\times$ semantic autoencoder shows that semantic latents alone are insufficient for high-fidelity reconstruction under aggressive compression. A straightforward way to improve reconstruction is to use an unconstrained, high-channel autoencoder, since a larger latent capacity can preserve more pixel-level information. However, prior works have shown that such reconstruction-oriented latents are often difficult for diffusion models to learn. To improve latent generability, existing methods usually introduce additional constraints or regularization, such as KL regularization, representation alignment, structured latent design, or channel masking [19, 35, 44, 7]. This reflects a common trade-off: increasing autoencoder capacity improves reconstruction, but may hurt diffusion convergence.

Our key motivation is that this trade-off can be alleviated by semantic guidance. Recent works suggest that semantic representation latents can provide useful structure for diffusion training, and Latent Forcing further indicates that semantic latents can guide pixel-level generation trajectories [1]. Therefore, we hypothesize that, even under high compression, a frozen semantic encoder can serve as a structural guide for an unconstrained high-channel pixel autoencoder. Instead of constraining the pixel latent itself, we pair it with a semantic latent and jointly train the two branches inside a single autoencoder.

As shown in Fig. 2, DC-SAE introduces an unconstrained pixel encoder in parallel with the frozen $32\times$ semantic branch. The semantic branch provides a compact and structured latent space that captures object-level layout and semantic relations. The pixel branch is trained from scratch and focuses on reconstruction-critical details, such as texture, color statistics, local boundaries, and high-frequency information. The two latents are concatenated along the channel dimension, fused by a lightweight projection, and decoded jointly. In this way, the final latent space combines the

Table 1: Comparison of $16\times$ semantic autoencoders on reconstruction quality and ImageNet 256×256 generation after 80 epochs. DC-SAE-16x achieves strong reconstruction while retaining competitive generation efficiency, making it a flexible baseline for high-compression tokenizer design. *Notice:* The $16\times$ DC-SAE setting is used only in this ablation; all other DC-SAE results in this paper use the $32\times$ setting.

Metric	RAE [52]	SD-VAE [35]	VA-VAE [44]	SVG [37]	REPA [46]	**DC-SAE-16x**
Reconstruction						
PSNR ↑	19.20	24.08	**27.96**	23.89	–	<u>27.80</u>
rFID ↓	0.62	0.87	<u>0.28</u>	0.65	–	**0.26**
Generation						
gFID ↓	**2.16**	–	5.96	6.57	7.90	<u>3.09</u>
IS ↑	**214.80**	–	128.00	137.90	118.60	<u>189.95</u>

generative structure of semantic representations with the reconstruction capacity of unconstrained pixel features. Since the pixel branch is directly optimized for reconstruction, the decoder may over-rely on it and under-utilize the frozen semantic branch. We therefore apply sample-wise dropout to the pixel latent during tokenizer training, forcing the decoder to use semantic structure when the pixel branch is dropped and to treat the pixel branch as complementary detail when it is present.

This decomposition leads to a more suitable latent space for high-compression diffusion training. As shown in Fig. 3(b), the joint semantic–pixel latent converges faster than both semantic-only and pixel-only alternatives, reaching 4.27 gFID after 80 epochs compared with 7.15 and 11.04, respectively. This indicates that neither branch alone is sufficient: semantic features provide a structured, diffusion-friendly representation but lose reconstruction-critical details, whereas pixel latents preserve such details but are harder to model at high channel capacity. By combining them, DC-SAE obtains a latent representation that supports both faithful reconstruction and efficient generative modeling.

After tokenizer training, the entire tokenizer is frozen, and DiT is trained on the fused latent. Because the semantic and pixel branches are jointly optimized through the same decoder, the resulting latent is more generation-friendly than simply concatenating an independently trained pixel autoencoder with semantic features.

3.3 ViT Decoder with Spatial Demerger

The semantic–pixel latent introduced above improves the information content of the tokenizer under $32\times$ compression. However, the compact latent grid also creates a decoder-side spatial bottleneck. For diffusion modeling, a sparse $32\times$ grid is desirable because it significantly reduces the sequence length. For reconstruction, however, this grid is less suitable for a ViT decoder, which relies on spatial tokens and attention to recover local image details. A denser decoder-side token grid can thus provide more spatial positions for modeling textures, boundaries, and high-frequency structures.

To address this mismatch, we introduce Spatial DeMerger before the ViT decoder. Spatial DeMerger expands each compact latent token into a small group of spatial sub-tokens before decoding. Importantly, this expansion is applied only on the decoder side. The DiT still operates on the original compact $32\times$ compressed latent, so the efficiency benefit of high compression is preserved, while the decoder receives a denser grid for high-fidelity reconstruction.

