DecQ: Detail-Condensing Queries for Enhanced Reconstruction and Generation in Representation Autoencoders

Tianhang Wang[1,2,*] **Yitong Chen**[2,3,*] **Wei Song**[1,2,4]

Zuxuan Wu[2,3,†] **Min Li**[1,†] **Jiaqi Wang**[2,5,†]

[1]Zhejiang University [2]Shanghai Innovation Institute [3]Fudan University
[4]Westlake University [5]JD.COM

Abstract

Representation Autoencoders (RAEs) leverage frozen vision foundation models (VFMs) as tokenizer encoders, providing robust high-level representations that facilitate fast convergence and high-quality generation in latent diffusion models. However, freezing the VFM inherently constrains its spatial reconstruction capacity, limiting fine-grained generation and image editing; in contrast, incorporating reconstruction-oriented signals via fine-tuning disrupts the pretrained semantic space and degrades generative fidelity. To address this trade-off, we propose **DecQ**, a simple yet effective framework for RAEs. Specifically, DecQ introduces lightweight detail-condensing queries that extract fine-grained information from intermediate VFM features through condenser modules. These queries are incorporated into the decoder to support reconstruction and are jointly generated with patch tokens during generative modeling. By aggregating information from both shallow and deep layers, DecQ effectively mitigates the reconstruction–generation trade-off, improving both reconstruction quality and generative performance. Our experiments demonstrate that: (1) with only 8 additional queries and 3.9% extra computation, DecQ improves reconstruction over the frozen DINOv2-based RAE, increasing PSNR from 19.13 dB to 22.76 dB; and (2) for generative modeling, DecQ matches RAE's 800-epoch gFID using 3.3× fewer training epochs, and attains an FID of 1.41 without guidance and 1.05 with guidance.

1 Introduction

In visual generation, state-of-the-art diffusion models [1–3] are typically built upon a two-stage training paradigm: first learning a tokenizer, and then training a generative model in the resulting latent space, where the tokenizer is usually implemented as an autoencoder. Recently, Representation Autoencoders (RAEs) [4] revisit this design by replacing the tokenizer encoder with frozen pretrained vision foundation models (VFMs) [5, 6] and training only an additional decoder. Their results demonstrate that the semantically rich latent space induced by such pretrained representations can substantially accelerate the convergence of diffusion models.

Despite their advantages, directly using frozen VFMs as image tokenizers introduces a clear objective mismatch. Existing VFMs are typically trained with multimodal alignment [7, 8, 6] or

*Equal contribution. † Corresponding authors. Project page: https://github.com/Tianhang-Wang/DecQ

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

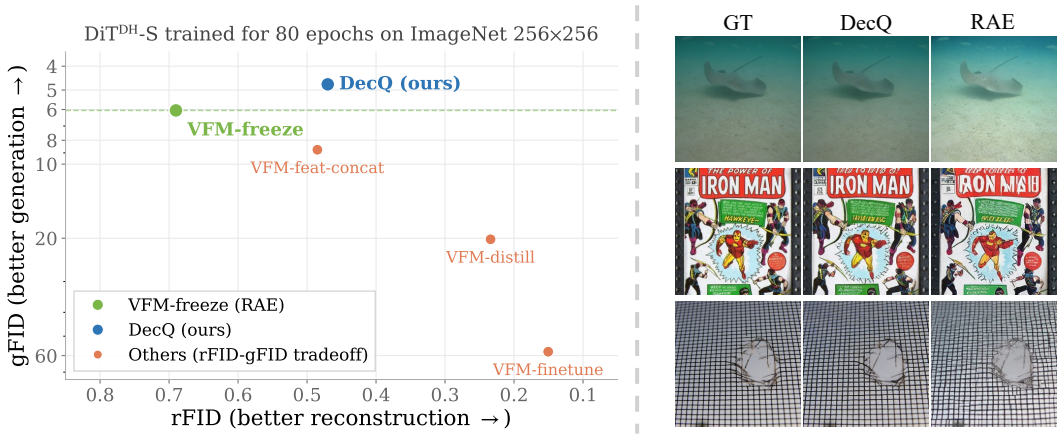


Figure 1: **(Left) An empirical study of different VFM-based image tokenizer paradigms based on DINOv2.** *VFM-freeze*, corresponding to the RAE baseline, keeps the VFM encoder frozen and directly uses its representations for reconstruction. *VFM-finetune* denotes directly fine-tuning the VFM encoder, *VFM-distill* uses a frozen VFM copy to distill the encoder outputs, and *VFM-feat-concat* keeps the VFM frozen while concatenating low-level information along the feature dimension. These variants improve reconstruction quality but reveal a clear trade-off between rFID and gFID, leading to degraded generative performance. In contrast, DecQ improves reconstruction while also enhancing generation compared with RAE, demonstrating that detail-condensing queries can enrich low-level details without compromising the VFM semantic latent space. Detailed results are provided in Tab. 3. **(Right) Reconstruction examples of our DecQ framework.** Our reconstruction preserves more color and texture information than RAE, enabling more accurate recovery of background colors and better reconstruction of textual content and fine-grained textures.

self-distillation [5, 9] objectives, rather than explicit pixel-level reconstruction losses [10–12]. These objectives often encourage invariance across augmented views [5, 6], which improves semantic robustness but may reduce sensitivity to low-level cues such as color and texture [13–15]. As a result, frozen VFM latent representations are not well suited to serve as information-preserving image codes. When used as frozen encoders, their limited preservation of low-level details can lead to reconstruction artifacts such as texture loss and color shifts, as shown in Fig. 1 Right. Due to the limited invertibility of such frozen representations, RAE models built upon frozen VFMs may exhibit weaker fine-grained generation and editing capabilities. In other words, although RAEs often enable faster convergence, their ultimate generative performance can be substantially constrained by suboptimal reconstruction fidelity.

To address this challenge, a straightforward strategy is to inject more low-level, reconstruction-oriented features into the latent space. Prior works [13, 16, 17] explore fine-tuning VFMs on reconstruction tasks while introducing a semantic distillation loss to preserve the original VFM outputs. However, such designs impose conflicting objectives, leading to an inherent trade-off between semantic consistency and reconstruction fidelity. Other approaches [18, 14] instead directly augment the latent space with reconstruction-relevant information. Nevertheless, these methods also inject low-level signals that can interfere with the original semantic representations, potentially hindering the convergence of downstream generative models.

For a more controlled and fair comparison, we conduct an empirical study of different VFM-based image tokenizer paradigms under a unified setting in Fig. 1 (Left), where all methods use DiT^{DH}-S as the generative model and are trained on ImageNet at 256×256 resolution for 80 epochs. Specifically, *VFM-finetune* directly unfreezes the VFM encoder during training; *VFM-distill* trains the encoder with an additional distillation loss from a frozen VFM teacher; and *VFM-feat-concat* freezes the VFMs while augmenting reconstruction information through feature-dimensional concatenation. Despite their differences, all variants exhibit a consistent reconstruction–generation trade-off: improved reconstruction fidelity comes at the cost of degraded generative performance.

