IDEAL: In-DEpth ALignment Makes A Discrete Representation AutoEncoder

Yitong Chen[1,2,*] **Zijie Diao**[1,*] **Junke Wang**[1,†] **Lingyu Kong**[1] **Yixuan Ren**[3] **Bo He**[3]
Yu-Gang Jiang[1] **Zuxuan Wu**[1,2,†]

[1]Institute of Trustworthy Embodied AI, Fudan University,
[2]Shanghai Innovation Institute,
[3]University of Maryland, College Park

Abstract

Built on pretrained vision foundation models (VFMs), representation autoencoders (RAEs) have recently emerged as a promising approach for constructing semantic-rich latent spaces for image generation. However, their reconstruction quality often remains suboptimal, largely because deep VFM features do not preserve sufficient fine-grained visual detail. This limitation becomes even more severe after discretization, where low-level information is difficult to recover. In fact, we observe that shallow VFM features retain considerably richer local appearance and structural detail, which complements the high-level semantics carried by deep features. Motivated by this complementary property, we propose **IDEAL**, an **I**n-**DE**pth **AL**ignment framework for discrete representation autoencoding. By jointly aligning quantized tokens with both shallow and deep VFM features, IDEAL enables the resulting discrete visual tokens to preserve both visual fidelity and semantics. Extensive experiments demonstrate that IDEAL yields superior reconstruction performance, achieving $\mathbf{0.61}$ rFID on ImageNet and outperforming the previous best method by $\mathbf{0.28}$. When used for autoregressive image generation, IDEAL further produces a gFID of $\mathbf{1.89}$, establishing a new state-of-the-art result. We open-source our code in https://github.com/ShareLab-SII/IDEAL.

1 Introduction

Pretrained vision foundation models (VFMs)[1, 2, 3, 4, 5, 6] encode images into semantically rich latent spaces that exhibit strong transfer across a broad spectrum of downstream vision tasks. More recently, representation autoencoders (RAEs)[7] have shown that such frozen VFM features can also serve as effective latent representations for diffusion-based image generation [8, 9, 10], improving both optimization efficiency and synthesis quality. This emerging connection between representation learning and generative modeling suggests that pretrained visual representations may offer a strong and scalable foundation for image generation.

However, this promising paradigm still faces a fundamental reconstruction bottleneck. Pretrained VFMs are primarily optimized for semantic discrimination [3, 6], rather than detail-preserving reconstruction [11, 12, 13]. As a consequence, their deep features emphasize high-level semantics but are relatively insensitive to fine-grained visual attributes such as color, texture, and local structure [14, 15]. Existing RAEs therefore remain suboptimal for faithful reconstruction, despite their benefits for generation. This issue is further amplified in autoregressive (AR) generation, where VFM latents must be discretized and missing low-level information is difficult to recover after quantization [16, 17].

*Equal contribution.
†Corresponding authors.

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

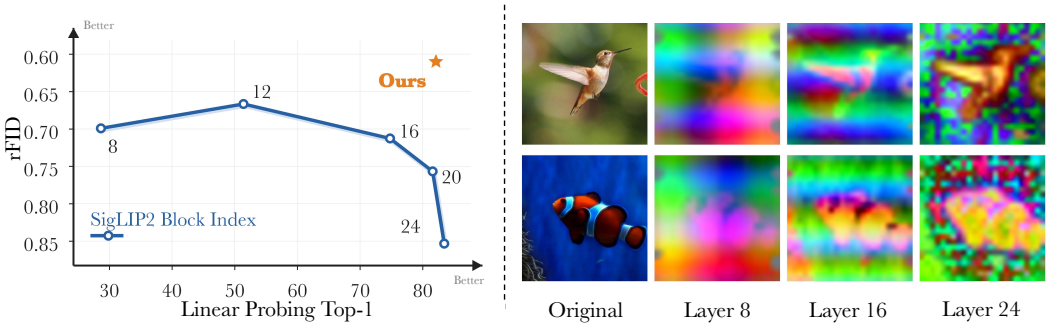

Figure 1: **(Left) Depth-wise linear probing of SigLIP2 [3] features.** Each point represents a different VFM block, showing the trade-off between reconstruction fidelity and semantic preservation: shallow blocks reconstruct better but are less semantic, while deeper blocks are more semantic but reconstruct worse. **(Right) PCA visualization.** By visualizing features across different layers of SigLIP2, we observe a consistent depth-dependent transition: the representations gradually evolve from low-level visual details to high-level semantic concepts.

In this work, we ask a simple question: how can discrete representation autoencoding capture fine-grained visual detail without sacrificing high-level semantics? To answer this question, we conduct a systematic depth-wise study by discretizing intermediate VFM representations and evaluating them from two complementary perspectives: semantic preservation and reconstruction fidelity. As shown in Fig. 1, a clear trade-off emerges across layers: shallow representations yield stronger reconstruction but weaker semantics, whereas deeper representations better preserve semantics at the cost of reconstruction fidelity. This trend is consistent with the hierarchical nature of VFMs, whose representations evolve from local texture and geometry in early layers to high-level semantic concepts in later layers [18, 19, 20]. Taken together, these findings point to a simple yet effective solution: rather than committing to a single layer for tokenization, we enrich deep semantic representations with shallow visual cues, yielding a unified representation that preserves rich semantics while supporting higher-fidelity reconstruction.

With this in mind, we propose **IDEAL**, a simple yet effective **I**n-**DE**pth **AL**ignment framework for discrete representation autoencoding. Rather than choosing a single VFM layer for tokenization, IDEAL combines appearance-rich shallow features with semantically informative deep features prior to vector quantization, forming a unified representation that preserves both visual details and high-level semantics. The resulting tokens are further supervised to recover the corresponding shallow and deep features, explicitly encouraging the discrete representation to retain information from both ends of the hierarchy. Finally, the reconstructed deep features are passed to a lightweight pixel decoder for high-fidelity image reconstruction. In this way, IDEAL turns frozen VFM features into discrete visual tokens that remain both semantically expressive and suitable for faithful reconstruction.

We evaluate IDEAL on ImageNet [21] from three complementary perspectives: reconstruction fidelity, semantic preservation, and autoregressive generation. For reconstruction, IDEAL obtains an rFID of $\mathbf{0.61}$, outperforming previous tokenizers by $\mathbf{0.28}$ and demonstrating the advantage of incorporating shallow appearance cues. For semantic preservation, the learned discrete representation maintains strong VFM semantics, reaching 80.89% zero-shot ImageNet classification accuracy. When used for autoregressive image generation, IDEAL yields a gFID of $\mathbf{1.89}$ on ImageNet at 256×256 resolution, establishing a new state of the art.

2 Related Work

Conventional Tokenizers. Existing tokenizers can be roughly divided into two categories: continuous tokenizers and discrete tokenizers. Continuous tokenizers are typically realized as VAEs, with an encoder parameterizing a continuous latent distribution and a decoder reconstructing images from it [11, 22, 8]. In contrast, discrete tokenizers (*e.g.*, VQ-VAE [23]) learn a finite codebook and quantize encoder features via nearest-neighbor lookup to yield token indices. Building on VQ-VAE, VQGAN [24] augments the reconstruction objective with perceptual and adversarial losses, while ViT-VQGAN [25, 13] further modernizes the tokenizer with Transformer-based architectures. Recent advances refine VQ-based tokenizers along two axes: improved quantization strategy to reduce

discretization error [26, 27, 28, 29, 16, 30], and more stable codebook update approach to mitigate codebook collapse [31, 32, 33].