This design decouples the latent grid used for diffusion modeling from the token grid used for image decoding. The compact grid keeps DiT training and sampling efficient, whereas the expanded decoder-side grid improves the reconstruction ability of the ViT decoder. In Sec. 4.4, we further provide ablation studies showing that Spatial DeMerger consistently improves reconstruction quality under high compression.

4 Experiments

4.1 Experimental Setting

We evaluate DC-SAE on class-conditional ImageNet generation at both 256×256 and 512×512 resolutions, as well as text-to-image generation at 1024×1024 resolution. For ImageNet tokenizer training, we first train DC-SAE at 256×256 resolution and then finetune it at 512×512 resolution under the target spatial compression ratio. Detailed tokenizer training settings are provided in Appendix A.2. After tokenizer training, the tokenizer is frozen, and a latent diffusion/transformer

Table 2: Comparison with state-of-the-art image generative models on ImageNet 512×512 class-conditional generation. DC-SAE uses a $32\times$ high-compression semantic tokenizer and trains a DiT-XL generator in the learned latent space.

ImageNet 512×512								
Image Generative Model	Autoencoder	Compression Rate	Params (B)	gFID ↓		Inception Score ↑	PSNR ↑	rFiD ↓
				w/o CFG	w/ CFG			
DiT-XL [32]	Flux-VAE-f8c16 [20]	8×	0.68	27.35	8.72	53.09	–	–
DiT-XL [32]	SD-VAE-f8c4 [35]	8×	0.67	12.03	3.04	105.25	–	–
SiT-XL [27]	SD-VAE-f8c4 [35]	8×	0.67	–	2.62	–		
MAGVIT-v2 [45]	–	–	–	3.07	1.91	213.10	–	–
MAR-L [22]	–	–	–	2.74	1.73	205.20	–	–
EDM2-XXL [18]	–	–	–	1.91	1.81	–	–	–
SiT-XL [27]	DC-AE-f32c32 [6]	32×	0.67	7.47	2.41	131.37		
USiT-H [6]	DC-AE-f32c32 [6]	32×	0.50	3.80	1.89	174.58	26.25	0.20
USiT-2B [6]	DC-AE-f32c32 [6]	32×	1.58	2.90	1.72	187.68		
DiT-XL + DDT wide head [41]	**DC-SAE**	**32×**	0.83	**3.37**	1.99	**205.83**	**29.79**	**0.16**



Figure 4: Qualitative results on ImageNet. The images are generated with 512×512 resolutions in the latent space of DC-SAE under $32\times$ compression, with randomly sampled class conditions.

generator is trained in the learned latent space. For ImageNet diffusion training, we follow the training protocol of DC-AE [6]. We report reconstruction quality using rFID, and ImageNet generation quality using gFID and Inception Score. Text-to-image training and evaluation are detailed in Sec. 4.3.

4.2 Main Result

ImageNet 256×256 **generation** We first compare DC-SAE with DC-AE and DC-AE-1.5 under the ImageNet 256×256 setting. This setting evaluates whether the proposed semantic–pixel decoupled tokenizer can improve generation quality under the same high-compression regime. As shown in Tab. 5, DC-SAE improves the reconstruction rFID from 0.69 to 0.45 and achieves a gFID of 3.31 with an Inception Score of 180.71 using a DiT-XL generator. Compared with DC-AE and DC-AE-1.5 under the same DiT-XL backbone, DC-SAE provides substantially better generation quality, showing that the proposed tokenizer produces a more generation-friendly latent space while preserving reconstruction fidelity. These results suggest that DC-SAE is already a strong high-compression tokenizer at 256×256 resolution. Although our generator uses the standard DiT-XL backbone rather than the specialized USiT architecture, it achieves generation quality comparable to or better than stronger DC-AE baselines, while also improving reconstruction rFID. Following RAE, we also use a DiT-XL variant with a DDT-wide head for diffusion training [41].

ImageNet 512×512 **generation** We further evaluate DC-SAE on ImageNet 512×512 class-conditional generation. The tokenizer is finetuned at 512×512 resolution under a $32\times$ spatial compression ratio, and a DiT-XL generator is trained in the frozen DC-SAE latent space for 400 epochs following the DC-AE training setting. As shown in Tab. 2, DC-SAE achieves a gFID of 3.37 without CFG under the 512×512 setting, demonstrating that the proposed tokenizer scales to higher-resolution generation while maintaining a compact latent representation. We provide qualitative generation samples in Fig. 4.

