Vision Foundation Models Can Be Good Tokenizers for Latent Diffusion Models

Supplementary Material

Tianci Bi[1*] Xiaoyi Zhang[2] Yan Lu[2†] Nanning Zheng[1†]
[1] State Key Laboratory of Human-Machine Hybrid Augmented Intelligence,
Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
[2] Microsoft Research Asia
tiancibi@stu.xjtu.edu.cn, {xiaoyizhang, yanlu}@microsoft.com, nnzheng@mail.xjtu.edu.cn

Overview. This supplementary material provides additional technical details, analyses, and results that complement the main paper.

- Sec. 1 discuss the relation with concurrent works, represented by RAE [28] and SVG [20], clarifying the different design choice.
- Sec. 2 summarizes the CKNNA and SE-CKNNA metrics and provides the detailed evaluation setup used throughout our alignment analyses.
- Sec. 3 describes the architecture of VFM-VAE, including the latent projection, global or spatial branches, and detailed progressive decoder design.
- Sec. 4 presents additional experimental results: including detailed tokenizer reconstruction, representation alignment, and ablations; VFM-VAE compatibility across various VFMs; comparison with the fair VA-VAE baseline; and preliminary performance in high-resolution and unified text-to-image generation.
- Sec. 5 lists key architectural configurations, training hyperparameters for all stages, and training efficiency.
- Sec. 6 provides extended qualitative visualizations, including tokenizer reconstructions, stage-wise generation visualizations, and sample generation results.

1. The relation with concurrent works

We note that concurrent works, such as RAE [28] and SVG [20], share our core insight of leveraging a frozen Vision Foundation Model (VFM) as a tokenizer for LDMs. However, while RAE and SVG directly utilize high-dimensional VFM features, we strictly adhere to the latent channel compression characteristic of standard LDMs. Specifically, RAE focuses on enhancing the diffusion process through latent constraints, modified schedules, and autoguidance. In contrast, we shift the focus to specialized decoder architecture and loss design to bridge the gap between semantic-rich VFM features and pixel-accurate synthesis. Our VFM-VAE design offers the following advantages:

- **Systematic analysis of VFM-enhanced LDM training.** We provide empirical insights into how VFM-based encoders improve representation quality across all LDM layers (Fig. 4 of our main body). Our analysis establishes that dual-side alignment, applied simultaneously to tokenizers and diffusion models, produces superior VFM-aligned representations within the diffusion framework.
- **Seamless integration with existing VAE-LDM infrastructure.** By maintaining standard latent dimensions and projection layers, VFM-VAE remains fully compatible with the established VAE-LDM ecosystem. This design enables pre-computed latent caching by avoiding the prohibitive storage overhead of high-dimensional latents. Furthermore, our specialized decoder is explicitly designed to operate on highly compressed, low-dimensional latents, thereby effectively bridging the gap between semantic-rich VFM features and pixel-accurate image synthesis.

2. Representation alignment metrics

This section outlines the alignment metrics used in our analysis, all aimed at characterizing how two representations maintain local geometric structure. We detail CKNNA, which measures agreement in neighborhood relations, and SE-CKNNA, which extends this evaluation to semantic-preserving perturbations for a distribution-aware assessment of alignment stability.

2.1. CKNNA formulation

CKNNA [7] (Centered Kernel Nearest-Neighbor Alignment) is a locality-sensitive variant of CKA [9] (Centered Kernel Alignment). While CKA measures global similarity between two representations using centered kernel matrices,

*Work done during the internship at Microsoft Research Asia.
†Corresponding authors.

CKNNA focuses on **local neighborhood geometry** by restricting the comparison to mutual k-nearest-neighbor pairs. This makes it more suitable for evaluating whether representation spaces preserve fine-grained geometric structure.

Formally, given kernel similarity matrices $K, L \in \mathbb{R}^{n\times n}$ computed from two representations of the same n samples, we construct a mutual k-nearest-neighbor mask:

$$A_{ij} = \mathbf{1}\{i \neq j,\ j \in \mathrm{knn}_k^K(i)\ \wedge\ j \in \mathrm{knn}_k^L(i)\}. \quad (1)$$

The locally masked kernels are:

$$K^{(k)} = K \odot A, \qquad L^{(k)} = L \odot A. \quad (2)$$

Each masked kernel is then double-centered with $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$:

$$\widetilde{K} = HK^{(k)}H, \qquad \widetilde{L} = HL^{(k)}H. \quad (3)$$

Flattening and L_2-normalizing gives:

$$u = \frac{\mathrm{vec}(\widetilde{K})}{\|\widetilde{K}\|_F}, \qquad v = \frac{\mathrm{vec}(\widetilde{L})}{\|\widetilde{L}\|_F}. \quad (4)$$

The CKNNA score is the normalized inner product:

$$\mathrm{CKNNA}(X) = u^\top v \in [0, 1]. \quad (5)$$

Since $\|u\| = \|v\| = 1$, this is equivalently:

$$u^\top v = 1 - \tfrac{1}{2}\|u - v\|_2^2, \quad (6)$$

meaning it captures the discrepancy between the masked and centered local kernels of the two representations. Higher CKNNA values indicate better preservation of local neighborhood structure and therefore stronger alignment between the two representation spaces.

