# On the Diffusibility of High-Dimensional Latents

Chao Feng[1] Zhiyang Xu[3] Bowei Chen[4] Yuanjun Xiong[2] Xiyao Wang[5]
Jui-Hsien Wang[2] Richard Zhang[2] Zhe Lin[2] Andrew Owens[1] Yijun Li[2]

[1]Cornell University [2]Adobe [3]Virginia Tech
[4]University of Washington [5]University of Maryland
cf583@cornell.edu

https://cfeng16.github.io/on_the_diffusibility/

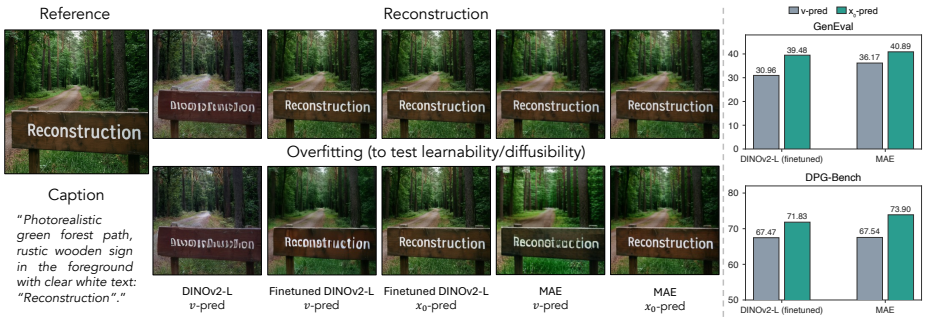
**Abstract.** Representation Autoencoders (RAEs) enable diffusion models to operate in the feature spaces of pretrained visual encoders. However, many off-the-shelf encoders are not optimized for faithful reconstruction and often discard fine-grained visual details. Finetuning these encoders for image reconstruction can recover such details, but we show that it also reduces the effective dimensionality of the resulting representation space. We analyze how this altered geometry affects generation in high-dimensional feature spaces. Under this geometry, standard velocity prediction in flow matching can require the model to fit orthogonal noise directions outside the low-dimensional signal manifold, making optimization inefficient. This motivates the clean data parameterization ($\boldsymbol{x}_0$-prediction), which focuses learning on the underlying signal manifold. Across experiments with multiple reconstruction-finetuned feature sets, we show that $\boldsymbol{x}_0$-prediction consistently improves text-to-image generation performance.

**Keywords:** Text-to-image generation · Representation autoencoder · $\boldsymbol{x}_0$-prediction

## 1 Introduction

Diffusion and flow-matching models [24, 35, 36, 61] have emerged as a standard paradigm for high-fidelity visual generation, including text-to-image synthesis [45,47,49]. To improve their computational efficiency, it is common to perform generation in latent space. Early work used features produced by variational autoencoders (VAEs) [28, 47]. Since these latent spaces are low dimensional and often discard perceptually important information [64, 85], an emerging line of work on *representational autoencoders* (RAEs) [55,64,85] has aimed to perform generation in the feature space of pretrained semantic visual encoders, such as self-supervised features [42] and vision-language features [65].

However, in contrast to the VAEs used in traditional latent diffusion models, there is no guarantee that these semantic encoders capture all of the information that is present in an image, since they are not explicitly trained for reconstruction. As shown in Fig. 1, this can lead to the loss of high-frequency information




(a) Reconstruction and overfitting experiment (b) Metrics

**Fig. 1: Reconstruction and learnability of high-dimensional representation autoencoders.** Finetuning by reconstruction improves detail preservation but makes simple single-image overfitting more difficult under standard velocity prediction in flow matching. Clean data ($\boldsymbol{x}_0$) prediction mitigates this optimization difficulty. (a) Frozen DINOv2-L [42] latents can be fit quickly with $\boldsymbol{v}$-prediction, but have limited reconstruction fidelity. Reconstruction-tuned DINOv2-L and MAE-RAE [22, 85] produce more faithful reconstructions, yet $\boldsymbol{v}$-prediction converges slowly in a simple single-image overfitting test. In contrast, $\boldsymbol{x}_0$-prediction optimizes more efficiently in these strong-reconstruction representation spaces. (b) This optimization advantage transfers to text-to-image generation, where $\boldsymbol{x}_0$-prediction improves GenEval [19] and DPG-Bench [25].

such as text, fine textures, and small objects. A natural way to address this is by finetuning the semantic encoders with an autoencoding loss [7, 52, 63, 80, 83].