4.3 Text-to-Image Generation

Experimental setup. To evaluate DC-SAE beyond class-conditional ImageNet generation, we train a text-to-image model with a 1.6B-parameter DiT backbone and Qwen3-1.7B [40] as the text

Table 3: Text-to-image generation results with CFG. The DC-SAE model is evaluated at 1024×1024 resolution. Generator sizes and tokenizer compression ratios are listed explicitly; the baselines use different model and training configurations. Higher scores are better for both benchmarks.

Image Generative Model	Autoencoder	Compression Rate	Params (B)	GenEval ↑	DPG-Bench ↑
DC-Gen-FLUX.1-Krea-12B [14]	DC-AE f32c32 [6]	32×	12	0.72	**87.073**
DC-Gen-FLUX.1-Krea-12B [14]	DC-AE-1.5 f64c128 [7]	64×	12	0.59	75.439
1.6B-DiT	**DC-SAE**	32×	1.6	**0.84**	86.007

Table 4: Comparison between post-merge and pre-merge strategies for constructing a 32× semantic tokenizer.

Semantic Encoder	PSNR ↑	rFID ↓	gFID ↓	IS ↑
Post-merge				
Qwen-ViT Frozen Merger	23.40	0.62	25.96	66.83
Qwen-ViT AvgPool Merger	23.07	0.61	17.71	84.45
Qwen-ViT Learnable Merger	23.85	0.58	16.84	78.34
DINOv2 Learnable Merger	24.03	0.35	8.60	108.11
Pre-merge				
DINOv2	23.13	0.45	**3.31**	**180.71**

encoder, using DC-SAE at 32× spatial compression. Training uses a subset of internal data together with LAION-COCO [36]. We progressively pretrain the model at 256×256 resolution for 100K iterations with a batch size of 1024, followed by another 100K iterations at 512×512 with the same batch size. We then perform supervised fine-tuning (SFT) at 1024×1024 resolution for 40K iterations with a batch size of 256. The final model is evaluated with classifier-free guidance (CFG) on GenEval [10] and DPG-Bench [15], which assess compositional text-to-image alignment and dense-prompt following, respectively.

Results. As shown in Tab. 3, our model achieves 0.84 on GenEval, exceeding the DC-AE and DC-AE-1.5-based DC-Gen-FLUX.1-Krea-12B baselines by 0.12 and 0.25, respectively. On DPG-Bench, our model obtains 86.007, compared with 87.073 for the DC-AE baseline and 75.439 for the DC-AE-1.5 baseline. Thus, the 1.6B-parameter generator achieves the highest GenEval score among these models and remains competitive on DPG-Bench against the 12B DC-AE-based generator. These results support the applicability of DC-SAE to open-ended text-to-image generation with a compact latent grid.

4.4 Ablation Studies

We conduct several ablation studies to understand the four design choices of DC-SAE. Specifically, we ask: whether the proposed semantic–pixel decoupled design is effective under the standard 16× setting, how to adapt pretrained semantic encoders to the 32× token grid, whether Spatial DeMerger is necessary for high-compression ViT decoding, and whether the gain comes from joint semantic–pixel autoencoding rather than simply exposing DiT to semantic features.

Validation under 16× compression. We first evaluate the proposed semantic–pixel decoupled design under the standard 16× setting. This setting isolates the effect of the pixel branch, since pretrained semantic encoders already provide a sufficiently dense token grid at 16× compression. As shown in Tab. 1, DC-SAE substantially improves reconstruction quality over semantic-only autoencoders. Compared with RAE, which obtains around 19 PSNR, DC-SAE reaches 27.80 PSNR and 0.26 rFID. Meanwhile, its latent space remains effective for diffusion training, achieving strong ImageNet 256×256 generation performance after 80 epochs. This result confirms the basic motivation of our design. The frozen semantic branch provides a structured latent space that benefits generation, but it lacks reconstruction-critical details. Adding a trainable pixel branch closes most of this reconstruction gap without destroying the diffusion-friendly structure of the semantic latent.

Pre-merge versus post-merge semantic compression. We next compare different ways of adapting pretrained semantic encoders to 32× compression. Motivated by recent MLLM efforts on high-compression vision tokenizers, a straightforward approach is to first extract dense visual tokens and then reduce the token count through local token merging. As shown in Fig. 3(c). In our setting, this

Table 5: Comparison with DC-AE and DC-AE-1.5 on ImageNet 256×256 class-conditional generation.