In this paper, we propose DecQ, a framework designed to resolve this dilemma. DecQ introduces a small set of learnable queries that attend to the intermediate features of a frozen VFM, form-

ing detail-condensing queries that capture low-level reconstruction details complementary to the semantic latent space. Since the VFM remains frozen, these queries enrich fine-grained details without modifying the original VFM parameters or perturbing its semantic representations. DecQ further incorporates these queries into the generative process by jointly denoising them with image patches. We find that predicting the detail-condensing queries also benefits generation, mitigating the reconstruction–generation trade-off. Our main contributions are summarized as follows:

- We propose DecQ, a representation autoencoder framework that uses a small set of learnable queries to capture low-level details under-represented by VFMs via cross-attention. It improves fine-grained reconstruction without altering the pretrained VFM patch-token space.
- We find that condensing features from shallow VFM layers mainly benefits reconstruction, while condensing features from deep VFM layers benefits generation. By condensing information from both shallow and deep VFM layers, these queries effectively improve reconstruction and generation simultaneously.
- Extensive experiments demonstrate that DecQ improves reconstruction over the frozen-VFM baseline while consistently benefiting generative performance, reaching the same target gFID with fewer training epochs and only limited additional overhead.

2 Related work

Representation Alignment in Diffusion Models. Latent Diffusion Models (LDMs) based on Diffusion Transformers (DiTs) have received increasing attention [1, 19, 20]. However, vanilla DiTs often suffer from slow convergence and limited generation performance. To accelerate DiT training, REPA [21] aligns the noisy hidden states of the diffusion model with clean representations from VFMs. Subsequent works further improve this framework from several complementary directions. iREPA [22] refines the alignment mechanism, showing that spatial structure is more crucial than global semantics and enhancing feature transfer via spatial normalization. REPA-E [23] leverages the representation alignment objective to unlock the end-to-end joint tuning of the VAE and DiT without causing latent space collapse. Furthermore, REG [24] addresses the absence of alignment during inference by jointly denoising image latents and a VFM class token, providing continuous semantic guidance for better generation fidelity.

VFM-Aligned Visual Tokenizers for Generation. From another perspective, several works focus on improving the visual tokenizer itself, arguing that its latent space should inherently possess strong semantics [25, 16, 26, 27]. For instance, VA-VAE [25] directly aligns the VAE latent space with pretrained foundation models. Similarly, AlignTok [16] aligns a pretrained VFM to a visual tokenizer rather than forcing the encoder to learn semantics from scratch. Furthermore, DMVAE [26] leverages Distribution Matching Distillation (DMD) to explicitly constrain the encoder's aggregate posterior to match a predefined reference distribution, such as a self-supervised learning prior.

VFMs as Direct Tokenizers for Generation. Recent works, particularly RAE, introduce the idea of directly adopting VFMs as latent encoders for LDMs, enabling generation in the high-dimensional semantic latent space of VFMs with techniques such as noise shift and the DDT head [4, 28–30]. Benefiting from the strong semantics of VFMs, RAE achieves faster convergence and improved generation performance. Concurrently, SVG [14] improves reconstruction in VFM-based latent spaces by concatenating additional reconstruction-oriented information along the feature dimension. Subsequent works [31, 17, 18] have improved upon RAE. For instance, FAE [31] uses a semantic autoencoder to compress the VFM latent space into a lower-dimensional latent space for more efficient generation. Unlike FAE, which completely freezes the encoder, RPiAE [17] proposes a multi-stage training process that initializes from the VFM but allows fine-tuning for reconstruction. To maintain pixel-wise reconstruction quality, LVRAE [18] adds the low-level information under-represented by VFMs back into the output space. However, these methods generally modify or reshape the original VFM semantic space. LVRAE introduces additional low-level information into the output representation, while FAE and RPiAE compress the semantic space into lower dimensions; RPiAE further changes the representation by fine-tuning the VFM itself. In contrast, DecQ preserves the original VFM semantic space. By introducing detail-condensing queries that capture reconstruction-oriented details from VFMs, DecQ simultaneously improves reconstruction fidelity and generation performance with limited extra overhead.

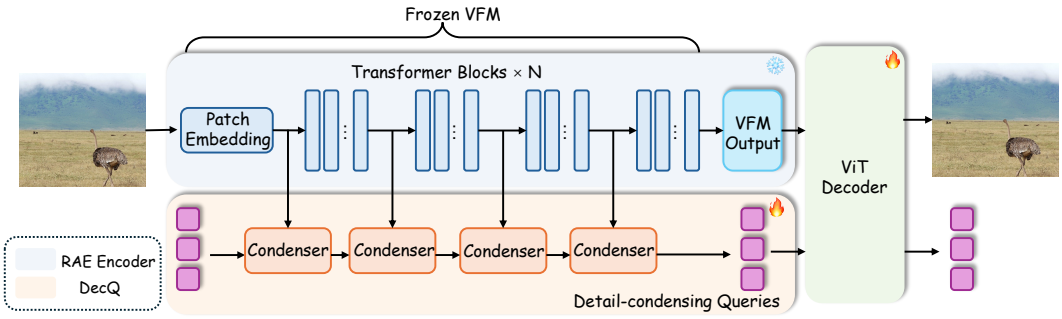


Figure 2: **Overview of the DecQ architecture.** Given an input image, the frozen VFM first converts it into patch tokens and processes them through a stack of Transformer blocks. DecQ attaches learnable queries to multiple intermediate VFM layers and uses condenser modules to progressively aggregate multi-level features into detail-condensing queries. These queries are then fed into the ViT decoder together with the VFM output tokens, providing complementary fine-grained details while keeping the VFM semantic space unchanged.

3 Method

In this section, we first review the preliminaries of representation autoencoders in Sec. 3.1. We then introduce the tokenizer training procedure of our DecQ framework in Sec. 3.2. Finally, we outline the diffusion modeling used for image generation with the trained DecQ tokenizer in Sec. 3.3.

3.1 Preliminary

The standard paradigm for Diffusion Transformers typically relies on a compressed latent space defined by a Variational Autoencoder (VAE). However, the reconstruction-centric objective of VAEs often results in representations that are less semantically structured. This motivates the RAE [4] framework, which redefines latent generative modeling by leveraging frozen, semantically-rich VFMs as the latent space. In RAE, a frozen VFM encoder E extracts high-dimensional latent tokens $z = E(x)$, while a ViT-based decoder D reconstructs images using a combination of pixel-wise (L_1), perceptual (LPIPS), and adversarial (GAN) losses [32–34]. To model this high-dimensional space, RAE adopts a flow matching formulation that interpolates between the latent distribution $p(z)$ and Gaussian noise $\mathcal{N}(0, I)$: $z_t = (1-t)z + t\epsilon$ for $t \in [0, 1]$. A Diffusion Transformer v_θ is then trained to approximate the optimal velocity field $v(z_t, t) = \mathbb{E}[\epsilon - z|z_t]$ by minimizing the mean-squared error objective:

$$\mathcal{L}_{velocity}(\theta) = \int_0^1 \mathbb{E}_{z,\epsilon}\left[\|v_\theta(z_t, t, y) - (\epsilon - z)\|^2\right] dt, \tag{1}$$

where y represents optional class-conditional information.

Despite RAE's effectiveness in capturing high-level semantics, its tokenizer has a key limitation: its latent space consists entirely of patch tokens from a VFM encoder, which are naturally biased toward semantic abstraction. While these tokens encode global semantics well, they under-represent low-level visual details essential for faithful reconstruction, such as color fidelity and fine-grained textures. This motivates a mechanism that supplements fine-grained low-level information while preserving the frozen VFM latent space.