2.2. Semantic-Equivariant CKNNA formulation

SE-CKNNA (short for **S**emantic-**E**quivariant CKNNA) extends CKNNA to a **semantic-preserving perturbation distribution**. Rather than evaluating alignment only on clean images, it measures CKNNA under semantic-preserving transformations and aggregates the results. We provide visualized example of these preserving transformations on Fig. 1. Thus, SE-CKNNA is a distribution-aware enhancement of CKNNA, which is **not** a brand-new metric, but inherits all assumptions and limitations of the original formulation.

Let $\mathcal{T}$ denote a set of semantic-preserving transformations. In practice, these are uniformly sampled from:

- additive noise (added in the $[0, 1]$ pixel range) with strengths $\{0.05, 0.10, 0.15, 0.20\}$,
- scale interpolations $\{0.25, 0.50, 0.75, 1.0\}$,
- discrete rotations $\{0°, 90°, 180°, 270°\}$.

Noise introduces mild pixel-level perturbation within the normalized $[0, 1]$ range, while scaling and rotation modify only spatial arrangement and are treated jointly as equivariant transformations.

For each $T \in \mathcal{T}$, CKNNA is computed as:

$$\mathrm{CKNNA}(T) = u_T^\top v_T, \quad (7)$$

where u_T and v_T are the unit-norm vectorizations of the masked and double-centered kernels derived from the transformed features $\{F(Tx_i)\}_{i=1}^n$.

Since u_T and v_T lie on the unit sphere:

$$u_T^\top v_T = 1 - \frac{1}{2}\|u_T - v_T\|_2^2, \quad (8)$$

showing that CKNNA(T) measures the discrepancy between the transformed local kernels of the representations.

SE-CKNNA is defined as the expected CKNNA over a perturbation distribution $p(T)$:

$$\mathrm{SE-CKNNA} = \mathbb{E}_{T\sim p(T)}[u_T^\top v_T]. \quad (9)$$

Using Eq. (8), this can be written as:

$$\mathrm{SE-CKNNA} = 1 - \frac{1}{2}\,\mathbb{E}_{T\sim p(T)}\left[\|u_T - v_T\|_2^2\right], \quad (10)$$

clarifying that SE-CKNNA reflects the **average discrepancy** between the masked, centered local kernels induced by different semantic-preserving transformations. In practice, SE-CKNNA is the averaged CKNNA score across sampled transformations, and the relative deviation between SE-CKNNA and clean-image CKNNA reveals the stability of alignment under semantic-preserving perturbations.

2.3. Evaluation setup

Following [7], we use top-$k = 10$ with channel-wise normalization and outlier filtering before computing CKNNA. For layer-wise analysis in generative models, we use spatial tokens only (discarding [CLS] tokens as in REG [24]) and compute alignment after global pooling along the spatial dimension for both model and reference features. All generative models are evaluated in a unified setting, using the noised latent at diffusion timestep $t = 0.5$ with null-conditioning, consistent with the protocol in REPA [26].

3. Architectural details of VFM-VAE

This section provides implementation-level details of the VFM-VAE architecture that are omitted from the main paper for clarity. We describe the projection module used for latent compression and expansion, the structure of the progressive reconstruction blocks in the decoder, the upsampling unit, and the two-branch latent routing strategy. These components together form the complete tokenizer and decoder design used in our experiments.

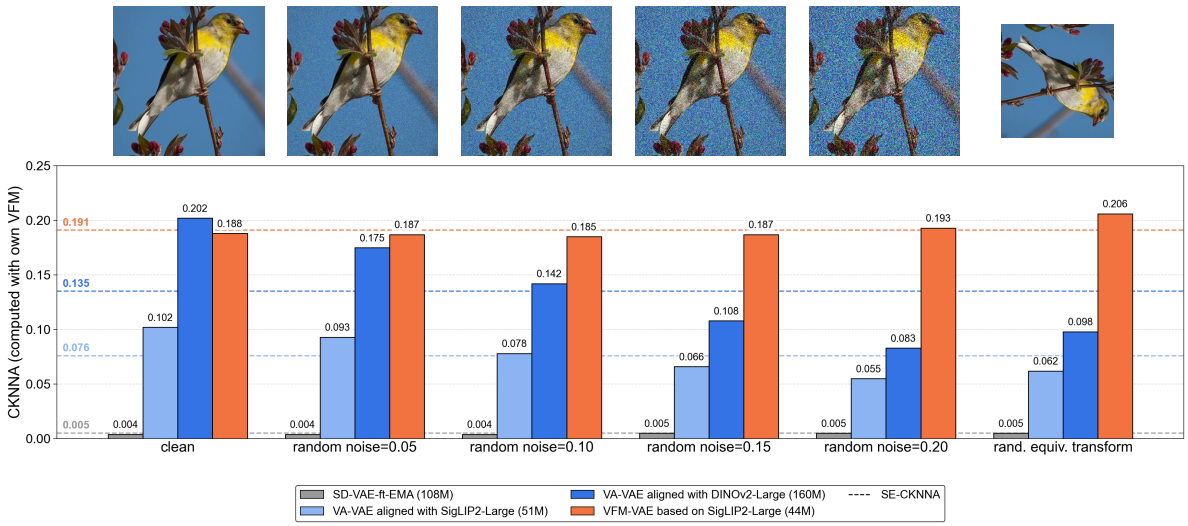

Figure 1. Brittleness of aligned representations under semantic-preserving transformations. Specifically, CKNNA [7] value for SD-VAE [17] is computed with DINOv2-Large [16]. *M* denotes the number of the training images in millions. Under semantic-preserving transformations, VFM-VAE demonstrates notably stronger alignment with VFM features than all VA-VAE variants and SD-VAE.