ImageNet 256×256					
Diffusion Model	Setting	Autoencoder	rFID ↓	gFID ↓	Inception Score ↑
DiT-XL [32]	f32c32	DC-AE [6]	0.69	10.18	107.49
DiT-XL [32]	f32c128	DC-AE [6]	0.26	26.44	53.41
DiT-XL [32]	f32c32	DC-AE-1.5 [7]	–	10.50	107.99
DiT-XL [32]	f32c128	DC-AE-1.5 [7]	0.26	17.31	80.38
DiT-XL [32]	**f32c64**	**DC-SAE (Ours)**	**0.45**	**5.67**	**156.68**
USiT-H [6]	f32c32	DC-AE [6]	0.69	3.89	167.40
USiT-H [7]	f32c32	DC-AE-1.5 [7]	–	4.15	166.94
DiT-XL [32]	f32c128	TC-AE [21]	0.35	7.16	-
DiT-XL + DDT wide head [41]	**f32c64**	**DC-SAE (Ours)**	**0.45**	**3.31**	**180.71**

corresponds to first extracting dense $16\times$ semantic tokens and then merging each local token group into a compact $32\times$ token. We refer to this family of methods as *post-merge compression*, since compression is applied after semantic feature extraction. We evaluate several variants, including average pooling [12, 11], frozen merger modules [3], and learnable mergers. Although these methods are simple and commonly used for reducing visual token count, they are not ideal for generative tokenization. As shown in Tab. 4, post-merge variants consistently underperform. For example, Qwen-ViT, with its frozen merger, obtains poor generation quality, and even learnable merger variants remain limited. This indicates that reducing tokens after semantic abstraction discards information that is difficult to recover later. In contrast, the resizing-based strategy used in Sec. 3.1 compresses the input before semantic abstraction. We refer to this strategy as *pre-merge compression*, since the semantic encoder directly forms representations under the target $32\times$ token budget rather than compressing already-extracted dense semantic tokens. Empirically, this strategy leads to better generation quality, showing that compression should be introduced before semantic encoding.

Effectiveness of Spatial DeMerger. Finally, we evaluate the role of Spatial DeMerger in the decoder. At $32\times$ compression, the latent grid is highly compact, which is favorable for diffusion throughput but challenging for reconstruction. This issue is particularly important for ViT decoders, since they benefit from a denser spatial token grid for local detail recovery.

As shown in Fig. 3(a), under the standard DINOv2 pre-merge setting on ImageNet 256×256 with $32\times$ compression, adding Spatial DeMerger consistently improves validation PSNR over the No-DeMerger variant. This supports our motivation that the compact diffusion latent and the decoder-side token grid should be treated differently: the diffusion model should operate on the sparse $32\times$ latent for efficiency, while the decoder benefits from a denser grid for high-fidelity reconstruction. Additional results with QwenViT-based semantic encoders are provided in Appendix A.4.

Joint semantic–pixel training versus late concatenation. A natural question is whether the gain of DC-SAE comes from joint semantic–pixel autoencoding, or simply from exposing DiT to semantic features. To isolate this factor, we compare two settings in Tab. 6: *Joint* and *Late-Concat*. *Joint* is the DC-SAE baseline, where the decoder reconstructs images from the concatenated semantic and pixel latents. Thus, the pixel branch is learned together with the semantic branch inside the autoencoder. In contrast, *Late-Concat* first trains a pixel autoencoder alone, whose decoder only receives the pixel latent. The frozen DINOv2 semantic latent is concatenated with the pixel latent only later, during DiT training. Therefore, DiT receives semantic features in both settings, but only *Joint* learns a coupled semantic–pixel latent space during autoencoding.

Table 6: Standalone autoencoder ablation. Entries are FID / IS after DiT training.

Metric	Joint	Late-Concat	Late-Concat-E
PSNR ↑	29.79	31.91	29.15
rFID ↓	0.17	0.09	0.17
20 ep.	8.12 / 134.23	10.95 / 122.48	10.35 / 127.78
40 ep.	5.23 / 166.76	7.41 / 153.16	6.91 / 156.97
60 ep.	4.33 / 180.31	6.22 / 168.13	6.01 / 167.48
80 ep.	4.02 / 185.46	5.64 / 174.52	5.40 / 173.00

As shown in Tab. 6, late concatenation does not match the joint training variant. Although *Late-Concat* achieves better reconstruction quality than *Joint*, it gives worse generation performance, with an 80-epoch FID of 5.64 versus 4.02. To rule out a reconstruction–generation conflict, *Late-Concat-E* uses an earlier checkpoint of the same pixel autoencoder, whose reconstruction quality is closer to *Joint*. It still underperforms, reaching an 80-epoch FID of 5.40. These results indicate that DC-SAE's gain is not merely from adding semantic features to DiT, but from jointly training the decoder to use semantic and pixel latents as a compatible representation for generation.