3.2 DecQ Tokenizer

To address this limitation, we introduce *DecQ*, a lightweight tokenizer extension that augments frozen VFM patch tokens with detail-condensing queries. These queries condense complementary low-level information from intermediate layers of the frozen encoder, improving reconstruction quality with minimal additional cost. An overview of DecQ is shown in Fig. 2.

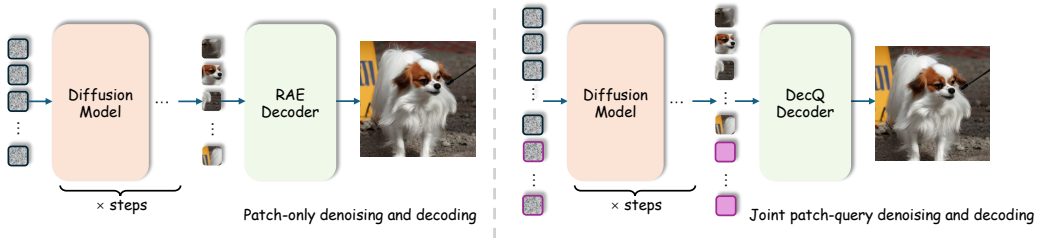


Figure 4: **Image generation with detail-condensing queries.** Compared with the RAE baseline that denoises and decodes only VFM patch tokens, DecQ jointly denoises detail-condensing queries with patch tokens during diffusion. Both patch and query tokens are initialized from Gaussian noise and generated as a unified latent sequence, and are then jointly decoded into the output image.

Encoder with Condensers. We introduce K learnable query tokens $\mathbf{Q}^{(0)} \in \mathbb{R}^{K\times C}$ alongside the frozen VFM backbone, where C is the feature dimension of the patch tokens. In practice, $K \ll N$, so the query tokens provide a compact representation for complementary fine-grained information. To aggregate multi-level features without modifying the pretrained VFM representations, we attach condenser modules to intermediate layers of the frozen encoder.

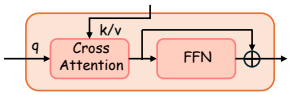


Figure 3: **Architecture of the condenser.**

As shown in Fig. 3, each condenser consists of a cross-attention block followed by an FFN. In the cross-attention block, the query tokens serve as queries, while the intermediate patch tokens serve as keys and values. Let $\mathbf{Q} \in \mathbb{R}^{K\times C}$ and $\mathbf{P} \in \mathbb{R}^{N\times C}$ denote the query and patch tokens, respectively. The cross-attention is defined as:

$$\mathrm{CrossAttn}(\mathbf{Q},\mathbf{P}) = \mathrm{Softmax}\left(\frac{\mathbf{Q}W_Q(\mathbf{P}W_K)^\top}{\sqrt{d}}\right)\mathbf{P}W_V, \tag{2}$$

where W_Q, W_K, W_V are learnable projection matrices, and d denotes the attention head dimension. Let $\mathcal{S} = \{\ell_1, \ldots, \ell_M\}$ denote the ordered VFM layers equipped with condensers. At the i-th condenser, the running query state is updated using the patch tokens from VFM layer ℓ_i through a residual cross-attention block followed by an FFN:

$$\tilde{\mathbf{Q}}^{(i)} = \mathbf{Q}^{(i-1)} + \mathrm{CrossAttn}\big(\mathrm{LN}(\mathbf{Q}^{(i-1)}), \mathrm{LN}(\mathbf{P}^{(\ell_i)})\big), \tag{3}$$

$$\mathbf{Q}^{(i)} = \tilde{\mathbf{Q}}^{(i)} + \mathrm{FFN}\big(\mathrm{LN}(\tilde{\mathbf{Q}}^{(i)})\big). \tag{4}$$

Here, LN denotes layer normalization, and the final query latent is $\mathbf{Z}_{\text{query}} = \mathbf{Q}^{(M)}$. Since patch tokens are only used as keys and values, information flows from patches to queries. This unidirectional design prevents query tokens from altering the pretrained VFM representations, thereby preserving the encoder-defined VFM patch-token representation. The encoder outputs semantic patch tokens $\mathbf{Z}_{\text{patch}}$ and detail-condensing query tokens $\mathbf{Z}_{\text{query}}$.

Dual-Stream Decoder. We follow the ViT decoder recipe of RAE and incorporate both patch and query tokens. Unlike the encoder, where unidirectional cross-attention prevents modification of patch tokens, the decoder aims to fuse patch semantics and query details. We therefore project the two streams separately, concatenate them, and use joint self-attention for bidirectional fusion. Patch tokens lie on a regular 2D grid and use fixed 2D sinusoidal positional embeddings, whereas query tokens are latent slots without predefined spatial positions and use independent learnable embeddings:

$$\mathbf{H}^{(0)} = [\mathbf{Z}_{\text{patch}} + \mathbf{PE}_{\text{2D}} \parallel \mathbf{Z}_{\text{query}} + \mathbf{PE}_Q], \tag{5}$$

where $[\cdot\|\cdot]$ denotes concatenation. The combined sequence is processed jointly by the decoder. Only patch tokens are used for pixel prediction, while query tokens participate in decoder self-attention to provide fine-grained details. Finally, following the regularization strategy of RAE, we apply noise augmentation to both patch and query latents during training.

3.3 Generation with Detail-Condensing Queries

As shown in Fig. 4, DecQ preserves the encoder-defined VFM patch-token representation while expanding the state modeled by the diffusion model with a coupled auxiliary query stream: $\mathbf{Z} = [\mathbf{Z}_{\text{patch}} \parallel \mathbf{Z}_{\text{query}}]$. During generative modeling, patch and query tokens interact through self-attention, are jointly denoised, and are then fed into the decoder to produce the output image.

Preserved Patch Representation and Expanded Generative State. During encoding, $\mathbf{Z}_{\text{patch}} = E(x)$ remains identical to the patch-token output of RAE, since the VFM is frozen and the condensers only access its intermediate features without modifying the patch tokens. During generation, the diffusion model operates on the expanded joint state $[\mathbf{Z}_{\text{patch}} \| \mathbf{Z}_{\text{query}}]$, where the patch and query streams interact through self-attention throughout denoising. This design retains the original encoder-defined semantic patch representation while introducing an auxiliary query stream that captures complementary visual details and participates in generative modeling. Consequently, the generated patch tokens remain compatible with the original RAE decoding interface, while the query stream can both guide patch denoising and provide fine-grained details to the DecQ decoder, as shown in Tab. 4.

Sequence Modeling. We model the extended latent sequence $\mathbf{Z}$ using the DiT^{DH} architecture adopted in RAE [4, 29], trained under the flow matching objective. Patch and query tokens are jointly denoised with global self-attention and are then jointly fed into the decoder to produce the final image. To account for their different token types, we use separate input projections and positional encodings: patch tokens are equipped with 2D positional embeddings, while query tokens use independent learnable positional embeddings.

Optimization and Inference. During training, the flow matching velocity prediction loss is computed over the full sequence and decomposed as

$$\mathcal{L} = \mathcal{L}_{\text{patch}} + \lambda_{\text{query}} \cdot \mathcal{L}_{\text{query}}, \tag{6}$$

where $\mathcal{L}_{\text{patch}}$ and $\mathcal{L}_{\text{query}}$ denote the mean squared error (MSE) over patch and query tokens, respectively, and λ_{query} controls the weight of query-token prediction. At inference time, we sample Gaussian noise for the full latent sequence and integrate the flow ODE to obtain both patch and query latents, which are then decoded into the final image.