VFM-based Tokenizers. Despite steady progress, most visual tokenizers still lack global semantic structure, which is significant for generation quality [7]. Recent advances show that incorporating pretrained VFM semantics during tokenization [34, 35, 36] or generation [37, 38] can substantially improve generation quality and training efficiency. These findings have spurred continuous semantic tokenizers like RAE [7], to directly apply tokenization on VFM features. FAE [39] then successfully reduces the high dimensional latent space of VFMs to a lower dimension using a single attention layer. On the discrete side, VQRAE [40] introduces vector quantization into the RAE framework to obtain discrete tokens. VFMTok [41] discretizes multi-scale frozen VFM features into codebook indices with deformable attention layers [42]. DINO-Tok [43] stabilizes vector quantization in DINO [5, 6] latent space through global PCA reweighting.

Autoregressive Visual Generation. Alongside diffusion-based approaches [9, 44, 45, 46, 47, 48, 49, 50], autoregressive (AR) modeling has also emerged as a major paradigm for visual generation. With a strong discrete visual tokenizer, images and videos can be compressed into discrete sequences suitable for next-token prediction. Autoregressive models then perform sequence modeling over these tokens and generate diverse high-quality images [17, 51, 52, 53, 54, 55] and videos [56, 57]. VAR [58] further redefines autoregressive learning from raster-scan next-token prediction to coarse-to-fine next-scale prediction. xAR [59] extends the autoregressive framework further by introducing next-X prediction, enabling flexible prediction targets such as tokens, cells, and entire images.

3 Method

3.1 Preliminary: Vector Quantized Image Tokenizers

A quantized image tokenizer is commonly formulated as an encoder $E(\cdot)$, a vector-quantizer $\mathrm{VQ}(\cdot)$ with a learnable codebook $C(\cdot)$, and a decoder $D(\cdot)$. Given an input image $x \in \mathbb{R}^{H\times W\times 3}$, the encoder first compresses it into a 2D patch embedding, and then applies a CNN/ViT backbone to produce the latent embedding z.

$$z = E(x) \in \mathbb{R}^{H/p\times W/p\times d}, \tag{1}$$

where p denotes the downsampling patch size and d is the channel dimension. The quantizer maintains a codebook $C = \{c_k\}_{k=1}^{K}$ with each $c_k \in \mathbb{R}^d$. For each spatial location i, the continuous embedding z_i is mapped to its nearest codebook entry:

$$\mathrm{VQ}(z_i) = \tilde{z}_i = c_{k_i}, \quad k_i = \arg\min_{k\in\{1,\dots,K\}} \|z_i - c_k\|_2. \tag{2}$$

The resulting discrete representation is the index map $\{k_i\}$, which can be flattened into a token sequence for AR modeling.

De-quantization retrieves the corresponding embeddings $\tilde{z}$ from the indices and decodes them back to the image domain. In practice, the decoder often consists of a feature-decoding backbone followed by a lightweight pixel head.

$$\hat{x} = D(\tilde{z}) = D(\mathrm{VQ}(z)). \tag{3}$$

To optimize the codebook, we use the standard VQ objective

$$\mathcal{L}_{\mathrm{VQ}} = \sum_i \|\mathrm{sg}(z_i) - c_{k_i}\|_2^2 + \beta\,\|\mathrm{sg}(c_{k_i}) - z_i\|_2^2\,, \tag{4}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator [60] and β is the weight of commitment loss [23].

For image reconstruction, we minimize an auto-encoding loss

$$\mathcal{L}_{\mathrm{AE}} = \mathcal{L}_2(x,\hat{x}) + \mathcal{L}_{\mathrm{P}}(x,\hat{x}) + \lambda_{\mathrm{G}}\,\mathcal{L}_{\mathrm{G}}(\hat{x}), \tag{5}$$

where $\mathcal{L}_2$ is a pixel-wise reconstruction loss, $\mathcal{L}_{\mathrm{P}}$ is a perceptual loss (*e.g.*, LPIPS [61]), and $\mathcal{L}_{\mathrm{G}}$ is an adversarial loss (*e.g.*, PatchGAN [62]) weighted by λ_{G}.

In this work, we follow the quantized-tokenizer paradigm above and focus on learning discrete codes that are suitable for AR modeling while preserving VFM semantics.

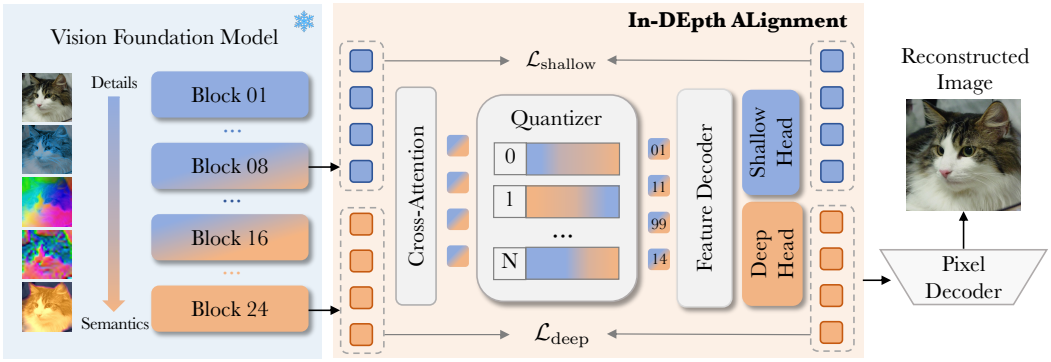

Figure 2: **Illustration of IDEAL.** IDEAL first extract shallow and deep features from a frozen VFM. A lightweight cross-attention module then fuses them into a unified representation. After vector quantization, a feature decoder reconstructs both shallow and deep features. The reconstructed deep semantic feature is finally mapped to pixels by a lightweight pixel decoder for image reconstruction.

3.2 Semantic-Spatial Complementarity in VFMs

Protocol. To understand which VFM features can provide fine-grained details for discrete semantic tokenization, we conduct a depth-wise probe by freezing a pretrained VFM $\Phi(\cdot)$ and tokenizing its intermediate features, as mentioned in Sec. 1. Given an image x, we extract a layer feature $f^{(\ell)} = \Phi_\ell(x)$, quantize it with the VQ module in Sec. 3.1, and reconstruct it in two steps: a feature decoder produces a reconstructed feature, which is then mapped to pixels by a pixel decoder. We evaluate each layer ℓ using (i) pixel reconstruction fidelity after the pixel decoder and (ii) linear probing classification Top-1 accuracy on the reconstructed feature.

Layer-wise trade-off. We probe a set of VFM layers $\ell \in \{8, 12, 16, 20, 24\}$ as tokenization targets. As shown in Table 3.2, shallow-layer features are easier to reconstruct with higher pixel fidelity, while their reconstructed features exhibit weak semantic transfer. Meanwhile, deeper-layer features preserve semantic ability better after quantization, but their pixel reconstruction performance tend to degrade. Overall, VFMs provide complementary signals across depth: shallow features are more reconstruction-friendly, whereas deep features are more semantic.