4.5 Discussion: Understanding Component Contributions to Reconstruction Fidelity

We further analyze where the reconstruction ability of DC-SAE comes from. Since DC-SAE contains both a semantic branch and a pixel branch, the decoder may rely on either branch for reconstruction. To study this, we reconstruct images using the full latent, the latent with the semantic component masked, and the latent with the pixel component masked.

As shown in Figure 5, masking out the semantic component has little effect on reconstruction quality, while masking out the pixel component causes severe degradation. This suggests that high-fidelity reconstruction mainly comes from the pixel branch. The semantic branch, in contrast, primarily provides high-level structure and a generation-friendly representation space rather than detailed appearance information.

This finding is consistent with the design of DC-SAE. The pixel branch preserves local texture, color, and spatial details for reconstruction, while the semantic branch guides the joint latent toward a structure that is easier for DiT to learn. Thus, DC-SAE does not require the frozen semantic encoder to carry all low-level information; instead, reconstruction and generative structure are handled by two complementary branches.

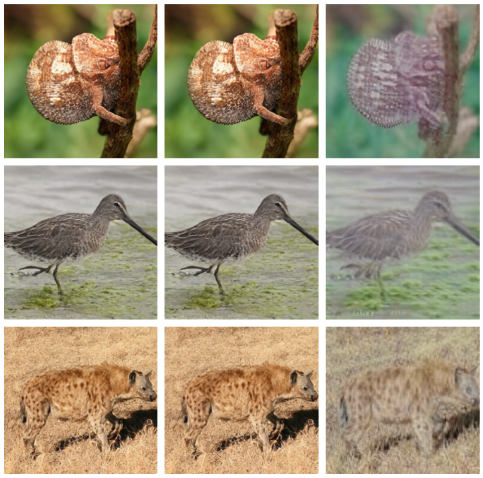

Figure 5: Investigating where DC-SAE's reconstruction ability comes from. We compare three reconstruction settings: using the full latent, masking the semantic component, and masking the pixel component. Masking the semantic branch has little impact on reconstruction quality, whereas masking the pixel branch causes severe degradation. This indicates that DC-SAE primarily relies on the pixel branch for reconstruction.

5 Conclusion

We propose DC-SAE, a decoupled compact semantic autoencoder for high-compression latent image generation. DC-SAE addresses the reconstruction–generation tradeoff by combining a frozen semantic encoder for generation-friendly structure with an unconstrained pixel branch for low-level reconstruction details. We further identify post-hoc semantic token merging as a key obstacle to extending semantic autoencoders to $32\times$ compression, as it discards spatial and low-level details after semantic abstraction. We address this with input-level semantic compression and a decoder-side Spatial DeMerger, enabling high-compression semantic tokenization with strong reconstruction fidelity. Experiments on ImageNet show that DC-SAE achieves strong reconstruction quality, faster diffusion convergence, and competitive generation performance, reaching 29.79 PSNR and 3.37 gFID at 512×512 resolution with $32\times$ compression. Text-to-image experiments further demonstrate its applicability beyond ImageNet: a 1.6B-parameter DiT using DC-SAE achieves 0.84 on GenEval and 86.007 on DPG-Bench at 1024×1024 resolution with CFG.

References

[1] Alan Baade, Eric Ryan Chan, Kyle Sargent, Changan Chen, Justin Johnson, Ehsan Adeli, and Li Fei-Fei. Latent forcing: Reordering the diffusion trajectory for pixel-space image generation, 2026. URL https://arxiv.org/abs/2602.11401.

[2] Roman Bachmann, Jesse Allardice, David Mizrahi, Enrico Fini, Oğuzhan Fatih Kar, Elmira Amirloo, Alaaeldin El-Nouby, Amir Zamir, and Afshin Dehghan. Flextok: Resampling images into 1d token sequences of flexible length. *arXiv preprint arXiv:2502.13967*, 2025.

[3] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[5] 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. In *International Conference on Machine Learning*, 2025.