Table 1. Ablation of VFM-VAE components. Modules are added while maintaining light-weight alignment to the VFM.

Setting	Trainable params	rFID↓	rIS↑	LPIPS↓	PSNR↑	SSIM↑
SD-VAE-style Baseline	43.0M	19.69	74.9	0.456	14.59	0.264
+ Multi-scale Latent Fusion	88.0M	14.35	93.6	0.433	14.71	0.241
+ Our Modern Blocks	132.3M	1.08	194.6	0.291	18.06	0.388
+ Encoder Modifications	140.6M	**0.71**	**206.8**	**0.202**	**22.54**	**0.571**
Scaled SD-VAE-style Baseline	150.9M	38.46	40.3	0.549	13.59	0.208

3.1. Attention Projection

Motivation. We adopt the *Attention Projection* module from UniTok [14] for its simplicity, stability, and computational efficiency. It provides a unified mechanism for channel compression before sampling and decompression during decoding, maintaining a well-structured representation distribution that remains aligned with VFM features.

Encoder-side (compression). We extract shallow, middle, and final features from the frozen VFM. If their spatial sizes (and thus tokenizations) do not match the target latent configuration, we first apply `PixelShuffle` [21] to reconcile the spatial mismatch and then concatenate the tokens. **Importantly, the concatenated tokens are passed only once through the *Attention Projection*** to obtain the latent representation, from which we compute distribution statistics (e.g., mean and variance) and sample the latent code.

Decoder-side (decompression). During decoding, *Attention Projection* is used to expand channel dimensions and distribute the outputs to the semantic and spatial branches of the decoder, allowing the model to recombine abundant low-level cues for high-quality reconstruction.

3.2. Global and spatial branches

The decoder receives two complementary latent streams: a **global branch** and a **spatial branch**. The global branch is a lightweight MLP mapping network that transforms the pooled latent vector into a global style latent feature, providing semantic control and ensuring consistent appearance across scales. The spatial branch processes multi-resolution latent features through hierarchical convolutional blocks. Each latent is adapted to the target resolution via `PixelUnshuffle` or `PixelShuffle`, combined with 3×3 and 1×1 convolutions, `GroupNorm`, and `GELU` activation. Together, the two branches fuse holistic semantics with fine-grained spatial details, enabling the decoder to generate coherent and detailed reconstructions across progressive resolutions.

3.3. Progressive reconstruction blocks

Each reconstruction block $\mathcal{B}_i$ is responsible for processing and refining features at its designated spatial resolution. Let $\mathbf{h}^{(i)}$ denote the feature map at stage i, and let $\mathbf{z}_g$ and $\mathbf{z}_s^{(i)}$ be the global and spatial latents. The decoder updates $\mathbf{h}^{(i)}$

Table 2. Separate ablation of the global pooled feature. Modules are added while maintaining light-weight alignment to the VFM.

Method	gFID↓	gIS↑	rFID↓	rIS↑	rPSNR↑
w/o global feature	10.51	96.44	0.72	**208.86**	**22.56**
with global feature	**9.88**	**99.05**	**0.71**	206.80	22.54

Table 3. VFM-VAE's reconstruction metrics across 256 and 512 resolutions, evaluated on ImageNet validation set.

Resolution	rFID	rIS	LPIPS	PSNR	SSIM
256x256	0.52	214.1	0.221	22.99	0.593
512x512	0.44	235.8	0.210	25.52	0.682

using the following stage-dependent formulation:

$$\mathbf{h}^{(i)} = \begin{cases} \mathcal{B}_i(\mathrm{Up}(\mathbf{z}_s^{(1)}),\, \mathbf{z}_g), & i = 1, \\ \mathcal{B}_i(\mathrm{Up}(\mathrm{Concat}[\mathbf{h}^{(i-1)},\, \mathbf{z}_s^{(i)}]),\, \mathbf{z}_g), & 2 \le i \le 4, \\ \mathcal{B}_i(\mathrm{Up}(\mathbf{h}^{(i-1)}),\, \mathbf{z}_g), & 5 \le i \le 6, \end{cases} \tag{11}$$

where $\mathrm{Up}(\cdot)$ denotes a $2\times$ spatial upsampling. Spatial latents $\mathbf{z}_s^{(i)}$ are injected only in the first four stages, where coarse and mid-level structures are reconstructed. The final stages operate solely on the progressively refined features $\mathbf{h}^{(i-1)}$ under global modulation from $\mathbf{z}_g$, focusing on high-frequency detail synthesis.