4 Experiments

4.1 Experimental Settings

We follow the RAE experimental protocol and keep key generation settings, including the dimension-dependent time shift and wide DDT head [30, 29], consistent with the original RAE configuration. Unless otherwise specified, we use DINOv2-B as the default VFM and a ViT-XL decoder with approximately 500M parameters, and conduct experiments on ImageNet [35] at 256×256 resolution. By default, DecQ uses 8 detail-condensing queries, with condensers attached to VFM layers 0, 3, 6, and 9. During diffusion training, query and patch tokens share the same noise schedule, and the query-token loss weight is set to 1. Additional details are provided in Appendix A.

For reconstruction, we report PSNR and SSIM [36] for pixel-level fidelity, and Fréchet Inception Distance (FID) [37], denoted as rFID, for distributional quality and visual realism. For generation, we report FID, Inception Score (IS), Precision (Prec.), and Recall (Rec.), with generation FID denoted as gFID. Metrics are computed using the ADM evaluation suite on 50,000 class-uniform generated samples [38]. Unless otherwise specified, we use 50 sampling steps following the RAE protocol.

Table 2: **Class-conditional generation performance on ImageNet 256×256.** DecQ achieves superior generation quality compared to various types of tokenizers, both with and without guidance. These results suggest that DecQ provides strong generative modeling capability.

Method	Epochs	#Params	Generation w/o guidance				Generation w/ guidance			
			gFID↓	IS↑	Pre.↑	Rec.↑	gFID↓	IS↑	Pre.↑	Rec.↑
Traditional VAE Tokenizer										
SiT [20]	1400	675M	8.61	131.7	0.68	**0.67**	2.06	270.3	**0.82**	0.59
REPA [21]	800	675M	5.90	–	–	–	1.42	**305.7**	0.80	0.65
MaskDiT [39]	1600	675M	5.69	177.9	0.74	0.60	2.28	276.6	0.80	0.61
REG [24]	480	677M	2.2	–	–	–	1.40	296.9	0.77	0.66
VFM-Aligned VAE Tokenizer										
VAVAE [25]	64	675M	5.14	130.2	0.76	0.62	2.11	252.3	0.81	0.58
	800	675M	2.17	205.6	0.77	0.65	1.35	295.3	0.79	0.65
DMVAE [26]	64	675M	3.22	171.7	–	–	–	–	–	–
	800	675M	1.64	216.3	–	–	–	–	–	–
Low-dim VFM Tokenizer										
FAE [31]	80	675M	2.08	207.6	**0.82**	0.59	1.70	243.8	**0.82**	0.61
	800	675M	1.48	239.8	0.81	0.63	1.29	268.0	0.80	0.64
RPiAE [17]	60	675M	2.46	201.1	0.80	0.59	2.06	208.5	0.80	0.61
	80	675M	2.25	208.7	0.81	0.60	1.51	225.9	0.79	0.65
High-dim VFM Tokenizer										
RAE [4]	80	839M	2.16	214.8	**0.82**	0.59	1.74	233.2	**0.82**	0.61
	800	839M	1.51	242.9	0.79	0.63	1.13	262.6	0.78	**0.67**
LV-RAE [18]	800	839M	2.42	223.8	0.77	0.64	1.82	249.7	0.75	**0.67**
DecQ	80	841M	1.80	223.9	0.81	0.61	1.33	234.3	0.80	0.64
	800	841M	**1.41**	**251.9**	0.81	0.63	**1.05**	259.6	0.79	0.66

4.2 Main Results

4.2.1 Reconstruction Ability

We report reconstruction results in Tab. 1. Under the same frozen-VFM setting, DecQ substantially improves both distributional and pixel-level reconstruction over RAE at a resolution of 256×256. Compared with reconstruction-oriented VAEs, its contribution is a more favorable reconstruction–generation balance while retaining the encoder-defined VFM patch representation. Notably, DecQ does not introduce any additional encoder to extract information directly from the input image. Instead, it leverages intermediate features within the frozen VFM to recover fine-grained information that is progressively lost along the forward pass. This design is both lightweight and structurally consistent with the original representation space. Additional qualitative results are provided in Appendix B.

Table 1: **Quantitative comparison of reconstruction performance across different VFM-based tokenizers**. Our proposed DecQ achieves the lowest rFID within VFM-based encoders and significantly outperforms RAE in pixel-wise reconstruction metrics.

Method	rFID↓	SSIM↑	PSNR↑
SD-VAE [1]	0.61	0.74	26.90
VA-VAE [25]	0.28	0.79	27.96
SVG [14]	0.65	**0.65**	**23.89**
RAE [4]	0.69	0.49	19.13
RPiAE [17]	0.50	0.53	21.30
FAE [31]	0.68	–	–
DecQ	**0.47**	0.63	22.76

4.2.2 Generation Ability

We report the main generation results in Tab. 2. Existing LDM-based methods can be broadly categorized into four groups: (1) traditional approaches based on standard VAEs, (2) methods that enhance VAEs with semantic alignment, (3) methods that employ VFMs as tokenizers and perform generation in a low-dimensional latent space, and (4) methods that directly generate in the high-dimensional VFM feature space. For high-dimensional generation, DecQ follows RAE and

Table 3: **Performance comparison of different VFM-based tokenizer training frameworks.**

Method	rFID↓	SSIM↑	PSNR↑	w/o guidance 80 epoch			
				FID↓	IS↑	Pre.↑	Rec.↑
VFM-freeze (RAE)	0.67	0.51	19.61	6.05	146.6	**0.80**	0.53
VFM-finetune	**0.15**	**0.94**	**33.83**	57.81	26.32	0.40	0.56
VFM-distill	0.23	0.81	27.09	20.20	81.02	0.62	0.57
VFM-feat-concat	0.49	0.61	22.60	8.75	131.1	0.75	0.52
DecQ	0.47	0.63	22.76	**4.74**	**148.9**	0.78	**0.59**

adopts the same generative architecture and training settings. Additional implementation details are provided in Appendix A. Experimental results show that DecQ achieves an FID of 1.80 at 80 epochs and 1.41 at 800 epochs without guidance, and further improves to 1.05 at 800 epochs with guidance, outperforming previous state-of-the-art methods. Additional sampling details and qualitative generation results are provided in Appendix B.

4.3 Ablation Study

In Tab. 3, we compare different backbone training paradigms, including freezing the VFM (RAE), unfreezing with and without distillation, feature concatenation, and our proposed DecQ. For distillation, we use an L_2 loss to align the encoder outputs with those of a frozen copy of the VFM. For the feature concatenation baseline, we set the number of query tokens equal to the number of patch tokens, train a low-dimensional bottleneck during reconstruction, and concatenate the resulting query features with patch tokens to form a new latent space.

Overall, the results reveal a clear reconstruction–generation trade-off. Freezing the VFM preserves strong generative performance but limits reconstruction, while full fine-tuning substantially improves reconstruction at the cost of degraded generation. Adding distillation slightly alleviates this issue but does not resolve the trade-off. Feature concatenation improves reconstruction but still underperforms in generation, suggesting that directly concatenating query features with patch tokens does not necessarily yield a well-aligned latent space for generative modeling. In contrast, DecQ improves both reconstruction and generation by preserving the original semantic structure while augmenting it with complementary fine-grained information, effectively mitigating the trade-off.