[6] Junyu Chen, Han Cai, Junsong Chen, Enze Xie, Shang Yang, Haotian Tang, Muyang Li, and Song Han. Deep compression autoencoder for efficient high-resolution diffusion models. In *International Conference on Learning Representations*, 2025.

[7] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. arXiv:2508.00413.

[8] Yufeng Cui, Honghao Chen, Haoge Deng, Xu Huang, Xinghang Li, Jirong Liu, Yang Liu, Zhuoyan Luo, Jinsheng Wang, Wenxuan Wang, Yueze Wang, Chengyuan Wang, Fan Zhang, Yingli Zhao, Ting Pan, Xianduo Li, Zecheng Hao, Wenxuan Ma, Zhuo Chen, Yulong Ao, Tiejun Huang, Zhongyuan Wang, and Xinlong Wang. Emu3.5: Native multimodal models are world learners. *arXiv preprint arXiv:2510.26583*, 2025.

[9] Lijie Fan, Luming Tang, Siyang Qin, Tianhong Li, Xuan Yang, Siyuan Qiao, Andreas Steiner, Chen Sun, Yuanzhen Li, Tao Zhu, et al. Unified autoregressive visual generation and understanding with continuous tokens. *arXiv preprint arXiv:2503.13436*, 2025.

[10] Dhruba Ghosh, Hanna Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2310.11513.

[11] Google DeepMind. Gemma 4. https://deepmind.google/models/gemma/gemma-4/, 2026. Accessed: 2026-05-02.

[12] Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, et al. Seed1. 5-vl technical report. *arXiv preprint arXiv:2505.07062*, 2025.

[13] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[14] Wenkun He, Yuchao Gu, Junyu Chen, Junyi Wu, Wenhang Ge, Dongyun Zou, Yujun Lin, Zhekai Zhang, Haocheng Xi, Muyang Li, et al. Dc-gen: Post-training diffusion acceleration with deeply compressed latent space. In *European Conference on Computer Vision*, pages 249–269. Springer, 2026.

[15] Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. ELLA: Equip diffusion models with LLM for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024. URL https://arxiv.org/abs/2403.05135.

[16] Ziyuan Huang, Dandan Zheng, Cheng Zou, Rui Liu, Xiaolong Wang, Kaixiang Ji, Weilong Chai, Jianxin Sun, Libin Wang, Yongjie Lv, et al. Ming-univision: Joint image understanding and generation with a unified continuous tokenizer. *arXiv preprint arXiv:2510.06590*, 2025.

[17] Jiachun Jin, Zetong Zhou, Xiao Yang, Hao Zhang, Pengfei Liu, Jun Zhu, and Zhijie Deng. Latentum: Unleashing the potential of interleaved cross-modal reasoning via a latent-space unified model. *arXiv preprint arXiv:2604.02097*, 2026.

[18] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proc. CVPR*, 2024.

[19] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. URL https://arxiv.org/abs/1312.6114.

[20] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024.

[21] Teng Li, Ziyuan Huang, Cong Chen, Yangfu Li, Yuanhuiyi Lyu, Dandan Zheng, Chunhua Shen, and Jun Zhang. TC-AE: Unlocking token capacity for deep compression autoencoders. *arXiv preprint arXiv:2604.07340*, 2026.

[22] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization, 2024. URL https://arxiv.org/abs/2406.11838.

[23] Dongxu Liu, Yuang Peng, Haomiao Tang, Yuwei Chen, Chunrui Han, Zheng Ge, Daxin Jiang, and Mingxue Liao. Dgae: Diffusion-guided autoencoder for efficient latent representation learning. *arXiv preprint arXiv:2506.09644*, 2025.

[24] Shizhan Liu, Xinran Deng, Zhuoyi Yang, Jiayan Teng, Xiaotao Gu, and Jie Tang. Delving into latent spectral biasing of video VAEs for superior diffusability. *arXiv preprint arXiv:2512.05394*, 2025.

[25] Jiasen Lu, Liangchen Song, Mingze Xu, Byeongjoo Ahn, Yanjun Wang, Chen Chen, Afshin Dehghan, and Yinfei Yang. Atoken: A unified tokenizer for vision. *arXiv preprint*, 2025.

[26] Chuofan Ma, Yi Jiang, Junfeng Wu, Jihan Yang, Xin Yu, Zehuan Yuan, Bingyue Peng, and Xiaojuan Qi. Unitok: A unified tokenizer for visual generation and understanding. *arXiv preprint arXiv:2502.20321*, 2025.

[27] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pages 23–40. Springer, 2024.