Table 1: **Layer-wise probing results on SigLIPv2 [3] features.** We report reconstruction fidelity using rFID and semantic preservation using linear probing classification Top-1 accuracy. Deeper layers retain more semantics, but leads to inferior reconstruction performance.

Layer	rFID↓	LP-Top1↑
8	<u>0.69</u>	28.66
12	**0.66**	51.40
16	0.71	74.78
20	0.75	<u>81.57</u>
24	0.85	**83.43**

3.3 IDEAL

Motivated by the complementary behavior of shallow and deep VFM features, we propose **IDEAL**, a VFM-based semantic tokenizer that produces discrete token indices for AR modeling and preserves semantic capability after de-quantization. The overall architecture of IDEAL is illustrated in Figure 2.

Frozen VFM encoder and fusion before quantization. We freeze a pretrained VFM $\Phi(\cdot)$ as the encoder. Given an image x, we extract a shallow feature $f^{(s)} = \Phi_{\ell_s}(x)$ and a deep feature $f^{(d)} = \Phi_{\ell_d}(x)$ from two VFM layers. In our setting, both features are sequences with matched shapes, *i.e.*, $f^{(s)}, f^{(d)} \in \mathbb{R}^{B\times L\times D}$, allowing fusion without any additional resizing or projection. We implement $\mathrm{AttnFuse}(\cdot)$ as a single lightweight cross-attention block where deep features provide queries and shallow features provide keys/values, followed by a Feed Forward Network(FFN) to produce the fused representation z.

$$z = \mathrm{AttnFuse}\Big(f^{(d)},\, f^{(s)}\Big)\,. \tag{6}$$

We adopt the VFM's original normalization for $f^{(d)}$ and a learnable normalization for $f^{(s)}$.

Vector quantization. To avoid introducing additional complexity, we quantize z using the standard VQ formulation in Equation 2, yielding discrete token indices y and de-quantized embeddings $\tilde{z}$. We apply an ℓ_2 normalization on codebook vectors to stabilize nearest-neighbor assignment during training. Following common practice [25], we apply down-factorization to map the fused feature z into a lower-dimensional quantization space before lookup, and recover the original dimension after de-quantization. This design mitigates codebook collapse and achieves full codebook utilization in our experiments. The resulting token indices y are used for AR modeling in Sec. 3.4.

Two-step decoding with dual feature heads. We decode $\tilde{z}$ using a ViT backbone feature decoder D_{feat} to reconstruct the unified feature.

$$g = D_{\text{feat}}(\tilde{z}). \tag{7}$$

Following previous work [41], we also append a [CLS] token and several register tokens to the input sequence to enhance representation learning and capture global context. These tokens are not used for reconstruction.

From g, we apply two lightweight linear heads to reconstruct the deep semantic feature and the shallow spatial feature, producing $\hat{f}^{(d)}$ and $\hat{f}^{(s)}$, respectively. We use $\hat{f}^{(d)}$ as the interface feature for semantic preservation evaluation, and also feed it into the pixel decoder to reconstruct the image:

$$\hat{x} = D_{\text{pixel}}\left(\hat{f}^{(d)}\right). \tag{8}$$

Objectives In addition to the standard VQ loss $\mathcal{L}_{\text{VQ}}$ (Equation 4) and the auto-encoding loss $\mathcal{L}_{\text{AE}}$ (Equation 5), we align reconstructed features with their VFM targets on both the deep and shallow branches. These alignment terms encourage the feature decoder to produce representations that preserve both semantic structure and fine-grained details.

The deep alignment loss is

$$\mathcal{L}_{\text{deep}} = \left\| \hat{f}^{(d)} - f^{(d)} \right\|_2^2 + \left(1 - \cos\left(\hat{f}^{(d)}, f^{(d)}\right)\right), \tag{9}$$

and the shallow alignment loss is

$$\mathcal{L}_{\text{shallow}} = \left\| \hat{f}^{(s)} - f^{(s)} \right\|_2^2 + \left(1 - \cos\left(\hat{f}^{(s)}, f^{(s)}\right)\right). \tag{10}$$

For the adversarial term in $\mathcal{L}_{\text{AE}}$, we replace the conventional PatchGAN discriminator [62] with a frozen DINOv1-s model [4], yielding semantically meaningful adversarial guidance that consistently improves reconstruction quality. The full objective is then

$$\mathcal{L} = \mathcal{L}_{\text{AE}} + \mathcal{L}_{\text{VQ}} + \mathcal{L}_{\text{deep}} + \mathcal{L}_{\text{shallow}}. \tag{11}$$

3.4 Autoregressive Image Generation

Once a tokenizer is trained, its discrete codes can be modeled by an autoregressive Transformer via next-token prediction. Let $y = (y_1, \ldots, y_T)$ denote the flattened token indices, and let c be the conditioning signal such as a class label or text embedding. An AR model parameterized by θ factorizes the likelihood as

$$p_\theta(y \mid c) = \prod_{t=1}^{T} p_\theta(y_t \mid y_{<t}, c), \tag{12}$$

and is trained with the standard cross-entropy objective

$$\mathcal{L}_{\text{AR}} = -\sum_{t=1}^{T} \log p_\theta(y_t \mid y_{<t}, c). \tag{13}$$

During sampling, the model generates $\hat{y}$ sequentially, after which the tokenizer decoder maps $\hat{y}$ back to an image. In the AR model, we use 2D RoPE [63] to better capture spatial locality.

Table 2: **ImageNet-1K reconstruction.** rFID uses outputs resized to 256×256. f is the downsampling factor; K and d are codebook size and dimension. ^{oim} denotes OpenImages training [64].

Method	f	K	d	Input res.	rFID↓	Usage (%)
Conventional tokenizer						
TiTok [65]	–	8192	64	256	1.05	100.0
ImageFolder [66]	–	32768	32	256	0.69	100.0
VQGAN [24]	–	16384	256	256	4.98	–
VQGAN [24]	–	8192	256	256	1.49	–
VQGAN^{oim} [24]	–	16384	4	256	1.19	–
ViT-VQGAN [25]	–	8192	32	256	1.28	95.0
MaskGiT [67]	16	–	–	256	2.28	–
VAR [58]	16	4096	32	256	0.92	100.0
RQ-VAE [26]	32	16384	256	256	1.83	–
LlamaGen [17]	16	16384	8	336	1.21	99.2
LlamaGen [17]	16	16384	8	384	0.95	99.7
VFM-based tokenizer						
VQRAE [40]	16	16384	1536	256	1.31	–
DINO-Tok [43]	16	16384×2	832	256	1.15	–
VFMTok [41]	–	16384	12	336	0.89	100.0
IDEAL *(Ours)*	16	16384	64	384	**0.61**	100.0



Figure 3: **Image reconstruction.** Input images are shown on the left and reconstructions on the right.

4 Experiments

4.1 Setup

Image tokenizer. We train IDEAL on ImageNet-1K [21] and evaluate reconstruction on its validation set, following the tokenizer training protocol of VQGAN [24] unless otherwise specified. Our default encoder is a frozen SigLIP2-Large/16 operating at 384×384 resolution [3]. The outputs of Transformer blocks 8 and 24 serve as the shallow and deep feature targets, $f^{(s)}$ and $f^{(d)}$, respectively. The 24×24 patch grid produces 576 spatially indexed tokens per image. We use a single vector-quantization (VQ) codebook with $K = 16{,}384$ entries of dimension $d = 64$ and a six-layer Transformer feature decoder [41].