In Fig. 5, we compare the training efficiency of DecQ with REPA [21] and RAE [4] using FID-50K on ImageNet at 256×256 resolution. DecQ achieves a gFID of 1.80 after 80 epochs and reaches 1.51 after 240 epochs, matching RAE trained for 800 epochs. At the same 80-epoch budget, DecQ also improves gFID from 2.16 for RAE to 1.80, showing that the benefit emerges early and persists throughout training. Thus, DecQ requires $3.3\times$ fewer training epochs to reach the same gFID under the evaluated protocol. Together, these results show that the query-augmented latent sequence reaches a strong operating point with substantially fewer optimization steps.

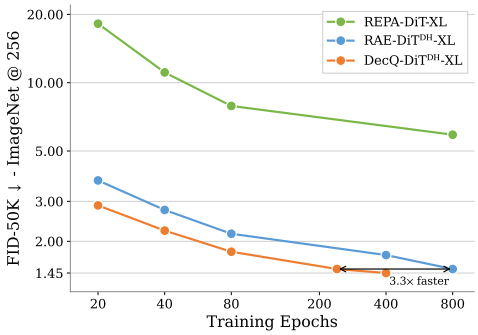


Figure 5: **Convergence of the proposed DecQ compared with REPA [21] and RAE [4].**

Table 5: **Ablation on the number of queries.** More queries improve reconstruction but do not always benefit generation. Using 8 queries provides the best reconstruction-generation trade-off.

Number of queries	rFID↓	SSIM↑	PSNR↑	w/o guidance 80 epoch		Additional overhead	
				FID↓	IS↑	Flops	Params
RAE	0.67	0.51	19.61	6.05	146.6	–	–
$K = 2$	0.57	0.54	20.89	5.94	146.9	+1.7%	+29.3 M
$K = 4$	0.54	0.56	21.33	5.22	**151.1**	+2.4%	+29.3 M
$K = 8$	0.47	0.63	22.76	**4.74**	148.9	+3.9%	+29.3 M
$K = 16$	**0.32**	**0.71**	**24.41**	6.43	138.7	+6.8%	+29.3 M

Table 6: **Ablation on condenser placement across VFM layers.** Shallow layers favor reconstruction, while deeper layers benefit generation. The sparse configuration at layers 0, 3, 6, and 9 achieves a balanced trade-off between fidelity, quality, and efficiency.

Layers	rFID↓	SSIM↑	PSNR↑	w/o guidance 80 epoch		Additional overhead	
				FID↓	IS↑	Flops	Params
RAE	0.67	0.51	19.61	6.05	146.6	–	–
Layer 0,1,2,3	**0.42**	**0.66**	**23.47**	6.35	143.5	+3.9%	+29.3 M
Layer 8,9,10,11	0.54	0.54	20.72	5.09	**156.1**	+3.9%	+29.3 M
All layers	0.45	0.62	22.67	4.77	150.5	+6.1%	+86.1 M
Layer 0,3,6,9	0.47	0.63	22.76	**4.74**	148.9	+3.9%	+29.3 M

In Tab. 4, we compare RAE, DecQ, and DecQ (RAE decoder). DecQ (RAE decoder) uses the DecQ tokenizer for diffusion training, but discards the generated query tokens at inference and decodes only the generated patch tokens with the RAE decoder. Since DecQ preserves the original VFM patch-token latent space, replacing the DecQ decoder with the RAE decoder does not introduce a latent-space mismatch for the patch tokens, allowing direct comparison of decoding effects.

Table 4: **Ablation of the roles and persistence of query tokens.**

Method	w/o guidance 80 epoch	
	FID↓	IS↑
RAE	2.16	214.8
DecQ (RAE decoder)	1.99	**229.8**
DecQ	**1.80**	223.9
DecQ (query-state reset)	3.44	–

Interestingly, DecQ (RAE decoder) still outperforms RAE even when the generated query tokens are discarded at inference. This suggests that predicting detail-condensing queries may itself help the diffusion model generate better patch tokens, in a way reminiscent of REG [24]. Moreover, the full DecQ model further improves over DecQ (RAE decoder), showing that the generated query tokens carry fine-grained information that directly benefits decoding and final generation quality. In the query-state reset setting, we reset query states to zero before every Euler-step model evaluation, discard them after sampling, and decode only the generated patch tokens with the RAE decoder. Resetting the query trajectory degrades gFID to 3.44, indicating that persistent query states provide useful cross-step guidance for patch denoising rather than merely supplying additional information to the final decoder.

We study the effect of varying the number of queries in Tab. 5. Increasing the number of queries consistently improves reconstruction, as additional queries can condense richer low-level information from intermediate VFM features. However, better reconstruction does not always translate to better generation. With DiT^{DH}-S, FID first decreases and then increases as more queries are used, suggesting that a moderate number of queries provides useful complementary details, whereas excessive queries may introduce redundant low-level information that interferes with generative modeling. In practice, using 8 queries achieves the best generative performance, striking a favorable balance between reconstruction fidelity and generation quality. Detailed overhead analysis is provided in Appendix D.

We study the effect of applying condensers at different VFM layers in Tab. 6. Different depths show distinct behaviors: shallow layers provide richer low-level details and improve reconstruction,

Table 7: **Performance comparison of RAE and DecQ based on SigLIP2.** Results show that DecQ remains effective across different VFMs, highlighting its robustness and general applicability.

Method	rFID↓	SSIM↑	PSNR↑	w/o guidance 80 epoch		Additional overhead	
				FID↓	IS↑	Flops	Params
RAE	0.73	0.51	19.92	11.10	107.7	–	–
DecQ	**0.57**	**0.60**	**22.07**	**10.09**	**111.5**	+3.9%	+29.3M

but degrade generative performance; deeper layers offer better generation by capturing higher-level semantics. This reveals a layer-dependent reconstruction–generation trade-off. While applying condensers to all layers achieves strong performance, it also introduces higher computational and parameter overhead. We therefore adopt a sparse design at layers 0, 3, 6, and 9, which performs comparably to dense aggregation while better balancing reconstruction fidelity, generation quality, and computational cost.

We analyze the roles of query and patch tokens through a clustering-based study. Using the top-left image as the anchor, we retrieve its nearest neighbors under query-token and patch-token representations, as shown in Fig. 6. Query-token clusters tend to share color-related visual patterns with the anchor, suggesting that query tokens primarily capture low-level appearance details such as color and texture. In contrast, patch-token clusters consistently retrieve images with similar semantic content and object-level structures, indicating stronger high-level category information. This qualitative comparison highlights the complementary roles of the two representations: query tokens enrich fine-grained visual details, while patch tokens preserve semantic structure. More results are provided in Appendix B.



Figure 6: **Images clustered using different token representations.**

Generalization across different VFMs. To evaluate generality, we conduct analogous experiments with SigLIP2-B, as reported in Tab. 7. Consistent with DINOv2, reconstruction with a frozen SigLIP2 is limited, while introducing detail-condensing queries substantially improves pixel-level metrics. Although SigLIP2 shows lower reconstruction and generative performance than DINOv2, DecQ still brings consistent gains under the same setting. Overall, these results show that DecQ remains effective across different VFM architectures, highlighting its robustness and general applicability.