[28] Meituan LongCat Team. LongCat-Next: Lexicalizing modalities as discrete tokens. *arXiv preprint arXiv:2603.27538*, 2026. URL https://arxiv.org/abs/2603.27538.

[29] NVIDIA. Cosmos tokenizer: A suite of image and video neural tokenizers. *arXiv preprint arXiv:2501.03575*, 2025.

[30] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[31] Yueming Pan, Ruoyu Feng, Qi Dai, Yuqi Wang, Wenfeng Lin, Mingyu Guo, Chong Luo, and Nanning Zheng. Semantics lead the way: Harmonizing semantic and texture modeling with asynchronous latent diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[32] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[33] Liao Qu, Huichao Zhang, Yiheng Liu, Xu Wang, Yi Jiang, Yiming Gao, Hu Ye, Daniel K. Du, Zehuan Yuan, and Xinglong Wu. Tokenflow: Unified image tokenizer for multimodal understanding and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

[35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[36] Christoph Schuhmann, Andreas Köpf, Richard Vencu, Theo Coombes, and Romain Beaumont. LAION COCO: 600M synthetic captions from LAION2B-en. `https://laion.ai/blog/laion-coco/`, 2022.

[37] 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. In *International Conference on Learning Representations*, 2026. arXiv:2510.15301.

[38] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

[39] Hao Tang, Chenwei Xie, Xiaoyi Bao, Tingyu Weng, Pandeng Li, Yun Zheng, and Liwei Wang. Unilip: Adapting clip for unified multimodal understanding, generation and editing. *arXiv preprint arXiv:2507.23278*, 2025.

[40] Qwen Team. Qwen3 technical report, 2025. URL `https://arxiv.org/abs/2505.09388`.

[41] Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. Ddt: Decoupled diffusion transformer, 2025. URL `https://arxiv.org/abs/2504.05741`.

[42] Junfeng Wu, Dongliang Luo, Weizhi Zhao, Zhihao Xie, Yuanhao Wang, Junyi Li, Xudong Xie, Yuliang Liu, and Xiang Bai. Tokbench: Evaluating your visual tokenizer before visual generation. *arXiv preprint arXiv:2505.18142*, 2025.

[43] Yushu Wu, Yanyu Li, Ivan Skorokhodov, Anil Kag, Willi Menapace, Sharath Girish, Aliaksandr Siarohin, Yanzhi Wang, and Sergey Tulyakov. H3ae: High compression, high speed, and high quality autoencoder for video diffusion models. *arXiv preprint arXiv:2504.10567*, 2025.

[44] Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. *arXiv preprint arXiv:2501.01423*, 2025.

[45] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023.

[46] 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.

[47] Zhengrong Yue, Haiyu Zhang, Xiangyu Zeng, Boyu Chen, Chenting Wang, Shaobin Zhuang, Lu Dong, Yi Wang, Limin Wang, and Yali Wang. Uniflow: A unified pixel flow tokenizer for visual understanding and generation. *arXiv preprint arXiv:2510.10575*, 2025.

[48] Zhengrong Yue, Taihang Hu, Mengting Chen, Haiyu Zhang, Zihao Pan, Tao Liu, Zikang Wang, Jinsong Lan, Xiaoyong Zhu, Bo Zheng, and Yali Wang. What matters for diffusion-friendly latent manifold? prior-aligned autoencoders for latent diffusion. *arXiv preprint arXiv:2605.07915*, 2026.

[49] Kaiwen Zha, Lijun Yu, Alireza Fathi, David A. Ross, Cordelia Schmid, Dina Katabi, and Xiuye Gu. Language-guided image tokenization for generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[50] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023.

[51] Xuan Zhao, Zhongyu Zhang, Yuge Huang, Yuxi Mi, Guodong Mu, Shouhong Ding, Jun Wang, Rizen Guo, and Shuigeng Zhou. Glotok: Global perspective tokenizer for image reconstruction and generation. *arXiv preprint arXiv:2511.14184*, 2025.

[52] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025.

[53] Shaobin Zhuang, Yiwei Guo, Canmiao Fu, Zhipeng Huang, Zeyue Tian, Zhang Ying, Chen Li, and Yali Wang. Wetok: Powerful discrete tokenization for high-fidelity visual reconstruction. *arXiv preprint arXiv:2508.05599*, 2025.

A Appendix

A.1 Limitations and Future Work.