Class-conditional generation. We train LlamaGen-style raster-scan autoregressive (AR) models [17] to perform categorical next-token prediction over the discrete image codes. The Base (111M)

Table 3: **Semantic preservation and multimodal transfer.** (a) uses frozen SigLIP2 text embeddings; (b) uses 576 visual tokens and a LLaMA 3.0 8B [70]

(a) Zero-shot ImageNet-1K classification

Visual feature	Top-1 (%)	Top-5 (%)
SigLIP2 [3]	**83.23**	**97.11**
IDEAL	80.89	96.40

(b) Multimodal transfer

Visual feature	RealWorldQA	ChartQA	OKVQA	InfoVQA	SEEDBench	MME
MAR VAE [71]	41.38	10.03	31.95	14.29	31.25	652
DINOv2 [5]	46.26	10.80	54.12	21.33	57.00	1345
SigLIP2 [3]	47.19	**13.80**	59.88	20.56	58.24	1730
IDEAL	**52.68**	12.48	**61.06**	**22.88**	**68.02**	**1878**

and Large (343M) generators are trained for 300 epochs, while the XXL (1.4B) and 3B (3.1B) generators are trained for 200 epochs. We report guided generation results using classifier-free guidance (CFG). For the full-training IDEAL models, the best reported CFG scale is 1.75 for Base and 1.25 for the remaining scales.

Evaluation. We evaluate reconstruction and generation using Fréchet Inception Distance (rFID and gFID) [68]. We also report spatial FID (sFID), precision, and recall [69]. Images are decoded at each tokenizer's native resolution and resized to 256×256 for reconstruction and generation evaluation, following the ImageNet-256 protocol [17]. Codebook usage measures the fraction of entries assigned at least once on the evaluation set. We assess semantic preservation through zero-shot ImageNet-1K Top-1 and Top-5 classification using frozen SigLIP2 text embeddings [1, 3], complemented by multimodal understanding benchmarks. The ablation-specific training and evaluation protocols are detailed in Section 4.3.

4.2 Main Results

Image reconstruction. We compare IDEAL with representative discrete image tokenizers, including conventional visual tokenizers like VQGAN [24] and semantic tokenizers like VFMTok [41]. As shown in Tab. 2, IDEAL achieves 0.61 rFID, outperforming prior VQ-based baselines under comparable settings while maintaining 100% codebook utilization. Beyond pixel fidelity, IDEAL also attains the highest rIS of 230.4, indicating IDEAL's strong semantic consistency between reconstructed and original images. This suggests that IDEAL improves reconstruction without sacrificing the semantic structure inherited from the VFM. Refer to Tab. 7 for a fully controlled comparison on 256 resolution.

Overall, IDEAL can achieve superior performance in both reconstruction fidelity and semantic consistency while still maintaining 100% usage, demonstrating that our design can substantially improve discrete autoencoding. Figure 3 presents qualitative reconstructions.

Semantic preservation. The decoded deep features of IDEAL remain compatible with the frozen SigLIP2 text embedding space. On zero-shot ImageNet-1K classification, IDEAL achieves Top-1 and Top-5 accuracies of 80.89% and 96.40%, respectively (Table 3(a)). The corresponding results for the original SigLIP2 features are 83.23% and 97.11%. The gaps of 2.34 and 0.71 percentage points quantify the effect of quantization and feature decoding on this task, while demonstrating that the discrete representation retains a usable image–text interface.

Multimodal transfer. Table 3(b) evaluates multimodal understanding with a common budget of 576 visual tokens. All visual backbones are frozen and use the same 8B language model, connector training policy, and LLaVA-NeXT-780k training data. IDEAL improves over the original SigLIP2 features on five of the six benchmarks.

Class-conditional image generation. We compare against representative mainstream generators, including diffusion models (Diff.) [9, 10, 72, 73], masked generation models (Mask.) [67], and

Table 4: **Class-conditional ImageNet-256 generation with CFG.** Outputs above 256×256 are resized for evaluation. † marks results re-implemented in [41]; "-re" marks rejection sampling.

Type	Method	#Epoch	#Params.	Res.	Generation w/ CFG			
					gFID↓	sFID↓	Pre.↑	Rec.↑
Diff.	MaskDiT [73]	1600	675M	256	2.28	5.67	0.80	0.61
	DiT [9]	1600	675M	256	2.27	4.60	0.83	0.57
	SiT [10]	1600	675M	256	2.06	4.50	0.82	0.59
	FasterDiT [72]	400	675M	256	2.03	4.63	0.81	0.60
Mask.	MaskGiT-re [67]	555	227M	256	4.02	–	–	–
VAR	VAR [58]	350	310M	256	3.30	–	0.84	0.51
AR	**Base (≈111M params)**							
	TiTok-B† [65]	300	111M	–	6.76	7.82	0.85	0.43
	LlamaGen-B [17]	300	111M	384	6.09	7.24	0.85	0.42
	VFMTok-B [41]	300	111M	336	3.43	5.88	**0.85**	**0.53**
	IDEAL-B (Ours)	300	111M	384	**3.38**	**5.18**	0.84	0.51
	Large (≈343M params)							
	TiTok-L† [65]	300	343M	–	4.03	6.93	0.84	0.52
	LlamaGen-L [17]	300	343M	384	3.07	6.09	0.83	0.52
	VFMTok-L [41]	300	343M	336	2.75	5.58	**0.84**	0.57
	IDEAL-L (Ours)	300	343M	384	**2.26**	**5.10**	0.81	**0.58**
	XXL (≈1.4B params)							
	LlamaGen-XXL [17]	200	1.4B	384	2.34	6.00	0.81	0.60
	VFMTok-XXL [41]	200	1.4B	336	2.19	5.53	0.83	**0.60**
	IDEAL-XXL (Ours)	200	1.4B	384	**1.95**	**4.81**	**0.83**	0.59
	3B params							
	LlamaGen-3B [17]	200	3.1B	384	2.19	5.97	0.82	0.58
	VFMTok-3B [41]	200	3.1B	336	2.07	6.23	0.81	**0.62**
	IDEAL-3B (Ours)	200	3.1B	384	**1.89**	**5.08**	**0.83**	0.59

autoregressive models (AR) built on visual tokenizers or semantic tokenizers [17, 58, 65, 41]. All AR baselines are trained and evaluated under the same protocol as LlamaGen [17].

As shown in Table 4, IDEAL yields strong generation performance compared to mainstream image generation models. Notably, at the Base scale, IDEAL-B achieves a gFID of 3.38, outperforming masked autoregressive baselines [67] with fewer parameters. IDEAL-B also substantially outperforms AR baselines trained on visual tokenizers such as LlamaGen [17], with a gain of 2.71 in gFID and a gain of 37.3 in gIS. When scaled to the Large scale, IDEAL-L further reduces gFID to 2.26, which is comparable to some competitive diffusion models [73, 9, 10, 72]. However, IDEAL requires much shorter training length and approximately half of the parameters needed by diffusion models, demonstrating the efficiency of our model.

Scaling IDEAL to larger models further improves generation quality. At the XXL scale, IDEAL-XXL reaches a gFID of 1.95 and the best sFID of 4.81, surpassing strong AR baselines such as VFMTok-XXL [41] and LlamaGen-XXL [17] under the same training length. Notably, when scaled to 3B parameters, IDEAL continues improving and achieves a gFID of **1.89**, establishing a new state-of-the-art result for autoregressive modeling.