5 Conclusion

We presented DecQ, a framework that introduces detail-condensing queries to attend to intermediate VFM layers via cross-attention, recovering fine-grained information progressively lost in VFM representations. DecQ enriches low-level details while preserving the encoder-defined VFM patch-token representation. During generation, it jointly denoises patch and query tokens, enabling richer details and higher generation quality. Experiments consistently show that DecQ improves both reconstruction and generation with minimal computational overhead.

Acknowledgments and Disclosure of Funding

Funding. This work received computational support from JD.COM. No other external funding or third-party support was received.

Competing interests. The authors declare no competing interests.

References

[1] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1, 3, 7

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

[3] 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. 1

[4] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *ICLR*, 2026. 1, 3, 4, 6, 7, 8, 17

[5] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *TMLR*, 2024. 1, 2

[6] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 1, 2

[7] 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 *ICML*, 2021. 1

[8] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023. 1

[9] 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. 2

[10] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 2

[11] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *NeurIPS*, 2017.

[12] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021. 2

[13] 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. 2

[14] 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 *ICLR*, 2026. 2, 3, 7

[15] Wei Song, Yuran Wang, Zijia Song, Yadong Li, Zenan Zhou, Long Chen, Jianhua Xu, Jiaqi Wang, and Kaicheng Yu. Dualtoken: Towards unifying visual understanding and generation with dual visual vocabularies. In *ICLR*, 2026. 2

[16] 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. In *ICLR*, 2026. 2, 3

[17] Yue Gong, Hongyu Li, Shanyuan Liu, Bo Cheng, Yuhang Ma, Liebucha Wu, Xiaoyu Wu, Manyuan Zhang, Dawei Leng, Yuhui Yin, et al. Rpiae: A representation-pivoted autoencoder enhancing both image generation and editing. *arXiv preprint arXiv:2603.19206*, 2026. 2, 3, 7

[18] Siyu Liu, Chujie Qin, Hubery Yin, Qixin Yan, Zheng-Peng Duan, Chen Li, Jing Lyu, Chun-Le Guo, and Chongyi Li. Improving reconstruction of representation autoencoder. *arXiv preprint arXiv:2602.08620*, 2026. 2, 3, 7

[19] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 3

[20] 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 *ECCV*, 2024. 3, 7

[21] 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. In *ICLR*, 2025. 3, 7, 8

[22] Jaskirat Singh, Xingjian Leng, Zongze Wu, Liang Zheng, Richard Zhang, Eli Shechtman, and Saining Xie. What matters for representation alignment: Global information or spatial structure? In *ICLR*, 2026. 3

[23] Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning with latent diffusion transformers. In *ICCV*, 2025. 3

[24] Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, jian Yang, Ming-Ming Cheng, and Xiang Li. Representation entanglement for generation: Training diffusion transformers is much easier than you think. In *NeurIPS*, 2026. 3, 7, 9

[25] Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *CVPR*, 2025. 3, 7, 14

[26] Sen Ye, Jianning Pei, Mengde Xu, Shuyang Gu, Chunyu Wang, Liwei Wang, and Han Hu. Distribution matching variational autoencoder. *arXiv preprint arXiv:2512.07778*, 2025. 3, 7

[27] Qifan Li, Xingyu Zhou, Jinhua Zhang, Weiyi You, and Shuhang Gu. Taming sampling perturbations with variance expansion loss for latent diffusion models. *arXiv preprint arXiv:2603.21085*, 2026. 3

[28] Tianci Bi, Xiaoyi Zhang, Yan Lu, and Nanning Zheng. Vision foundation models can be good tokenizers for latent diffusion models. *arXiv preprint arXiv:2510.18457*, 2025. 3

[29] Kaihang Pan, Wang Lin, Zhongqi Yue, Tenglong Ao, Liyu Jia, Wei Zhao, Juncheng Li, Siliang Tang, and Hanwang Zhang. Generative multimodal pretraining with discrete diffusion timestep tokens. In *CVPR*, 2025. 6

[30] 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. 3, 6

[31] Yuan Gao, Chen Chen, Tianrong Chen, and Jiatao Gu. One layer is enough: Adapting pretrained visual encoders for image generation. *arXiv preprint arXiv:2512.07829*, 2025. 3, 7

[32] 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. 4

[33] Axel Sauer, Tero Karras, Samuli Laine, Andreas Geiger, and Timo Aila. Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis. In *ICML*, 2023.

[34] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 4

[35] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6, 21, 22

[36] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004. 6

[37] 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. 6

[38] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021. 6

[39] Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *TMLR*, 2023. 7

[40] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *ICLR*, 2024. 14

[41] Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. In *NeurIPS*, 2025. 14

A Implementation Details

A.1 DecQ Implementation

We follow the training scheme of RAE. For the encoder, we use DINOv2 with Registers [40] to process images resized to 224×224, producing 256 patch tokens that are then used to reconstruct images at 256×256 resolution. Both the [CLS] and [REG] tokens are discarded after encoding. For both patch tokens and the newly introduced query tokens, we apply layer normalization with `elementwise_affine=False` to ensure proper normalization. We adopt the same noise injection strategy as RAE for patch tokens, and apply noise with the same variance to the query tokens.

We follow the decoder design of RAE. The query tokens are projected independently, assigned learnable positional embeddings, and concatenated with patch tokens to form a unified input sequence. The combined sequence is processed jointly by the Transformer decoder. Only patch tokens are used for final reconstruction, while query tokens serve as auxiliary latent variables that enhance the decoding process. For experiments with SigLIP2, we use the same DecQ configuration as DINOv2 unless otherwise specified.

A.2 Diffusion Model Implementation

Following RAE, we use LightningDiT [25] as the backbone of our diffusion model. We adopt a continuous-time flow matching formulation, where the timestep is defined over the real interval $[0, 1]$, and replace the standard timestep embedding with Gaussian Fourier feature embeddings.

For DiT^{DH}, we largely follow RAE, using DiT^{DH}-XL for the main results and DiT^{DH}-S for ablations. When the hidden dimension of the DiT backbone differs from that of the DDT head, we use a linear projection layer to map the encoder output to the decoder dimension.

For optimization, we mainly follow the LightningDiT training recipe. We use AdamW with a constant learning rate of 2.0×10^{-4}, a batch size of 1024, and an EMA decay of 0.9999. We also apply gradient clipping with a threshold of 1.0. The query-token loss weight λ_{query} is set to 1, since query and patch tokens are constrained to have the same variance during tokenizer training. All diffusion models are trained on 8 NVIDIA H200 GPUs.

A.3 Sampling Details

We use standard ODE sampling with an Euler solver and default to 50 sampling steps. We also observe that increasing the number of steps to 250 can yield further improvements. For FID-50K evaluation, we follow the RAE protocol and sample 50 images per class, resulting in 50,000 images in total.

Following RAE, we adopt AutoGuidance [41] as our primary guidance strategy, which uses a weaker diffusion model to guide a stronger one. Consistent with RAE, we use the minimal variant, DiT^{DH}-S, as the guiding model, initialized from a relatively early checkpoint. Our best results are obtained using the 60-epoch checkpoint of DiT^{DH}-S with a guidance scale of 1.6.

B More Qualitative Results

We present additional qualitative results, including clustering analyses that illustrate the role of query tokens, reconstruction comparisons on DINOv2 and SigLIP2, and qualitative generated samples.