3.4. Upsampling modules

We improve the StyleGAN-T [19] upsampling unit with PyTorch implementation for better readability and extensibility. The module normalizes input features via `GroupNorm`, applies 3×3 depthwise and 1×1 pointwise convolutions for local extraction and channel mixing, upsamples with `PixelShuffle`, and finally applies a fixed Gaussian blur to suppress checkerboard artifacts. It serves two roles: (1) progressively increasing spatial resolution across backbone blocks, and (2) refining features in the output pathway before the ToRGB head. This design retains StyleGAN-T's efficiency while improving stability and visual fidelity.

4. More ablation studies of VFM-VAE

In this section, we provide additional analyses of VFM-VAE, including detailed reconstruction and alignment results, its compatibility with different VFMs, comparison with VA-VAE fair baseline, higher-resolution generation, and text-to-image generation.

4.1. Detailed reconstruction metrics

To comprehensively evaluate the reconstruction capability of VFM-VAE, we report a full set of perceptual and pixel-level metrics, including LPIPS [27], PSNR, and SSIM, on both ImageNet-256 and ImageNet-512 validation set, as shown in Tab. 3. The 512-resolution model is obtained by fine-tuning from the 256-resolution pre-trained weights (details in Sec. 4.5). Benefiting from the architectural designs introduced in Sec. 3 and the main paper Sec. 3, VFM-VAE achieves considerable reconstruction performance across all metrics at both resolutions.

In addition, we provide more detailed ablation studies in Tab. 1. We attempt to scale a SD-VAE-like baseline to match the trainable parameter count of VFM-VAE. However, this leads to unstable or even collapsed training, ultimately resulting in worse reconstruction quality. Furthermore, in Tab. 2, we conduct a separate ablation on the global pooled feature design; removing it has almost no impact on reconstruction but weakens the generation performance. These results indicate that the additional parameters introduced in VFM-VAE together with its modular design are effective and more stable to train than SD-VAE original design, delivering consistent improvements across all reconstruction metrics.

4.2. Detailed representation alignment metrics

In Fig. 1, we further present a detailed CKNNA-based analysis under small image perturbations. The example images above illustrate the types of applied transformations, demonstrating that the semantic content remains intact. Besides the original VA-VAE [25] (aligned with DINOv2-Large [16]), we also train a fair VA-VAE variant that is aligned with the same SigLIP2 [23] backbone as VFM-VAE and uses a comparable amount of training data. Even under these matched conditions, VA-VAE still fails to achieve robust alignment. In contrast, VFM-VAE, which employs a frozen VFM encoder as its front-end, exhibits significantly stronger representation alignment performance.

4.3. Not only one VFM choice

We evaluate VFM-VAE's compatibility with three representative VFMs: EVA-CLIP-Large [22], DINOv2-Large [16], and SigLIP2-Large [23]. In detail, DINOv2-Large is a self-supervised vision model trained exclusively on image data, providing robust semantic representations without text alignment. EVA-CLIP-Large and SigLIP2-Large are VFMs trained on large-scale image-text pairs. While EVA-CLIP continue the original CLIP's contrastive learning approach to scale up, SigLIP2 employs a combination of image-language contrastive loss, caption prediction loss, and self-supervised loss to produce dense visual representations.

Table 4. Comparison of reconstruction and generation quality under fair setting of VFM and training duration. * denotes the reproduced VA-VAE by us with its open-sourced code. CKNNA is computed with each respective VFM. Details of CKNNA and SE-CKNNA are provided in Sec. 4.2. *Relative Change* represents the variation between SE-CKNNA and CKNNA, calculated as $|\text{SE-CKNNA} - \text{CKNNA}|/\text{CKNNA}$. Generation metrics are reported without CFG [5].

Tokenizer	VFM	Training Duration	CKNNA	SE-CKNNA	Relative Change	*Reconstruction*		*Generation*	
						rFID↓	rIS↑	gFID↓	gIS↑
VA-VAE	DINOv2-Large	160M (125 epochs)	0.202	0.135	-33.2%	**0.30**	213.6	5.14	130.2
VA-VAE*	SigLIP2-Large	51M (40 epochs)	0.102	0.076	-25.5%	0.84	207.4	7.83	115.1
VFM-VAE	SigLIP2-Large	44M (≈ 38 epochs)	0.188	0.191	+1.6%	0.52	**214.1**	**3.80**	**152.8**

Table 5. Comparison of VFM-VAE variants across different VFMs. Each tokenizer is trained through two-stage alignment (5M strong/weak). Generation results are reported at 100k training steps using LightningDiT-L/1 without CFG.