VFMTok [41] achieves higher gIS than IDEAL at similar parameter counts. We attribute this difference to a well-known trade-off between IS and FID. IS emphasizes classification confidence, which does not necessarily reflect the image realism captured by FID. Moreover, IDEAL has tighter training constraints: its decoded feature must stay close to the underlying VFM semantic geometry while remain directly decodable by a CNN pixel head for high-fidelity reconstruction. These additional constraints reduce the degrees of freedom available for generation-optimality, which can manifest as lower gIS even when fidelity-oriented metrics (*e.g.*, gFID/sFID) remain strong.

Overall, these results show that IDEAL provides a three-in-one unified representation, supporting AR modeling without sacrificing VFM semantics. Figure 4 shows samples from IDEAL-L.



Figure 4: **Class-conditional image generation.** ImageNet samples generated by IDEAL-L.

Table 5: **Fusion and feature-recovery ablations.** Fusion combines shallow and deep VFM features before VQ; the decoder recovers the listed targets.

Variant	VQ input	Recovered features	rFID↓	gFID↓
Deep only	Deep	Deep	1.20	5.05
No fusion	Deep	Shallow + deep	1.11	4.85
Linear fusion	Linear fusion	Shallow + deep	0.84	4.55
Full IDEAL	Attention fusion	Shallow + deep	0.81	**4.13**
w/o shallow target	Attention fusion	Deep	0.91	4.54
w/o deep target	Attention fusion	Shallow	**0.74**	6.95

4.3 Ablation Study

Fusion and feature recovery. Table 5 studies whether shallow and deep features should be fused before quantization and recovered jointly from the discrete codes. We train six tokenizer variants from scratch on the same ImageNet data, using the same codebook and decoder configuration for 100k steps (approximately 20 epochs). For each tokenizer, we train a 111M Base AR model for 50 epochs and evaluate gFID using 50k samples at a fixed CFG scale of 1.75.

Adding a shallow recovery target to the deep-only code reduces rFID from 1.20 to 1.11 and gFID from 5.05 to 4.85. Fusing both feature depths before quantization improves both metrics further, with attention fusion outperforming linear fusion (0.81 versus 0.84 rFID and 4.13 versus 4.55 gFID). Removing shallow recovery from the full model worsens rFID to 0.91 and gFID to 4.54. Conversely, removing deep recovery gives the lowest rFID of 0.74 but worsens gFID to 6.95, showing that

Table 6: **Controlled comparison and backbone sensitivity.**

(a) Controlled comparison with VFMTok

Tokenizer	Frozen VFM	rFID↓	gFID↓
VFMTok-B [41]	DINOv2-B	1.02	3.61
IDEAL-B	SigLIP2-B	0.98	3.43
IDEAL-B	DINOv2-B	**0.95**	**3.31**

(b) Backbone sensitivity

Frozen VFM	rFID↓	gFID↓
SigLIP2-L [3]	0.61	4.31
DINOv2-L [5]	0.60	**3.81**
DINOv3-L [6]	**0.54**	3.90

Table 7: Comparison of tokenizer performance and AR generation at 256×256 resolution.

Approach	Image recon.			Usage↑	#Epochs	#Params.	AR gen.	
	#Toks	rFID↓	rIS↑				gFID↓	gIS↑
LlamaGen-B	256	2.22	169.8	95.2%	300	111M	5.46	193.6
VFMTok-B	256	1.02	213.2	100.0%	300	111M	3.61	247.6
IDEAL-B	256	0.98	220.0	100.0%	300	111M	3.43	181.9

stronger reconstruction alone does not guarantee a better representation for generation. Together, these results support jointly recovering both feature depths from a shared spatial codebook.

Controlled comparison with VFMTok. Table 6(a) compares IDEAL with VFMTok [41] using 256×256 inputs, 256 tokens per image, a 16K codebook, and a 111M Base AR generator trained for 300 epochs. With DINOv2-Base as the common frozen backbone, IDEAL reduces rFID from 1.02 to **0.95** and gFID from 3.61 to **3.31**. The SigLIP2-Base variant achieves 0.98 rFID and 3.43 gFID under the same resolution and AR training protocol.

Backbone VFM. Table 6(b) compares released IDEAL tokenizer checkpoints based on SigLIP2-Large, DINOv2-Large, and DINOv3-Large. For each checkpoint, we train a fresh 111M Base AR model for 50 epochs with one seed and evaluate 50k samples at CFG 1.75. DINOv3 yields the lowest rFID (**0.54**), whereas DINOv2 achieves the lowest gFID (**3.81**) under this short-run protocol. We retain SigLIP2 as the default for its image–text-aligned feature space and the zero-shot interface evaluated above, rather than for uniformly optimal reconstruction or generation performance.

Controlled 256×256 AR Generation. Following VFMTok, we train both the image tokenizer and the AR generation model at 256×256 resolution. The tokenizer is trained for 50 epochs, and the AR-Base model is trained for 300 epochs. As shown in Tab. 7, IDEAL improves over LlamaGen and VFMTok in both reconstruction quality and generation fidelity under this controlled setting.

5 Conclusion

We introduced **IDEAL**, a discrete representation autoencoder that converts VFM features into discrete codes while preserving both semantic richness and high-fidelity reconstructability. The design is motivated by a simple and empirical observation: the hierarchies of VFM features exhibit a clear depth-dependent trade-off, where shallow layers retain more spatial details and deep layers encode stronger semantics. Exploiting this complementarity, IDEAL injects reconstruction-relevant signals into deep semantic features, yielding a latent space that preserves both detailed visual information and strong semantics.

Experiments on ImageNet show that IDEAL delivers strong performance on both reconstruction and generation. Equipped with IDEAL, a 3B autoregressive image generation model achieves a gFID of $\mathbf{1.89}$ at 256×256, establishing a new state-of-the-art results.

Acknowledgments and Disclosure of Funding

Funding. This work was supported by the National Natural Science Foundation of China (No. 62472098) and the Science and Technology Commission of Shanghai Municipality (No. 25511106100).

Competing Interests. The authors declare no competing interests.

References

[1] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 1, 7

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

[3] 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, 4, 6, 7, 9, 15

[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 (ICCV)*, pages 9650–9660, October 2021. 1, 5

[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, 3, 7, 9

[6] 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. 1, 3, 9

[7] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders, 2025. 1, 3

[8] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. 1, 2

[9] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 1, 3, 7, 8

[10] 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, 2024. 1, 7, 8

[11] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 1, 2

[12] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *NeurIPS*, 2017. 1

[13] Junke Wang, Yi Jiang, Zehuan Yuan, Bingyue Peng, Zuxuan Wu, and Yu-Gang Jiang. Omnitokenizer: A joint image-video tokenizer for visual generation. In *NeurIPS*, 2024. 1, 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. 1

[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. 1

[16] 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. In *ICLR*, 2024. 1, 3

[17] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024. 1, 3, 6, 7, 8, 14, 15

[18] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *ICLR*, 2024. 2

[19] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai C Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. In *NeurIPS*, 2024. 2

[20] Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, et al. Perception encoder: The best visual embeddings are not at the output of the network. In *NeurIPS*, 2025. 2