B.1 Cluster Analysis

Additional clustering results are shown in Fig. 7. They further show that query tokens primarily capture fine-grained visual details such as color and texture, while patch tokens preserve high-level semantics such as object identity.

B.2 Reconstruction Performance

Additional reconstruction comparisons with the DINOv2-based RAE are shown in Fig. 8, using the same setting as in Fig. 1 Right.

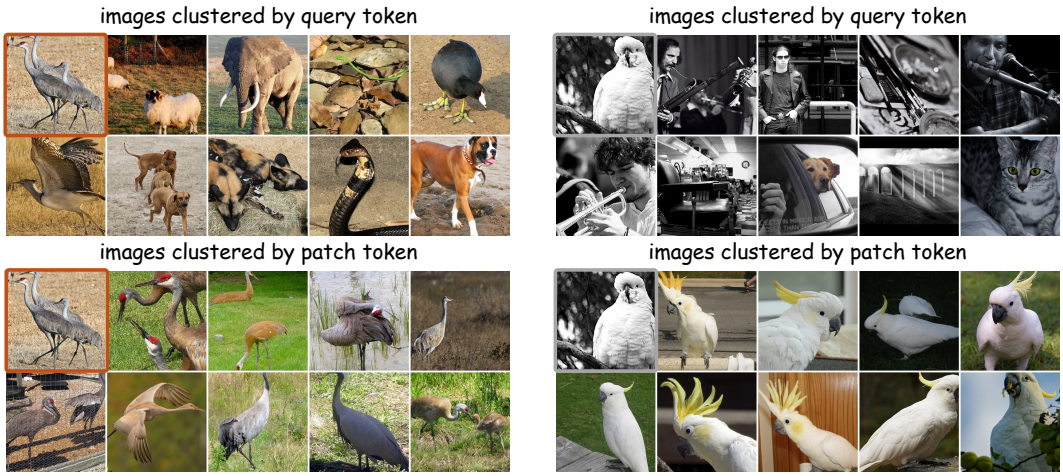


Figure 7: **More results on cluster visualization.** Similar to Fig. 6, images clustered by query tokens shares similar appearances like background color, while images clustered by patch tokens share high-level semantics such as main subjects. This suggests that query tokens mainly capture fine-grained visual details, while patch tokens preserve high-level semantics.

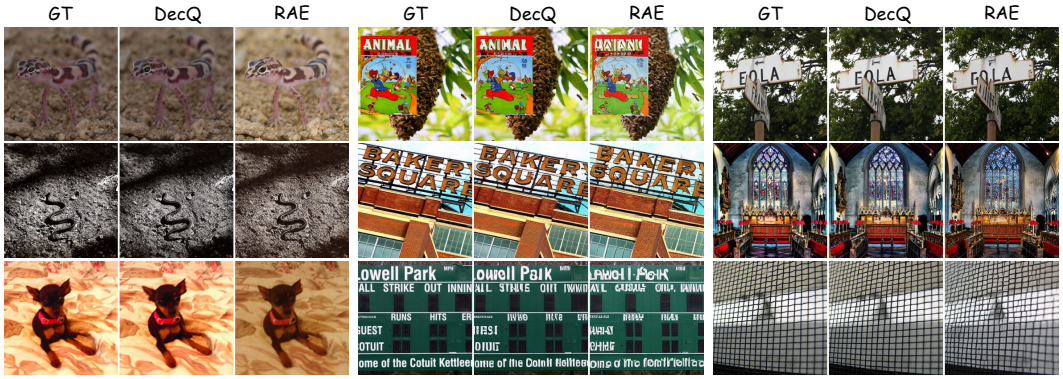


Figure 8: **More qualitative results of our image reconstruction compared with RAE based on DINOv2.** The cases share the same setting as in Fig. 1 Right. Compared with RAE, DecQ better preserves background colors, textual content, and fine-grained textures.

Reconstruction results compared with RAE based on SigLIP2 are presented in Fig. 9. In some cases, the SigLIP2-based RAE retains a semantic impression of textual content but fails to faithfully reproduce its colors, whereas DecQ better preserves these fine-grained details.

Failure cases. Although DecQ better preserves colors, textures, and local appearance than RAE, highly localized structures remain difficult to reconstruct exactly. As illustrated by the textual and repetitive-pattern examples in Figs. 8 and 9, the overall appearance of a textual region may be improved while individual characters remain incorrect, and fine repetitive patterns may still be blurred or locally misaligned. Small objects and thin structures can also be simplified or omitted. These failures may arise when fine-grained information is weakly retained in intermediate VFM features or cannot be fully represented and recovered through the current patch–query bottleneck.

B.3 Generation Performance

Our class-to-image generation results are presented in Fig. 10, demonstrating the strong generative capabilities of DecQ.



Figure 9: **Qualitative results of our image reconstruction compared with RAE based on SigLIP2.** In some cases, SigLIP2 appears to retain a semantic impression of textual content but fails to accurately reproduce its colors. In contrast, DecQ better preserves these fine-grained details.



Figure 10: **Qualitative results of our image generation.**

C Additional Ablation Studies

C.1 Decoder Design

The term "query" refers to the information-extraction role of these tokens in the encoder condensers. After condensation, the tokens serve as compact auxiliary detail latents. We compare the default joint self-attention decoder with an encoder-like cross-attention decoder in Tab. 8. The cross-attention variant is more restrictive because it permits only one-way interaction, whereas joint self-attention enables richer bidirectional fusion of patch semantics and query details throughout decoding.

C.2 Positional Embeddings

Patch tokens correspond to a regular 2D image grid, whereas query tokens are compact latent slots without predefined spatial positions. We therefore use fixed 2D positional embeddings for patches and independent learnable embeddings for queries. As shown in Tab. 9, learnable 2D patch embeddings remain competitive, while fixed 1D query embeddings substantially degrade query-token generation.

C.3 Query-Loss Weight

The query-loss weight λ_{query} controls the relative supervision strength assigned to patch-token and query-token denoising. We sweep this hyperparameter under the DiT^{DH}-XL setting in Tab. 10. At 20 epochs, $\lambda_{\text{query}} = 0.5$ performs best, while the default value of 1 performs best at 80 epochs.

Table 8: Ablation on the decoder design.

Decoder design	PSNR↑	SSIM↑	rFID↓
Cross-attention	21.65	0.58	0.51
Joint self-attention	**22.76**	**0.63**	**0.47**

Table 9: Ablation on positional embeddings for patch and query tokens.

Patch positional embedding	Query positional embedding	gFID↓	IS↑
Fixed 2D	Learnable 1D	**4.74**	148.90
Learnable 2D	Learnable 1D	4.97	**151.55**
Fixed 2D	Fixed 1D	6.71	132.24

One interpretation is that weaker query supervision allows the model to prioritize semantic patch-token prediction early in training, whereas more balanced supervision later helps it exploit the complementary information and interaction between the two streams. Overall, performance remains relatively stable across the tested range.

Table 10: Ablation on the query-token loss weight λ_{query}.

λ_{query}	gFID at 20 epochs↓	gFID at 80 epochs↓
0.25	2.99	1.99
0.50	**2.76**	1.86
1.00	2.87	**1.80**
2.00	3.20	1.89

D Computational Overhead Analysis

We analyze the additional computational and parameter overhead introduced by DecQ. All GFLOPs are reported under the MACs convention, following RAE [4].