VFM	*Reconstruction*					*Generation*				
	rFID↓	rIS↑	LPIPS↓	PSNR↑	SSIM↑	gFID↓	sFID↓	gIS↑	Precision↑	Recall↑
EVA-CLIP-Large	**1.35**	188.4	**0.300**	**19.33**	**0.431**	4.40	5.13	146.4	**0.80**	**0.58**
DINOv2-Large	1.55	**199.8**	0.329	17.60	0.362	**4.00**	5.16	**147.1**	**0.80**	**0.58**
SigLIP2-Large	1.61	178.0	0.322	18.73	0.408	5.59	**4.85**	127.8	0.79	0.57

Under the same training schedule, we report both reconstruction and generation metrics in Tab. 5. In terms of reconstruction, EVA-CLIP-Large achieves the best performance across most metrics. While DINOv2-Large slightly outperforms SigLIP2-Large in realism-oriented metrics (rFID and rIS), it falls significantly short in perceptual (LPIPS) and pixel-level (PSNR and SSIM) metrics. Interestingly, in terms of generation performance, DINOv2 leads the field, followed by EVA-CLIP-Large, with SigLIP2-Large ranking last. This superiority likely stems from DINOv2's purely image-centric self-supervised pre-training, which makes it more compatible with nearly pure image generation tasks like ImageNet, where only class information is introduced.

Despite these findings, considering SigLIP2-Large's diverse training objectives, it provides more robust downstream performance and native text-alignment capabilities, which are crucial for text-to-image (T2I) generation (see Sec. 4.6). We therefore adopt SigLIP2-Large as our default VFM for long-term training to balance reconstruction, alignment, and T2I performance. Even though SigLIP2 is not the optimal choice for an object-oriented dataset like ImageNet, we still achieve impressive results in both reconstruction and generation. These findings demonstrate that VFM-VAE maintains strong performance across diverse VFM families, confirming its architectural generality rather than a dependence on any specific model.

The differing feature emphases of the DINOv2-Large and SigLIP2-Large variants of VFM-VAE, together with the performance gap observed between their corresponding VFM-VAE variants, particularly the weaker pixel-level reconstruction of the DINOv2-Large variant, naturally raise a key question: *Are the two observed gaps, the tokenizer's alignment gap and the downstream diffusion generation gap between VFM-VAE based on SigLIP2-Large and VA-VAE aligned with DINOv2-Large, simply consequences of differences in the underlying VFMs?* We address this issue in Sec. 4.4.

4.4. Improvement is not due to a different VFM

VFM-VAE consistently achieves faster convergence and higher generation quality than the VA-VAE baseline across all generative models. A key distinction, however, lies in the choice of underlying VFMs for alignment: VFM-VAE is built upon SigLIP2-Large, trained primarily with contrastive objectives to support both vision-language alignment and dense visual representation learning, whereas VA-VAE is aligned with DINOv2-Large, a purely vision-based self-supervised model. This difference introduces a natural trade-off between alignment and reconstruction quality. As shown in Sec. 4.3, when VFM-VAE is aligned with DINOv2-Large, its reconstruction slightly lags behind that of the SigLIP2-Large variant.

To determine whether the performance gap originates from the VFMs themselves, we conducted a controlled comparison. Following VA-VAE's strong alignment setup, we retrained a VA-VAE using **SigLIP2-Large on 51M images (≈ 40 epochs)**, achieving adequate reconstruction performance before training the same generative model.

Combining Fig. 1 and Tab. 4, two observations emerge:

- In terms of the tokenizer's intrinsic alignment and reconstruction capability, VA-VAE requires substantially more training images to achieve competitive alignment and reconstruction performance, whereas VFM-VAE attains a more favorable balance with significantly fewer training images and exhibits more robust alignment to its underly-

Table 6. Comparison of ImageNet-512 generation performance after 100k training steps without CFG using LightningDiT-B/1. VFM-VAE demonstrates improved generation quality over the VA-VAE baseline.

Method	gFID↓	sFID↓	gIS↑	Precision↑	Recall↑
VA-VAE + LightningDiT-B/1	21.42	**5.65**	55.3	0.75	**0.60**
VFM-VAE + LightningDiT-B/1	**18.05**	7.11	**69.6**	**0.78**	**0.60**

ing VFM. Even when using the same VFM, the VA-VAE variant still experiences a notable drop in alignment under small input perturbations.

- In terms of generative performance, even when the VFM choice, the number of training images, and the diffusion-training configuration are matched, the VA-VAE system still lags significantly behind the VFM-VAE system.

In summary, the performance advantage of VFM-VAE does not arise merely from employing a stronger VFM. Instead, it stems from its architectural design, which leverages frozen VFM features as the starting point, enabling stronger and more stable representation learning the generative learning through the whole pipeline.

4.5. Preliminary generation on 512×512 resolution

Limited by resources, we present ablation studies comparing our tokenizer with VA-VAE on reduced model sizes. We evaluate VFM-VAE on ImageNet-512 [18].

Setting. To maintain continuity with the 256-resolution setup, we reuse the same VFM configuration (SigLIP2-Large-Patch16-512 [23]) and all corresponding model components, and fine-tune them jointly at 512 resolution. The main difference from the 256-resolution setup lies in how images are fed into the VFM: while 256-resolution images were first upsampled 2× before being passed into the VFM, the 512-resolution images are directly fed into the VFM, which avoids interpolation overhead and reduces memory consumption.

To enable efficient fine-tuning from the 256-resolution model, we set the output dimension of the "encoded features" in Tab. 11 to 256, and retrain only the Attention Projection module that extracts multi-layer VFM features. All remaining modules reuse the pretrained weights from the 256-resolution model. The training hyperparameters follow the first three stages in Tab. 12; we only adjust the number of fine-tuning images for the strong-alignment, weak-alignment, and SSIM phases to 1M, 500k, and 500k respectively, while keeping the multi-scale pixel loss enabled throughout. The reconstruction results after 512-resolution fine-tuning are reported in Tab. 3.