[21] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 2, 6, 14, 21, 22

[22] Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2016. 2

[23] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2018. 2, 3

[24] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021. 2, 6, 7

[25] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan, 2022. 2, 5, 6

[26] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization, 2022. 3, 6

[27] Herve Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2011. 3

[28] Yue Zhao, Hanwen Jiang, Zhenlin Xu, Chutong Yang, Ehsan Adeli, and Philipp Krähenbühl. Spherical leech quantization for visual tokenization and generation, 2025. 3

[29] Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple, 2023. 3

[30] Zhuoyan Luo, Fengyuan Shi, Yixiao Ge, Yujiu Yang, Limin Wang, and Ying Shan. Open-magvit2: An open-source project toward democratizing auto-regressive visual generation. *arXiv preprint arXiv:2409.04410*, 2024. 3

[31] Lei Zhu, Fangyun Wei, Yanye Lu, and Dong Chen. Scaling the codebook size of vqgan to 100,000 with a utilization rate of 99%, 2024. 3

[32] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2, 2019. 3

[33] Yongxin Zhu, Bocheng Li, Yifei Xin, Zhihua Xia, and Linli Xu. Addressing representation collapse in vector quantized models with one linear layer, 2025. 3

[34] Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 3

[35] Junke Wang, Xiao Wang, Jiacheng Pan, Xuefeng Hu, Feng Li, Jingxiang Sun, Chaorui Deng, Zilong Chen, Yunpeng Chen, Kaibin Tian, et al. Arm: An autoregressive large multimodal model with unified discrete representations. *arXiv preprint arXiv:2606.11188*, 2026. 3

[36] Wujian Peng, Lingchen Meng, Yuxuan Cai, Xianwei Zhuang, Yuhuan Yang, Rongyao Fang, Chenfei Wu, Junyang Lin, Zuxuan Wu, and Shuai Bai. Unified multimodal autoregressive modeling with shared context-visual tokenizer is key to unification. In *ICML*, 2026. 3

[37] 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*, 2024. 3

[38] 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, 2025. 3

[39] Yuan Gao, Chen Chen, Tianrong Chen, and Jiatao Gu. One layer is enough: Adapting pretrained visual encoders for image generation, 2025. 3

[40] Sinan Du, Jiahao Guo, Bo Li, Shuhao Cui, Zhengzhuo Xu, Yifu Luo, Yongxian Wei, Kun Gai, Xinggang Wang, Kai Wu, and Chun Yuan. Vqrae: Representation quantization autoencoders for multimodal understanding, generation and reconstruction, 2025. 3, 6

[41] Anlin Zheng, Xin Wen, Xuanyang Zhang, Chuofan Ma, Tiancai Wang, Gang Yu, Xiangyu Zhang, and Xiaojuan Qi. Vision foundation models as effective visual tokenizers for autoregressive image generation, 2025. 3, 5, 6, 7, 8, 9, 10, 14, 15

[42] Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Vision transformer with deformable attention, 2022. 3

[43] Mingkai Jia, Mingxiao Li, Liaoyuan Fan, Tianxing Shi, Jiaxin Guo, Zeming Li, Xiaoyang Guo, Xiao-Xiao Long, Qian Zhang, Ping Tan, and Wei Yin. Dino-tok: Adapting dino for visual tokenizers, 2025. 3, 6

[44] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2022. 3

[45] Shuyuan Tu, Qi Dai, Zuxuan Wu, Zhi-Qi Cheng, Han Hu, and Yu-Gang Jiang. Implicit temporal modeling with learnable alignment for video recognition. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19879–19890. IEEE, 2023. 3

[46] Shuyuan Tu, Yueming Pan, Yinming Huang, Xintong Han, Zhen Xing, Qi Dai, Chong Luo, Zuxuan Wu, and Yu-Gang Jiang. Stableavatar: Infinite-length audio-driven avatar video generation. *arXiv preprint arXiv:2508.08248*, 2025. 3

[47] Shuyuan Tu, Qi Tian, Zihan Yang, Yue Wu, Xintong Han, Weijie Kong, Jiangfeng Xiong, Jian-Wei Zhang, Zhao Zhong, Liefeng Bo, et al. Baton: Explicit semantic blueprints for joint video-audio generation. *arXiv preprint arXiv:2605.25195*, 2026. 3

[48] Shuyuan Tu, Qi Tian, Yinming Huang, Yue Wu, Xintong Han, Kaihang Pan, Weijie Kong, Jiangfeng Xiong, Jian-Wei Zhang, Zuxuan Wu, et al. Prism: Dynamic sparse attention for native 2k joint video-audio generation model training. *arXiv preprint arXiv:2610.05416*, 2026. 3

[49] Yinming Huang, Shuyuan Tu, Xi Yan, Zihan Yang, Jianhua Han, Hang Xu, Kaihang Pan, Yu-Gang Jiang, and Zuxuan Wu. Va-judger: Reward modeling from human preference feedback for joint video-audio generation. *arXiv preprint arXiv:2608.18607*, 2026. 3

[50] Yinming Huang, Shuyuan Tu, Xi Yan, Jiahao Zhan, Zihan Yang, Zhen Xing, Hui Zhang, Tiehua Zhang, Yu-Gang Jiang, and Zuxuan Wu. Agentic visual generation: From generative models to agentic control. *arXiv preprint arXiv:2609.06758*, 2026. 3

[51] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation, 2022. 3

[52] Junke Wang, Zhi Tian, Xun Wang, Xinyu Zhang, Weilin Huang, Zuxuan Wu, and Yu-Gang Jiang. Simplear: Pushing the frontier of autoregressive visual generation through pretraining, sft, and rl. *arXiv preprint arXiv:2504.11455*, 2025. 3

[53] Yitong Chen, Zuxuan Wu, Xipeng Qiu, and Yu-Gang Jiang. Catok: Taming mean flows for one-dimensional causal image tokenization. In *CVPR*, 2026. 3

[54] Rui Tian, Mingfei Gao, Mingze Xu, Jiaming Hu, Jiasen Lu, Zuxuan Wu, Yinfei Yang, and Afshin Dehghan. Unigen: Enhanced training & test-time strategies for unified multimodal understanding and generation. In *NeurIPS*, 2025. 3

[55] Rui Tian, Mingfei Gao, Haiming Gang, Jiasen Lu, Zhe Gan, Yinfei Yang, Zuxuan Wu, and Afshin Dehghan. Unigen-1.5: Enhancing image generation and editing through reward unification in reinforcement learning. In *CVPR*, 2026. 3

[56] Songwei Ge, Thomas Hayes, Harry Yang, Xi Yin, Guan Pang, David Jacobs, Jia-Bin Huang, and Devi Parikh. Long video generation with time-agnostic vqgan and time-sensitive transformer, 2022. 3

[57] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers, 2021. 3

[58] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *NeurIPS*, 2024. 3, 6, 8

[59] Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Beyond next-token: Next-x prediction for autoregressive visual generation, 2025. 3

[60] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation, 2013. 3

[61] 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. 3

[62] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 3, 5

[63] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. 5, 16

[64] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128, 03 2020. 6

[65] Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. In *NeurIPS*, 2024. 6, 8

[66] Xiang Li, Kai Qiu, Hao Chen, Jason Kuen, Jiuxiang Gu, Bhiksha Raj, and Zhe Lin. Imagefolder: Autoregressive image generation with folded tokens, 2024. 6

[67] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *CVPR*, 2022. 6, 7, 8

[68] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 7

[69] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models, 2019. 7

[70] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 7

[71] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. In *NeurIPS*, 2024. 7

[72] Jingfeng Yao, Wang Cheng, Wenyu Liu, and Xinggang Wang. Fasterdit: Towards faster diffusion transformers training without architecture modification, 2024. 7, 8

[73] Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers, 2024. 7, 8

[74] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 15, 16

[75] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models, 2023. 16

[76] Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering, 2019. 16

[77] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read, 2019. 16

A IDEAL Implementation Details

A.1 Tokenizer Training Details

Overall, our tokenizer training recipe closely follows prior work VFMTok [41]. Since VFMTok uses a VFM with a patch size of 14 and an input resolution of 336, we use a patch size of 16 and an input resolution of 384 to maintain consistency in the feature map size. We train IDEAL on ImageNet-1K [21] training set using random resized crop and horizontal flip, with an input resolution of 384×384 and evaluating reconstructions at 256×256 following the common protocol [17]. IDEAL requires 2 days of training on 8 Nvidia H200 GPUs. We summarize some key training configuration of our tokenizer in Table 8.