While these reconstruction-tuned feature spaces often improve generation quality, we find that training text-to-image diffusion models on them can be surprisingly challenging under standard formulations. As shown in Fig. 1, standard velocity prediction with flow matching [35, 36] becomes harder to optimize in the resulting high-dimensional feature space, converging slowly even in a simple single-image overfitting test.

Prior work addresses this issue by compressing the latents down to a lower dimensional space [7, 63, 83], but this can discard important information. Diffusion is not performed directly in the reconstruction-tuned high-dimensional feature space, leaving the potential of these representations for generation largely unexplored.

In this paper, we aim to address these optimization challenges and train image generation models directly in high-dimensional, reconstruction-tuned feature spaces. We observe that this optimization difficulty is closely tied to the *effective dimensionality* (the minimum number of components required to explain a certain percentage of total variance) of reconstruction-tuned representations. Although the feature dimension is large (e.g., 768 or 1024), the learned features concentrate near a much lower-dimensional subspace, exhibiting a sharp collapse in effective dimensionality. Under this geometry, the standard velocity target decomposes into a manifold-aligned component plus an orthogonal component that is largely uninformative about the clean representation. In high dimensions, this

orthogonal term can dominate, making optimization inefficient and degrading sample quality.

Guided by this analysis, we propose a simple but effective change. Instead of predicting velocity, we take inspiration from Li and He [33] and directly predict the clean representation ($\boldsymbol{x}_0$-prediction) in the high-dimensional feature space. We identify and quantify a reconstruction-induced effective-dimensionality collapse in high-dimensional latent spaces, and analyze why the resulting geometry makes standard $\boldsymbol{v}$-prediction inefficient in this regime. By focusing learning on the low-dimensional signal component and avoiding explicit regression of orthogonal noise directions, $\boldsymbol{x}_0$-prediction yields faster convergence and improved generation. Across two strong-reconstruction tokenizers (fine-tuned DINOv2-L and an MAE-based RAE [22, 85]), $\boldsymbol{x}_0$-prediction consistently improves text-to-image performance on GenEval [19], DPG-Bench [25], and COCO-30k FID [34], matching or surpassing the corresponding frozen semantic baselines.

Our main contributions are threefold:

- We identify and quantify an effective dimensionality collapse in reconstruction-tuned high-dimensional representations, and show that this collapse is correlated to slow convergence of standard velocity prediction.
- We analyze why $\boldsymbol{x}_0$ prediction is better suited to this regime, showing that it mathematically bypasses the orthogonal noise components that hinder standard velocity targets in high dimensions. This allows the diffusion model to efficiently isolate and learn the underlying low-dimensional signal.
- We demonstrate that $\boldsymbol{x}_0$-prediction consistently improves text-to-image generation in strong-reconstruction representation spaces, enabling a finetuned model to effectively surpass the generation quality of its frozen semantic baseline counterpart.

## 2 Related Work

### 2.1 Diffusion and Flow Matching

Diffusion and flow matching models have emerged as an important paradigm for text-to-image generation. Transitioning from early U-Net-based architectures [24, 41, 48] to modern diffusion transformers (DiT) [39, 44], these models have exhibited exceptional scalability and synthesis quality for high-fidelity generation [2, 16, 31, 45, 47, 49, 72]. Diffusion models [14, 24, 41, 58, 59, 61] define a probability path from noise to data through a predefined noise scheduler, which specifies how signal and noise are mixed over time. Training learns to reverse this path via iterative denoising, with the network commonly parameterized to predict noise $\boldsymbol{\epsilon}$ [24], clean data $\boldsymbol{x}_0$ [59], velocity $\boldsymbol{v}$ [50], or the score function $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$ [60]. Flow matching instead defines an explicit deterministic straight-line path between noise and data, and directly trains a neural network to predict the velocity field that transports samples along this path [35, 36]. Under a unified probability path perspective [27, 35], diffusion and flow matching mainly differ in the choice of path and parameterization, which in turn induces different time weightings in the training objective.