Results. Finally, we train both VA-VAE and VFM-VAE with LightningDiT-B/1 for 100k steps (80 epochs) on ImageNet-512 and report generation performance without CFG [5]. As shown in Tab. 6, the VFM-VAE system achieves clearly superior generative performance compared with the VA-VAE system.

4.6. Text-to-image generation

Motivation. While the main paper focuses on class-based image generation, it remains important to examine how VFM-VAE performs when integrated into text-to-image generation and multimodal systems. A strong visual tokenizer should not only reconstruct accurately but also provide semantically consistent latents that interface smoothly with Vision-Language Models (VLMs). To validate this, we combine VFM-VAE and VA-VAE respectively with **BLIP3-o** [3] in a unified text-to-image framework and compare their effectiveness in generative modeling. Due to limited computing resources, here we present an ablation comparing our tokenizer with VA-VAE on limited training steps.

Setting. Given a text prompt, BLIP3-o first produces a fixed number of tokens (default 64). We extract the hidden states before the LM head as semantic conditioning for the diffusion model. During training, the diffusion model predicts a latent that is *flow-matched* [12] to the VAE encoder latent; during inference, this latent is decoded by the VAE decoder to generate the final image with 256 resolution. The visual–language backbone is **Qwen2.5-VL-3B-Instruct** [1], and the diffusion backbone is Lumina-Next (DiT) [29], where the patch size is reduced to 1 and the input/output channels are aligned to a latent of $16 \times 32 \times 32$. We pretrain the system for **one epoch** on the official BLIP3-o pretraining corpora, focusing solely on the text-to-image objective. Since more than 80% of the dataset consists of long prompts, we omit GenEval and evaluate exclusively on DPG-Bench (higher score is better) [6] and MJHQ-30K (lower gFID is better) [11].

DPG-Bench results. VFM-VAE + BLIP3-o achieves a higher overall score (59.1) than VA-VAE + BLIP3-o (55.4) (see Tab. 7). At the L1 level, improvements are evident in *relation*, *global*, and *other* categories, while a slight decrease is observed in *entity*. This indicates that VFM-VAE provides stronger global and relational understanding, enhancing compositional reasoning and text–image alignment, albeit with slightly weaker recall of local structures.

MJHQ-30K results. VFM-VAE + BLIP3-o also achieves significantly lower gFID across nearly all categories (see Tab. 8), reducing the overall score from 23.00 to 16.98. Notable gains are seen in *animals* ($44.78 \rightarrow 32.08$), *fashion* ($42.80 \rightarrow 30.27$), *indoor* ($44.01 \rightarrow 34.37$), and *people*

$(48.65 \rightarrow 36.62)$, with comparable results on *plants* and a minor trade-off in *logo*.

Summary. Under identical VLM and diffusion backbones, replacing VA-VAE with VFM-VAE yields more semantically aligned and generation-friendly latents, leading to higher text–image consistency, and overall better visual quality—even with only one epoch of pretraining.

5. Implementation details

VFM-VAE training. Model hyperparameters are provided in Tab. 11, with the stable training recipe detailed in Tab. 12. We find that training stability is preserved under moderate weight adjustments, provided that the adversarial loss remains stable. Our multi-stage training strategy follows the general structure of VA-VAE [25]. In the strong alignment stage, large representation regularization losses are applied to quickly establish VFM–VAE alignment. In the weak alignment stage, the weight of this loss is reduced to maintain alignment while shifting focus toward reconstruction quality. Notably, VA-VAE is trained on the full ImageNet training set (1,281,167 images), whereas VFM-VAE is trained on a filtered subset containing only images with a minimum resolution of 256 (1,152,196 images). We further introduce two fine-tunings:

- **SSIM fine-tuning.** Rapid convergence in reconstruction can occasionally cause RGB channel misalignment, leading to color noise near edges. We apply an SSIM loss for targeted refinement.
- **PatchGAN [8] fine-tuning.** The original DINO [2]-based discriminator, due to its large patch size, provides weak supervision for fine details. Adding a finer-grained PatchGAN discriminator improves reconstruction fidelity.

The improvements of reconstruction and alignment across the training stages are summarized in Tab. 9.

Diffusion model training. LightningDiT follows the same configuration as the VA-VAE system. When using REG, we apply several adjustments: the latent size changes from $32 \times 32 \times 4$ to $16 \times 16 \times 32$; the SiT-XL [15] patch size is reduced from 2 to 1; batch size is increased from 256 to 1024; the learning rate is doubled; β_2 of AdamW [13] optimizer is reduced from 0.999 to 0.95; and QK normalization [4] is added to the attention module. These modifications collectively stabilize long-term training. All experiments are conducted on a single node with $8 \times$192GB NVIDIA B200 GPUs. In Tab. 10, we further measure total wall-clock time to reach equivalent performance. VFM-VAE shows clear speedup over VA-VAE.