Our evaluation covers class-conditional ImageNet generation and text-to-image generation at 1024×1024 resolution. However, the text-to-image study uses one 1.6B DiT configuration and a subset of internal data together with LAION-COCO; broader validation across training scales, datasets, and generator architectures remains future work. In addition, while we observe that semantic latents can accelerate diffusion training and that input-level pre-merge compression outperforms post-hoc token merging, we do not yet have a complete understanding of which properties of semantic encoders are most important for generation. A deeper analysis of semantic encoder selection, as well as a principled explanation of when pre-merge or post-merge compression is preferable, remains an important direction for future work.

A.2 Tokenizer Training Details

Datasets. We use ImageNet-1K for tokenizer training. Most experiments are conducted at 256×256 resolution. For 512×512 generation, we initialize the tokenizer from the trained 256×256 checkpoint and further finetune it on 512×512 images.

Tokenizer Architecture. Our tokenizer contains a frozen semantic encoder, a trainable pixel encoder, a fusion projection, and a ViT decoder with Spatial DeMerger. The semantic encoder is always frozen. The pixel encoder, fusion projection, Spatial DeMerger, and decoder are trained for reconstruction. The decoder architecture and discriminator largely follow RAE [52].

Training Configuration. We follow the basic tokenizer training setting of RAE [52], including the decoder and discriminator training setup. We set the L1 loss weight to $\lambda_1 = 1$,LPIPS loss weight to $\lambda_{\text{lpips}} = 1$ and the adversarial loss weight to $\lambda_{\text{GAN}} = 0.5$. At 256×256 resolution, we train the tokenizer for 30 epochs with a batch size of 512. For 512×512 resolution, we initialize from the converged 256×256 checkpoint and continue training for 20 epochs with a batch size of 512.

A.3 Throughput Comparison with DC-AE

As shown in Tab. 7, we compare the end-to-end throughput of DC-AE and DC-SAE under the same settings. All throughput numbers are measured with a batch size of 32 on a single NVIDIA H200 GPU using bf16, 32 batch-sizes with 10 warmup iterations and 50 measurement iterations. DC-SAE consistently achieves higher end-to-end throughput than the corresponding DC-AE baseline across all evaluated resolutions, with the speedup remaining above $4\times$ at every resolution.

Table 7: End-to-end throughput comparison between DC-AE and DC-SAE (imgs/s).

Model	256×256	512×512	1024×1024
DC-AE-f32c32	265.6	69.7	17.4
DC-SAE-f32c64	**1233.5**	**330.8**	**76.8**
DC-SAE-f32c256	1054.5	284.4	67.9

A.4 Additional Analysis of Spatial DeMerger

To further demonstrate the effectiveness of Spatial DeMerger, we conduct additional experiments with QwenViT-based semantic encoders. In particular, we consider two common vision-encoder configurations used in MLLM settings. The first one uses the original frozen QwenViT with its native merger module. This follows the design used in Qwen-VL, where the vision encoder consists of a $16\times$ patch-based QwenViT followed by a 2×2 merger trained together with the MLLM. The second one replaces the original QwenViT merger with a simple 2×2 average pooling operation, which is another commonly used way to reduce the spatial resolution of vision tokens in MLLM vision encoders.

In all experiments, the semantic encoder is kept fully frozen, and no additional pixel encoder is used, unlike the settings studied in the main paper. This setup isolates the role of Spatial DeMerger under a

stricter reconstruction setting, where the decoder must recover spatial details only from the frozen semantic representation. As shown in Tab. 8, Spatial DeMerger consistently improves reconstruction quality for both the original QwenViT merger and the 2×2 average pooling variant. These results suggest that Spatial DeMerger is especially useful for RAE-like settings, where the encoder is fully frozen, and the decoder receives only sparse semantic tokens without auxiliary pixel-level features.

Table 8: Additional analysis of Spatial DeMerger with QwenViT-based semantic encoders. We compare the original QwenViT merger and a 2×2 average pooling variant, with and without Spatial DeMerger. Both semantic encoders are frozen, and no additional pixel encoder is used.

Semantic encoder setting	Spatial DeMerger	
	w/o DeMerger	w/ DeMerger
QwenViT w/ original merger	15.39	17.52
QwenViT w/ 2×2 avg pooling	15.50	17.64

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses the limitations of the work performed by the authors.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper fully discloses all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Code and data will be released after the paper is accepted.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: For each experiment, the paper provides sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators or original owners of assets (e.g., code, data, models), used in the paper, are properly credited and the license and terms of use are explicitly mentioned and properly respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: Code and data will be released after the paper is accepted.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.