### 2.2 Unifying Representation Learning and Flow Matching

Recent work shows that strong representation learning benefits generative modeling, and existing approaches fall into two directions. The first direction aligns diffusion model features with pretrained representations during diffusion training [54, 69, 73, 77]. For example, REPA [77] explicitly supervises the intermediate features of diffusion models to match semantic encoder features [22, 42, 65], thereby injecting structured semantic priors directly into the diffusion model. This alignment improves semantic consistency and generation quality without modifying the latent space or redesigning the overall diffusion architecture.

The second direction redesigns the latent space itself via representation autoencoders [7, 8, 76, 84, 85]. Specifically, AlignTok [7] proposes a three-stage finetuning strategy for pretrained encoders, enabling them to capture high-frequency details for improved reconstruction while preserving semantic structure for better generation. However, it typically operates in lower-dimensional latent spaces, since diffusion models are difficult to train effectively in high-dimensional representation spaces. In contrast, RAE [85] leverages pretrained encoders [22, 42, 65] as frozen feature extractors and demonstrates that diffusion models can operate directly on such high-dimensional semantic latents using a wider diffusion head and modified noise scheduling. However, the use of a frozen encoder limits reconstruction quality.

Our work follows the second line of work. Following AlignTok [7], we extend it to high-dimensional latent spaces, which naturally improve reconstruction quality, and demonstrate that such representations can still be diffused effectively. Moreover, we find that simply adopting RAE-style designs is not sufficient to fully address the diffusibility challenges [32, 57] in high-dimensional latent spaces, motivating a more principled treatment of this regime.

### 2.3 Clean image ($\boldsymbol{x_0}$) prediction

Diffusion models admit several parameterizations of the reverse process, including predicting the reverse-process mean, the added noise $\boldsymbol{\epsilon}$, the clean data $\boldsymbol{x}_0$, or the velocity $\boldsymbol{v}$. Early diffusion probabilistic models learned reverse transition statistics directly [58]. DDPM popularized $\boldsymbol{\epsilon}$-prediction as a simple and effective training target, while also discussing alternatives such as $\boldsymbol{x}_0$-prediction [24]. Clean image prediction is also natural in image restoration, where the goal is to recover the underlying clean signal [12, 40, 75].

Recent work such as JiT [33] revisits $\boldsymbol{x}_0$-prediction and argues that it is particularly beneficial in high-dimensional settings, where clean images lie near a low-dimensional manifold while noise targets do not. Their analysis mainly focuses on pixel space. In contrast, we study $\boldsymbol{x}_0$-prediction in high-dimensional representation spaces, and connect its benefit to the effective-dimensionality collapse induced by reconstruction tuning.

# 3 Method

We first review the preliminaries, then analyze why the standard $\boldsymbol{v}$-prediction objective used in flow matching is difficult to optimize in high-dimensional reconstruction-tuned representation spaces, motivating $\boldsymbol{x}_0$-prediction as a simple alternative.

## 3.1 Preliminaries

**Flow matching.** In standard practice, flow matching [35, 36] uses linear interpolation between clean data $\boldsymbol{x}_0 \sim p_{\text{data}}(\boldsymbol{x})$ and randomly sampled isotropic Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ as input: $\boldsymbol{z}_t = (1-t)\boldsymbol{x}_0 + t\boldsymbol{\epsilon}$, where $t \in [0, 1]$ sampled from predefined time schedule. The velocity $\boldsymbol{v} = \boldsymbol{\epsilon} - \boldsymbol{x}_0$ is employed as the training target to supervise models. The loss function is defined as follows:

$$\mathcal{L} = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}} \|\boldsymbol{v}_\theta(\boldsymbol{z}_t, t) - \boldsymbol{v}\|^2, \tag{1}$$

where $\boldsymbol{v}_\theta$ is a function parameterized by $\theta$. Usually, $\boldsymbol{v}_\theta$ is the direct output of model [35, 36]. Velocity prediction can achieve strong performance for low-dimensional VAE latent [16, 31, 67].

**Representation autoencoder.** RAE [85] uses pretrained representation encoders (e.g., SigLIP2 [65] and DINOv2 [42]) to produce latents for diffusion models [44, 47]. It presents promising performance for class-conditioned generation. Scale-RAE [64] scales further for text-to-image generation by using SigLIP-2 [65].