6. More qualitative results

We provide additional qualitative results: tokenizer reconstruction comparisons are shown in Fig. 2, visualizations across the training stages of the generative model are presented in Fig. 3, and 256-resolution generation examples from VFM-VAE + REG are provided from Fig. 4 to Fig. 9.

Table 7. Text-to-image generation results with BLIP3-o on DPG-Bench (higher score is better, evaluated at 256 resolution after 1 epoch of pretraining). VFM-VAE + BLIP3-o achieves higher overall scores, indicating stronger text–image alignment.

Text-to-image Model	entity	global	other	attribute	relation	overall
VA-VAE + BLIP3-o	**73.2**	69.1	71.2	70.1	77.9	55.4
VFM-VAE + BLIP3-o	68.4	**70.9**	**75.2**	**71.1**	**80.0**	**59.1**

Table 8. Text-to-image generation results with BLIP3-o on MJHQ-30K (lower gFID is better, evaluated at 256 resolution after 1 epoch of pretraining). VFM-VAE + BLIP3-o achieves significantly lower gFID.

Text-to-image Model	animals	art	fashion	food	indoor	landscape	logo	people	plants	vehicles	overall
VA-VAE + BLIP3-o	44.8	43.4	42.8	46.1	44.0	47.4	**59.0**	48.7	**50.7**	40.5	23.0
VFM-VAE + BLIP3-o	**32.1**	**36.0**	**30.3**	**44.7**	**34.4**	**41.3**	60.4	**36.6**	50.9	**39.2**	**17.0**

Table 9. Reconstruction and alignment performance across 256-resolution training stages of VFM-VAE. CKNNA is computed wtih SigLIP2-Large.

Training stage	rFID↓	rIS↑	LPIPS↓	PSNR↑	SSIM↑	CKNNA↑
Stage 1: Strong alignment	1.05	198.2	0.266	20.39	0.473	**0.221**
Stage 2: + Weak alignment	0.60	210.3	**0.196**	22.89	0.586	0.188
Stage 3: + SSIM fine-tuning	0.54	211.2	0.209	22.85	0.586	0.188
Stage 4: + PatchGAN fine-tuning	**0.52**	**214.1**	0.221	**22.99**	**0.593**	0.188

Table 10. Comparison of generative performance and training costs in A100 GPU hours.

Method	gFID↓	gIS↑	A100 GPU hours↓		
			VAE	diffusion	total
VA-VAE + LightningDiT(800 epochs)	2.17	205.6	2836	3451	6287
VFM-VAE + LightningDiT(560 epochs)	**2.06**	**205.8**	**2400**	**2419**	**4819**

Table 11. VFM-VAE architecture hyperparameters at 256-resolution training.

Category	**Name**	**Value**
VFM Backbone	VFM name	SigLIP2-Large-Patch16-512
Encoded Features	from which layers	[0, 12, -1]
	output dims	[64, 64, 64]
Latent	how to compress / decompress	*Attention Projection*
	spatial-compression ratio	16
	latent channels	32
	channel-decompression ratio	32
Spatial Control	block indices	[0, 1, 2, 3]
	mapped dims	[512, 256, 128, 128]
Attention	block indices	[0, 1, 2]
	attention depths	[2, 2, 2]

Table 12. VFM-VAE training hyperparameters at 256-resolution training.

Setting	Strong Alignment	Weak Alignment	SSIM Fine-tuning	PatchGAN Fine-tuning
Batch size			512	
Optimizer			Adam	
Betas			(0.0, 0.99)	
Learning rate	1×10^{-4}	1×10^{-4}	5×10^{-5}	5×10^{-5}
L1 loss weight	1.0	1.0	1.0	-
LPIPS loss weight	10.0	10.0	2.0	-
DINO discriminator loss weight	1.0	1.0	1.0	1.0
PatchGAN discrminator loss weight	-	-	-	1.0
Feature matching loss weight [8]	-	-	-	10.0
SSIM loss weight	-	-	1.0	-
Multiscale pixel loss weight	0.1 (to 5M = 0)	-	-	-
Representation reglarization loss weight	5.0	1.0	-	-
KL loss weight	1×10^{-6}	1×10^{-6}	-	-
Trainable parameters	Entire tokenizer	Entire tokenizer	The decoder	The second half of the decoder
Equivariance regularization [10]	Yes	Yes	Yes	No
Training images	20M	20M	1M	3M

Figure 2. Qualitative comparison of reconstructions from different VAEs.