Table 8: Tokenizer implementation details

Hyperparameter	Value	Hyperparameter	Value
Backbone			
VFM type	SigLIPv2-Large [3]	VFM input resolution	384
VFM training	Frozen	image_size / eval_image_size	384 / 256
decoder backbone	ViT	decoder layer_num	6
decoder hidden_dim	1024	decoder attn_head	8
decoder cls_num	1	decoder reg_num	4
decoder dropout	0.1		
General			
mixed_precision	bf16	ema	True
codebook_l2_norm	True	max_grad_norm	1.0
Loss			
reconstruction_weight	1.0	perceptual_weight	1.0
vq_loss_ratio	1.0	commit_loss_beta	0.25
Adversarial			
disc_type	dino	disc_loss / gen_loss	hinge / hinge
disc_weight	0.5	disc_start	20000
use_diff_aug	True		
Optimization			
epochs	50	global_batch_size	256
optimizer	AdamW [74]	lr / lr_scheduler	1e-4 / cosine
weight_decay	5e-2	beta1 / beta2	0.9 / 0.95

A.2 Autoregressive Training Details

Following VFMTok [41], an AR generator is trained to model the discrete token sequences produced by the tokenizer. However, VFMTok extracts token sequences on-the-fly using the tokenizer at each training epoch, which introduces additional overhead. In contrast, we follow the original LlamaGen training pipeline [17]: we apply ten-crop preprocessing to training images and pre-extract all token sequences offline before AR training, significantly improving training throughput. We train Base and Large models for 300 epochs, and train XXL and 3B models for 200 epochs, consistent with the scaling recipe in VFMTok. IDEAL-B takes approximately 34 hours of training on 8 Nvidia H200 GPUs. Key AR training hyperparameters are summarized in Table 9.

B Additional Qualitative Results

We provide additional qualitative results on image reconstruction and generation, and further analyze representative failure cases of both our tokenizer and autoregressive model.

B.1 Reconstruction Results

As shown in Figure 5, our tokenizer produces fine-grained reconstructions across diverse scenes and objects.

B.2 Generation Results

Figure 6 presents more samples demonstrating that our method can synthesize images with varied styles, subjects, and compositions.

Table 9: Autoregressive training and sampling configuration for IDEAL.

Hyperparameter	Value	Hyperparameter	Value
Training protocol			
token extraction	offline	image preprocessing	ten-crop
epochs (Base/Large)	300	epochs (XXL/3B)	200
EMA	True	mixed_precision	bf16
Optimization			
optimizer	AdamW [74]	lr	1e-4
weight_decay	0.05	beta1 / beta2	0.9 / 0.95
max_grad_norm	1.0	dropout_p	0.1
token_dropout_p	0.1	drop_path_rate	0.0
Architecture & conditioning			
class_token_num	1	class_dropout_prob	0.1
positional embedding	2D RoPE [63]	rope_base	10000
Sampling			
top_k	0	top_p	1.0
temperature	1.0		

B.3 Failure Cases

Despite these strengths, we observe degraded reconstruction quality on faces and text, as illustrated in Figure 7. We attribute this to the limited domain coverage of our tokenizer training data. In particular, our tokenizer is trained only on ImageNet, which contains sparse coverage of close-up faces and rich-text images. We do not incorporate additional face- or text-centric data either. In generation, Figure 8 shows that artifacts can still appear in fine-structure regions such as hands and faces, suggesting that post-training refinement on autoregressive models may be beneficial for further improving fidelity.

C Limitation and Future Work

Our tokenizer is trained mainly on ImageNet, which has limited domain coverage. Thus, reconstruction can degrade on faces, text, and other long-tail visual patterns. In addition, our semantic-preservation evaluation focuses on ImageNet zero-shot classification, which mainly reflects category-level semantics and does not fully cover broader semantic capabilities.

A direct next step is to pretrain or adapt the tokenizer on larger and more diverse datasets to improve coverage of faces, text, and long-tail domains. We also plan to evaluate the decoded interface feature on broader semantic benchmarks to better characterize semantic preservation [75, 76, 77]. Finally, our discrete-token formulation may naturally extend to videos by incorporating temporal consistency for semantic tokenization and generation.

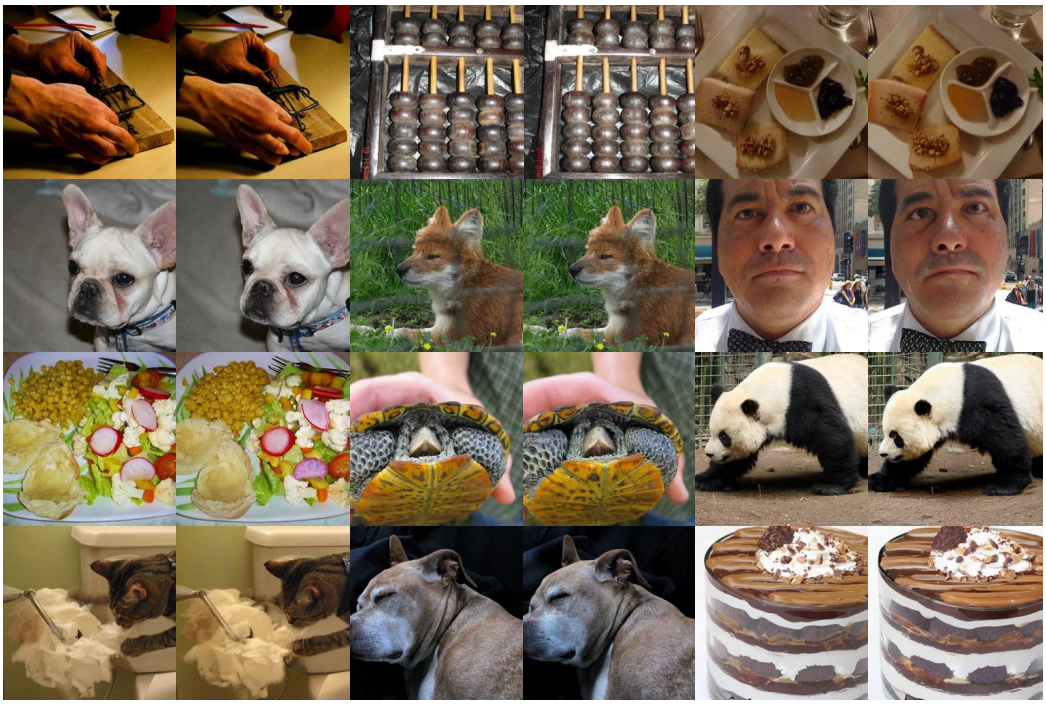

Figure 5: More visualization of reconstruction results from IDEAL. Left: input image; Right: output image.