**$\boldsymbol{x}_0$-prediction.** Recently, JiT [33] shows that standard velocity prediction struggles when data lies on a low-dimensional manifold within a high-dimensional space such as pixel space. Thus, it employs model to predict clean data $\boldsymbol{x}_0$ directly and velocity loss **$\boldsymbol{v}$-loss** by transformation: $\boldsymbol{v}_\theta = \frac{\boldsymbol{z}_t - \boldsymbol{x}_\theta}{t}$, where $\boldsymbol{x}_\theta$ is the model output, target is $\boldsymbol{v} = \frac{\boldsymbol{z}_t - \boldsymbol{x}_0}{t}$. The loss function becomes:

$$\mathcal{L} = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}} \|\boldsymbol{v}_\theta(\boldsymbol{z}_t, t) - \boldsymbol{v}\|^2 = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}} \left\| \frac{\boldsymbol{x}_0 - \boldsymbol{x}_\theta}{t} \right\|^2. \tag{2}$$

## 3.2 Strong-Reconstruction Representation Autoencoder

Representation Autoencoders (RAE) [64, 85] accelerate text-to-image diffusion training by adopting features from pretrained semantic encoders as their latent representation. Yet, these semantic latents typically omit high-frequency information crucial for precise reconstruction. While tolerable for standard generation, this degradation would be severely amplified during fine-grained generation. We show reconstruction and overfitting results in Fig. 2, where generation by using tokenizers from RAE [85] could be bottlenecked by reconstruction in some cases. Consequently, we are motivated to explore generative modeling utilizing strong-reconstruction representation autoencoders, which also inherit semantic information.

**Table 1: Effective dimensionality and reconstruction.** We report the effective dimensionality and reconstruction performance (PSNR) on the ImageNet [13] validation set. We evaluate two main groups of tokenizers/encoders. The first group utilizes checkpoints from RAE [85], where all encoders are kept frozen and the decoders are trained on ImageNet [13] (* indicates we trained the decoder for MAE-B from MAWS [56] on ImageNet using the same architecture). For the second group, we train two decoders separately: one with a frozen DINOv2-L [42] encoder and another with an unfrozen encoder following [7]. $R_\tau$ is defined in Eq. (4), which means minimum number of components required to account for at least $\tau$% of the variance.

| **Tokenizer/Encoder** | Training dataset | Dimensions | | | | Reconstruction (PSNR) |
|---|---|---|---|---|---|---|
| | | Full | $R_{90}$ | $R_{95}$ | $R_{99}$ | |
| DINOv1-B [4] | IN-1k [13] | 768 | 502 | 595 | 701 | — |
| DINOv2-B-RAE [42, 85] | LVD-142M | 768 | 507 | 611 | 726 | 18.85 |
| SigLIP2-B-RAE [65, 85] | WebLI [11] | 768 | 460 | 578 | 715 | 19.10 |
| MAE-B-RAE [22, 85] | IN-1k [13] | 768 | 103 | 252 | 578 | 28.13 |
| MAE-B-IG-3B* [56] | Instagram-3B | 768 | 197 | 348 | 595 | 27.67 |
| DINOv2-L [42] | LVD-142M | 1024 | 672 | 811 | 965 | 17.34 |
| Finetuned DINOv2-L | — | 1024 | 129 | 302 | 726 | 29.12 |

**Effective dimensionality of representations.** A straightforward approach to improving the reconstruction of the representation autoencoder is to finetune it using a reconstruction objective. However, as shown in Fig. 1, we find that it is challenging to perform standard flow matching $v$-prediction in the resulting representation space. Thus, we analyze the singular value spectrum of patch embeddings across different models. For each model, we extract L2-normalized per-patch features for 200k patches randomly sampled from ImageNet [13]. Concretely, for each model, we employ singular value decomposition (SVD) for the concatenated patch embedding matrix $H \in \mathbb{R}^{N\times D}$ to obtain singular values $\sigma_1, \sigma_2, ..., \sigma_D$ in non-increasing order, where $N$ is the number of patches and $D$ is the embedding dimension. The total energy is $E = \sum_{i=1}^{D} \sigma_i^2$, and fraction of energy explained by component $i$ is $f_i = \frac{\sigma_i^2}{E}$. The cumulative energy of the top $k$ components is:

$$C(k) = \sum_{i=1}^{k} f_i = \frac{\sum_{i=1}^{k} \sigma_i^2}{E}. \tag{3}$$

Therefore, we define the effective dimensionality as:

$$R_\tau = \min\{k \mid C(k) \geq \tau\}. \tag{4}$$

This represents the minimum number of components required to account for at least $\tau$% of the variance in the feature embedding. In our analysis, we evaluate $\tau \in \{90, 95, 99\}$. We present the effective dimensionality and reconstruction performance on ImageNet [13] validation set of two main groups of encoders/tokenizers in Tab. 1. We present qualitative reconstruction results in Fig. 2.


**Fig. 2: Reconstruction and overfitting for representation autoencoders.** We show reconstruction and overfitting results for two groups of tokenizers. The first group is RAE [85]. For the second group, we train two decoders separately: one with a frozen DINOv2-L [42] encoder and another with an unfrozen encoder (finetuned) following [7].

First, we reuse the decoder checkpoints from RAE [85], where all encoders are kept strictly frozen and the decoders are trained on ImageNet [13] for reconstruction. Additionally, we train a decoder for MAE-B from MAWS [56] on ImageNet [13] using the same architecture. This MAE-B variant is pretrained on the much larger Instagram-3B dataset. Finally, following prior work [7], we train two separate decoders for DINOv2-L [42] on ImageNet [13] for image reconstruction: one where the DINOv2-L encoder remains strictly frozen, and another where the encoder is unfrozen and finetuned during training.

As shown in Tab. 1, the effective dimensionality for the DINOv2 [42] and SigLIP2 [65] models remains high compared with the full feature dimension, which could partially explain their promising generation performance. DINOv1 [4], despite only being pretrained on ImageNet [13], also maintains a high effective dimensionality. In contrast, MAE [22], whose pretraining objective is reconstruction, exhibits a much lower effective dimensionality, even when trained on a large-scale dataset [56]. Similarly, the finetuned DINOv2-L has a significantly lower effective dimensionality than the original frozen DINOv2-L [42]. These results indicate that when an encoder is trained for reconstruction, the ratio of effective dimensionality to the full feature dimension shrinks significantly.

We further visualize this compression via normalized singular value decay and cumulative explained variance in Fig. 3. As observed, the normalized singular values of MAE [22] decay rapidly, a bottleneck that persists even when pretrained on the massive Instagram-3B dataset. Similarly, when DINOv2-L is finetuned via reconstruction, a much smaller number of principal components is required to capture the same amount of variance compared to the original model. A potential reason for this is that to perform reconstruction, the encoder must capture a significant amount of high-frequency information from the original image. Since