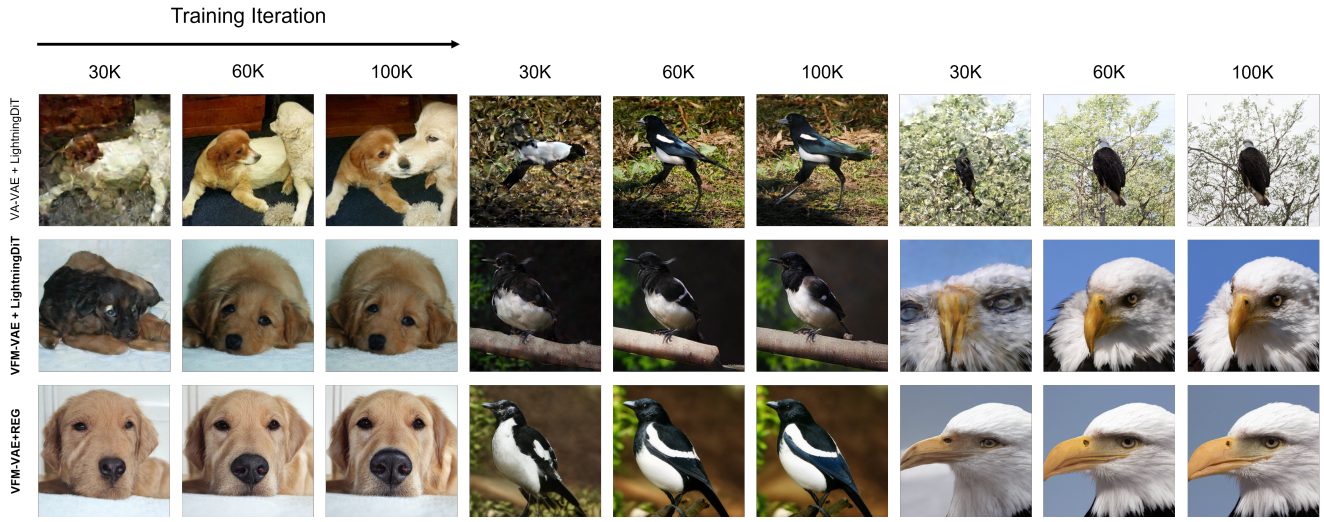

Figure 3. Stage-wise visualization of generative model training results. Shown under a fixed random seed and identical initial noise, our approach demonstrates impressive performance and greatly accelerates image generation learning.

Figure 4. **Visualization of VFM-VAE + REG (640 epochs).** Generation uses CFG with $w = 4.0$; class label is "Border collie" (232).

Figure 5. **Visualization VFM-VAE + REG (640 epochs).** Generation uses CFG with $w = 4.0$; class label is "Macaw" (88).

Figure 6. **Visualization of VFM-VAE + REG (640 epochs).** Generation uses CFG with $w = 4.0$; class label is "Bald Eagle" (22).

Figure 7. **Visualization of VFM-VAE + REG (640 epochs).** Generation uses CFG with $w = 4.0$; class label is "Giant Panda" (388).

Figure 8. **Visualization of VFM-VAE + REG (640 epochs).** Generation uses CFG with $w = 4.0$; class label is "Lakeside" (975).

Figure 9. **Visualization of VFM-VAE + REG (640 epochs).** Generation uses CFG with $w = 4.0$; class label is "Volcano" (980).

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 6

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 7

[3] Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025. 6

[4] Alex Henry, Prudhvi Raj Dachapally, Shubham Pawar, and Yuxuan Chen. Query-key normalization for transformers. *arXiv preprint arXiv:2010.04245*, 2020. 7

[5] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 5, 6

[6] Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024. 6

[7] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*, 2024. 1, 2, 3

[8] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 7, 9

[9] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMlR, 2019. 1

[10] Theodoros Kouzelis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Eq-vae: Equivariance regularized latent space for improved generative image modeling. *arXiv preprint arXiv:2502.09509*, 2025. 9

[11] Daiqing Li, Aleks Kamko, Ehsan Akhgari, Ali Sabet, Linmiao Xu, and Suhail Doshi. Playground v2. 5: Three insights towards enhancing aesthetic quality in text-to-image generation. *arXiv preprint arXiv:2402.17245*, 2024. 6

[12] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 6

[13] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 7

[14] Chuofan Ma, Yi Jiang, Junfeng Wu, Jihan Yang, Xin Yu, Zehuan Yuan, Bingyue Peng, and Xiaojuan Qi. Unitok: A unified tokenizer for visual generation and understanding. *arXiv preprint arXiv:2502.20321*, 2025. 3

[15] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pages 23–40. Springer, 2024. 7

[16] 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. *arXiv preprint arXiv:2304.07193*, 2023. 3, 4

[17] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 3

[18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 6

[19] 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 *International conference on machine learning*, pages 30105–30118. PMLR, 2023. 4

[20] Minglei Shi, Haolin Wang, Wenzhao Zheng, Ziyang Yuan, Xiaoshi Wu, Xintao Wang, Pengfei Wan, Jie Zhou, and Jiwen Lu. Latent diffusion model without variational autoencoder. *arXiv preprint arXiv:2510.15301*, 2025. 1

[21] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016. 3

[22] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023. 4

[23] 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. 4, 6

[24] Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, Jian Yang, et al. Representation entanglement for generation: Training diffusion transformers is much easier than you think. *arXiv preprint arXiv:2507.01467*, 2025. 2

[25] Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15703–15712, 2025. 4, 7

[26] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024. 2

[27] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 4

[28] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025. 1

[29] Le Zhuo, Ruoyi Du, Han Xiao, Yangguang Li, Dongyang Liu, Rongjie Huang, Wenze Liu, Xiangyang Zhu, Fu-Yun Wang, Zhanyu Ma, et al. Lumina-next: Making lumina-t2x stronger and faster with next-dit. *Advances in Neural Information Processing Systems*, 37:131278–131315, 2024. 6