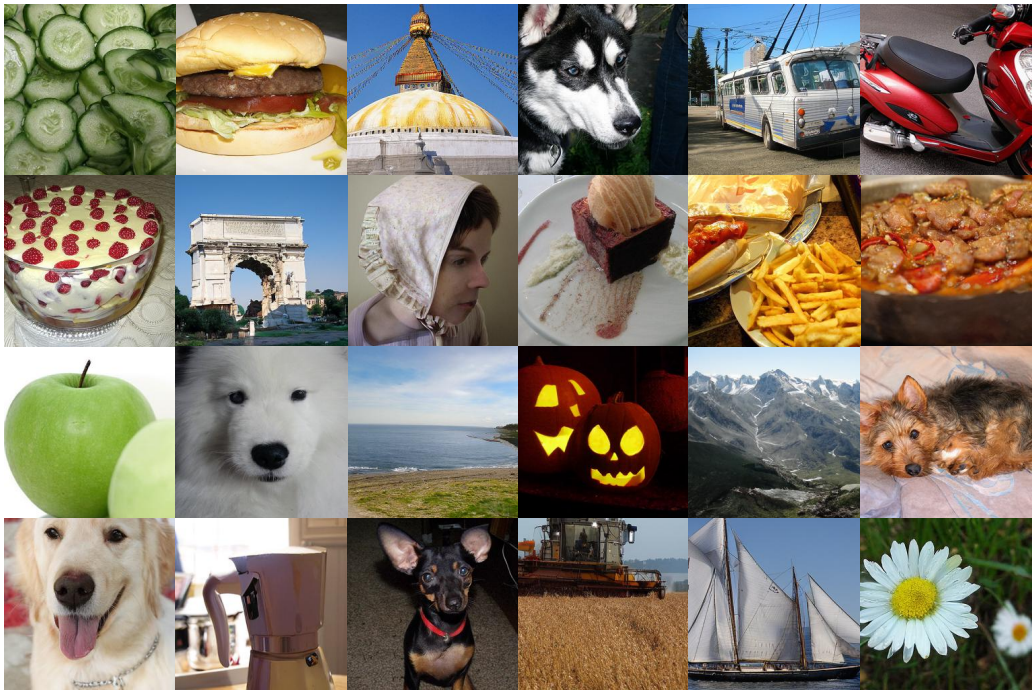

Figure 6: More visualization of class-conditional image generation results from IDEAL-L.

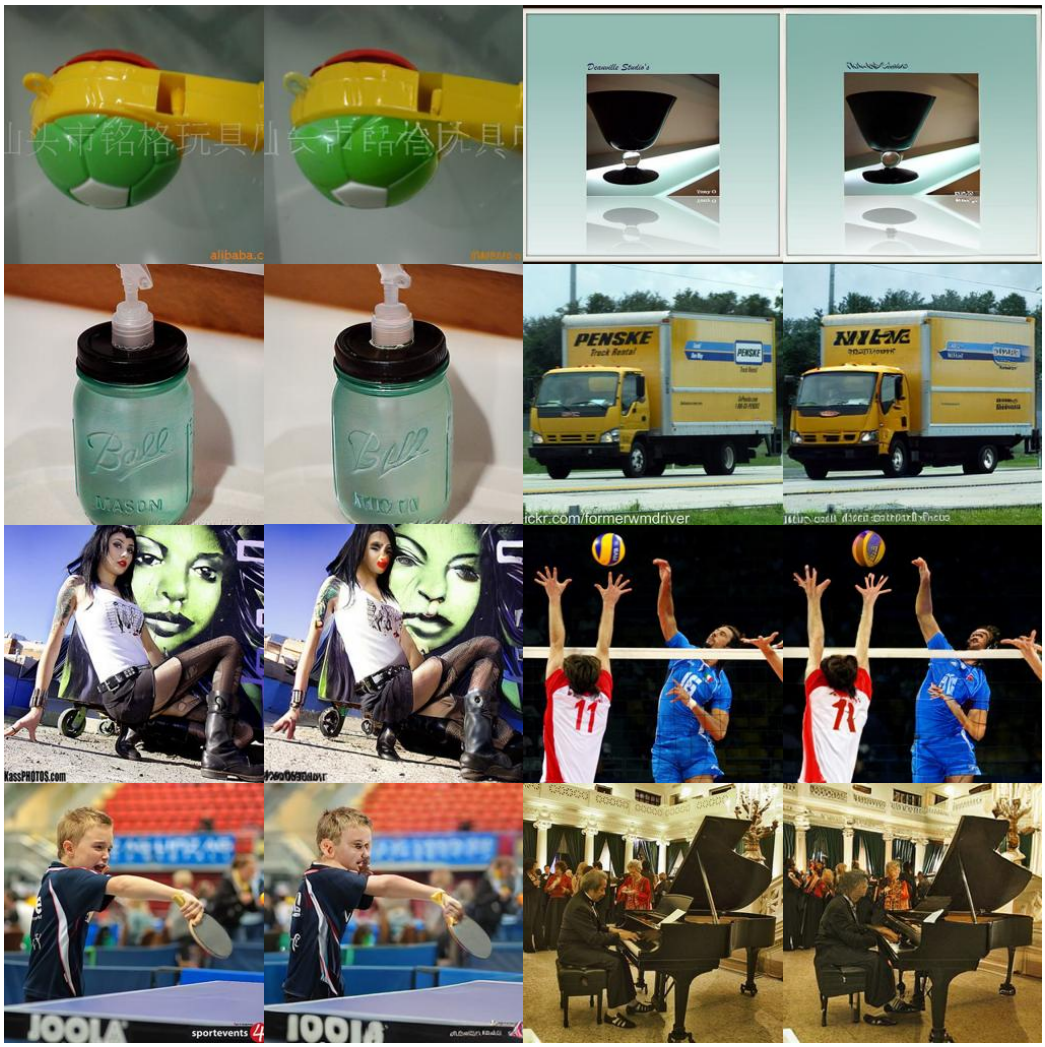

Figure 7: Visualization of failure reconstruction cases from IDEAL. Left: input image; Right: output image.



Figure 8: Failure generation cases. IDEAL still has artifacts in generating delicate text, human faces and fingers, which can be addressed with more training data on these images.

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. Our main contributions are also detailed in Sec. 1. Also see Sec. 4 for more theoretical and experimental evidence.

   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. C 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 use publicly-accessable dataset ImageNet [21], and we will release the 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-accessable dataset ImageNet [21], and we will release the 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 resource limitation, we do not report error bars. Please note that in Sec. 4, we spent numerous resources for different models and settings, which makes it prohibitively to run each experiments for multiple times.

   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. 4.1.

   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: This paper focuses on a academic, publicly-available datasets and benchmarks, and is not related to any private or personal data, hence there's no explicit negative social impacts.

    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: We do not foresee any high risk for misuse of this paper.

    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: Yes, we credited them in appropriate ways.

    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.