D.1 Tokenizer and Reconstruction Overhead

Baseline. The frozen ViT-B/14 encoder processes $N{=}256$ tokens through 12 Transformer layers ($d{=}768$, $d_{\text{ff}}{=}3072$), resulting in 22.2 GFLOPs. The ViT-MAE XL decoder processes $N_{\text{dec}}{=}257$ tokens through 28 layers ($d_{\text{dec}}{=}1152$, $d_{\text{ff}}{=}4096$), costing 106.7 GFLOPs. This gives a total baseline cost of 128.9 GFLOPs per image, with 501.9M active parameters.

Computational overhead. DecQ introduces two sources of additional computation. First, DecQ inserts $M{=}4$ condensers, where M denotes the number of VFM layers equipped with a condenser, to extract $K{=}8$ query tokens, adding 1.44 GFLOPs. Second, concatenating the query tokens increases the decoder sequence length from 257 to 265, adding 3.58 GFLOPs. In total, the default query mechanism adds 5.0 GFLOPs, corresponding to only +3.9% over the baseline.

Parameter overhead. Each condenser contains a cross-attention extractor and a feed-forward network, contributing 7.09M parameters per condenser. With $M{=}4$ condensers, this amounts to 28.36M parameters. The remaining components, including learnable queries, the decoder query projection, and query positional embeddings, contribute only a minor overhead. Overall, DecQ introduces 29.3M additional trainable parameters, corresponding to +5.8% of the 501.9M-parameter baseline. The parameter overhead is governed mainly by the condenser dimension and the number of inserted condensers, and is nearly independent of the number of query tokens K.

Summary. Table 11 summarizes the overhead under different configurations. The default setting ($M{=}4$, $K{=}8$) adds only +3.9% computation and +5.8% parameters, showing that DecQ improves the tokenizer with modest additional cost.

Table 11: Tokenizer and reconstruction overhead of DecQ. M denotes the number of condenser-equipped VFM layers, and K denotes the number of query tokens. GFLOPs follow the MACs convention. "Active Params" counts parameters that participate in forward computation.

Configuration	GFLOPs			Active Params	Overhead
$(M \times K)$	Encoder	Decoder	Total	(Δ)	(FLOPs / Params)
Baseline (no query)	22.2	106.7	128.9	—	—
4×2	23.5	107.6	131.1	+29.3 M	+1.7% / +5.8%
4×4	23.5	108.5	132.0	+29.3 M	+2.4% / +5.8%
4×8	23.6	110.3	133.9	+29.3 M	+3.9% / +5.8%
4×16	23.8	113.9	137.7	+29.3 M	+6.8% / +5.8%
4×32	24.2	121.1	145.3	+29.3 M	+12.7% / +5.8%
12×8	26.5	110.3	136.8	+86.1 M	+6.1% / +17.1%

Table 12: Generation overhead of DecQ with DiT$^{\mathrm{DH}}$-XL. GFLOPs are computed for complete generation, including 50 ODE sampling steps and one RAE decoding step. The baseline cost is 8,189.8 GFLOPs.

Query tokens K	DiT Δ/step	Total GFLOPs Δ	Param. Δ
2	+1.30	+65.8 (+0.80%)	+3.35 M
4	+2.60	+131.6 (+1.61%)	+3.36 M
8	+5.20	+263.4 (+3.22%)	+3.37 M
16	+10.40	+527.2 (+6.44%)	+3.39 M
32	+20.84	+1056.3 (+12.90%)	+3.42 M

D.2 Generation Overhead

We further analyze the overhead introduced by query tokens during diffusion generation with DiT$^{\mathrm{DH}}$-XL. The baseline generation pipeline consists of 50 ODE sampling steps followed by one RAE decoding step, resulting in 8,189.8 GFLOPs in total. For the default setting K=8, appending query tokens increases the DiT cost by 5.20 GFLOPs per sampling step. Together with the additional 3.58 GFLOPs in the final RAE decoder, the complete generation overhead is 263.4 GFLOPs, corresponding to +3.22% over the baseline.

The parameter overhead during generation is even smaller. Unlike the tokenizer stage, which requires cross-attention condensers, the generation stage only introduces lightweight embedding and projection layers for the additional query tokens. For K=8, this adds 3.37M parameters in total. Since most of these parameters come from embedding projections rather than query-specific positional embeddings, the parameter overhead is nearly independent of K.

Table 12 summarizes the generation overhead for different numbers of query tokens. The additional computation scales approximately linearly with K, while the quadratic attention term remains negligible in this regime. Overall, DecQ introduces only a modest overhead during generation, with the default setting adding +3.22% computation and 3.37M parameters.

E Limitations

While DecQ consistently improves both reconstruction and generation under our experimental settings, several limitations remain. First, our experiments are mainly conducted on ImageNet at 256×256 resolution. Although ImageNet is a standard benchmark for class-conditional image generation, evaluating DecQ on more diverse datasets, such as text-to-image datasets or domain-specific image collections, would provide a more comprehensive understanding of its generality. Second, we have not extensively studied higher-resolution generation, such as 512×512 or above. Since fine-grained details become increasingly important at higher resolutions, it would be valuable to examine how the number of queries, condenser placement, and computational overhead scale in such settings.

In addition, our method is evaluated primarily with DINOv2 and SigLIP2 as representative VFMs. While the results suggest that DecQ generalizes across different VFM architectures, a broader study covering more backbone families and model scales remains an important direction. DecQ narrows the reconstruction gap under the frozen-VFM setting, while dedicated reconstruction-oriented VAEs still retain higher pixel-level fidelity; evaluating DecQ on dedicated high-fidelity editing benchmarks is left for future work.

Finally, the number of queries K controls an operating point between reconstruction and generation rather than improving both objectives monotonically. DecQ also introduces additional query tokens and condenser modules. Although the default configuration incurs only modest overhead, the cost may increase when more queries or denser condenser placements are used. Future work may explore adaptive query allocation or more efficient condenser designs to further reduce overhead while preserving the reconstruction and generation benefits.

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: Yes. The main claims and contributions are stated in Sec. 1 and supported by experimental evidence in Sec. 4.

   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: Please see Sec. E for limitations.

   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: This 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: We describe our model architecture and experimental details in Sec. 4.1 and Sec. A. We use publicly accessible dataset ImageNet [35], and we will release our code and models to reproduce the results.

   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: [Yes]

Justification: We use publicly accessible dataset ImageNet [35], and we will release our code and models to reproduce the results.

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: See Sec. 4.1 and Sec. A.

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: [No]

Justification: Due to the large computational cost of training diffusion models, we do not report error bars or repeated-run statistics. We evaluate multiple models, backbones, and ablation settings in Sec. 4, making repeated runs for every setting prohibitively expensive.

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: See Sec. A.

   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: We followed the NeurIPS Code of Ethics.

   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: [No]

    Justification: We do not include a dedicated broader impacts discussion. The work is evaluated on public academic benchmarks and does not use private or personal data. As with other image generation methods, improved generation quality could potentially be misused for synthetic content creation, but this paper focuses on methodological improvements.

    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 does not release a high-risk system or new dataset. Any released code or checkpoints will follow standard responsible release practices.

    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: We cite the datasets, pretrained models, and codebases used in this work and we follow their corresponding usage terms and licenses.

    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: This paper does not release new assets.

    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: This 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: This 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: LLMs were only used for writing, editing, and formatting assistance, and did not affect the core methodology, experiments, scientific rigor, or originality of the research.

    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.