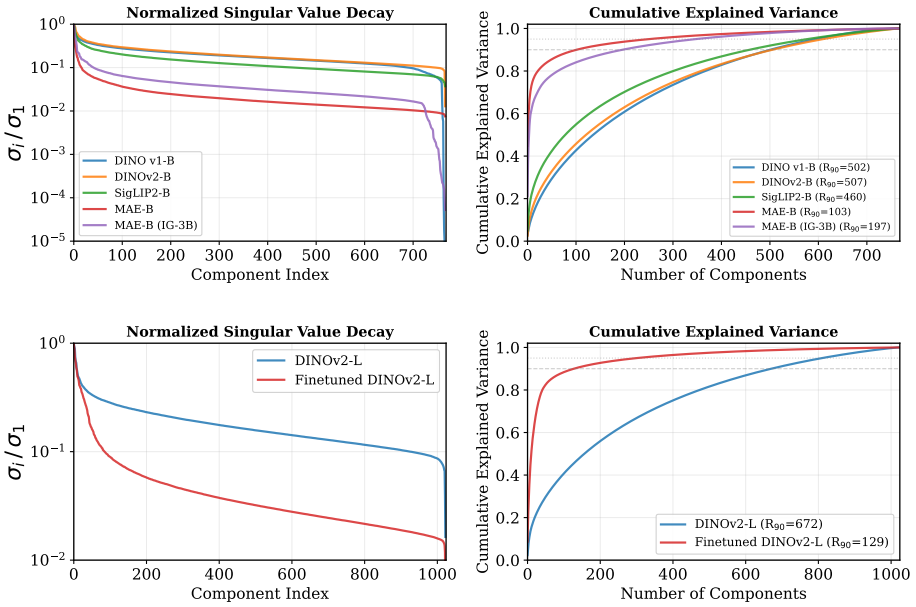



**Fig. 3: Effective dimensionality of vision encoder representations.** We analyze normalized singular value decay (log-scale y-axis, normalized by $\sigma_1$) and cumulative explained variance for the visual encoders listed in Tab. 1 and observe that reconstruction-based objectives significantly reduce effective dimensionality. For instance, the normalized singular values of MAE [22] decay rapidly, even when pretrained on the massive Instagram-3B dataset. Similarly, when DINOv2-L [42] is finetuned via reconstruction, a much smaller number of principal components is required to capture the same amount of variance compared to the original model.

natural images are suggested to reside on a low-dimensional manifold [3, 6, 33], the features are forced to mimic this lower-dimensional distribution.

Based on Tab. 1 and Fig. 3, our hypothesis is that high-dimensional encoders that are trained or finetuned for image reconstruction objective will concentrate near a low-dimensional subspace. Assume the observed high-dimensional feature $\boldsymbol{x}_0 \in \mathbb{R}^h$, extracted by a visual encoder, lies within a lower-dimensional subspace. Specifically, $\boldsymbol{x}_0$ is generated from a true latent variable $\boldsymbol{c}_0 \in \mathbb{R}^l$ via the linear mapping $\boldsymbol{x}_0 = Q\boldsymbol{c}_0$, where the columns of $Q \in \mathbb{R}^{h \times l}$ form an orthonormal basis for the subspace (i.e., $Q^TQ = I_l$). Input $\boldsymbol{z}_t^h = (1-t)\boldsymbol{x}_0 + t\boldsymbol{\epsilon}_h$ could be decomposed into $\boldsymbol{z}_t^h = Q((1-t)\boldsymbol{c}_0 + t\boldsymbol{\epsilon}_l) + t\boldsymbol{\epsilon}_\perp$, where $\boldsymbol{\epsilon}_\perp = \boldsymbol{\epsilon}_h - Q\boldsymbol{\epsilon}_l = (I - QQ^T)\boldsymbol{\epsilon}_h$. $\boldsymbol{z}_t^l = (1-t)\boldsymbol{c}_0 + t\boldsymbol{\epsilon}_l$ can be viewed as manifold component. $\boldsymbol{\epsilon}_\perp$ is the normal component and we have $(I - QQ^T)\boldsymbol{z}_t^h = t\boldsymbol{\epsilon}_\perp$. The model that predicts velocity in high-dimensional space is defined as: $\boldsymbol{v}_\theta^h(\boldsymbol{z}_t^h, t) = \mathbb{E}[\boldsymbol{\epsilon_h} - \boldsymbol{x}_0 | \boldsymbol{z}_t^h] = \mathbb{E}[Q(\boldsymbol{\epsilon}_l - \boldsymbol{c}_0) + \boldsymbol{\epsilon}_\perp | \boldsymbol{z}_t^h]$.

Therefore, we can obtain

$$\begin{aligned} \boldsymbol{v}_\theta^h(\boldsymbol{z}_t^h, t) &= \mathbb{E}[Q(\boldsymbol{\epsilon}_l - \boldsymbol{c}_0)|\boldsymbol{z}_t^h] + \mathbb{E}[\boldsymbol{\epsilon}_\perp|\boldsymbol{z}_t^h] \\ &= Q\boldsymbol{v}_\theta^l(Q^\top \boldsymbol{z}_t^h, t) + \frac{1}{t}(I - QQ^\top)\boldsymbol{z}_t^h. \end{aligned} \tag{5}$$

When the dimension of $\boldsymbol{x}_0$ is much higher than the dimension of $\boldsymbol{c}_0$ ($h \gg l$), the model needs to allocate substantial capacity for $\frac{1}{t}(I - QQ^\top)\boldsymbol{z}_t^h$. As discussed in [83], one solution is to train an adapter to reduce dimensionality. Instead, we focus on predicting high-dimensional features in this paper, as retaining dimensionality might be able to retain both rich semantic information and reconstruction capability. The second term in Eq. (5) is from $\boldsymbol{\epsilon}_\perp$, which is introduced by $\boldsymbol{\epsilon}_h$ in the vanilla velocity prediction training target. To avoid this, we employ the model to predict $\boldsymbol{x}_0$ directly:

$$\boldsymbol{x}_\theta^h(\boldsymbol{z}_t^h, t) = \mathbb{E}[\boldsymbol{x}_0|\boldsymbol{z}_t^h] = \mathbb{E}[Q\boldsymbol{c}_0|Q\boldsymbol{z}_t^l + t\boldsymbol{\epsilon}_\perp] = \mathbb{E}[Q\boldsymbol{c}_0|Q\boldsymbol{z}_t^l, t\boldsymbol{\epsilon}_\perp]. \tag{6}$$

Since $Q\boldsymbol{c}_0$ is independent of $t\epsilon_\perp$, Eq. (6) could be simplified to:

$$\boldsymbol{x}_\theta^h(\boldsymbol{z}_t^h, t) = \mathbb{E}[Q\boldsymbol{c}_0|Q\boldsymbol{z}_t^l] = Q\mathbb{E}[\boldsymbol{c}_0|Q\boldsymbol{z}_t^l] = Q\boldsymbol{x}_\theta^l(\boldsymbol{z}_t^l, t). \tag{7}$$

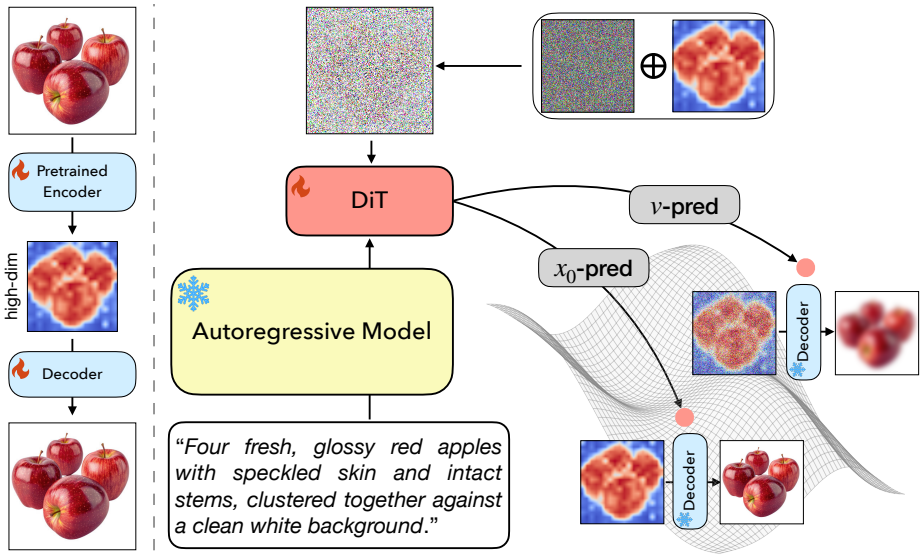
Eq. (7) shows that directly modeling $\boldsymbol{x}_0 \in \mathbb{R}^h$ in high-dimensional space could predict $\boldsymbol{c}_0 \in \mathbb{R}^l$ more efficiently, especially when $h \gg l$.

**$\boldsymbol{x}_0$-prediction for representation latent.** Based on our analysis above, we prefer to do $\boldsymbol{x}_0$-prediction for strong-reconstruction representation autoencoders. Specifically, given text-image pair $(\boldsymbol{I}_i, \boldsymbol{T}_i)$, extracted text embeddings act as the condition $\boldsymbol{y}_i$ for diffusion transformer $\boldsymbol{x}_\theta$. Feature $\boldsymbol{h}_i$ produced by representation autoencoder $g$: $\boldsymbol{h}_i = g(\boldsymbol{I}_i)$ is used as latent for diffusion. Given input $\boldsymbol{h}_{i,t} = (1-t)\boldsymbol{h}_{i,0} + t\boldsymbol{\epsilon}$, DiT is trained to denoise noisy image representations. The loss function is defined as follows:

$$\mathcal{L}_{\text{Diff}} = \mathbb{E}_{t, \boldsymbol{h}_{i,0}, \boldsymbol{\epsilon}} \left\| \frac{\boldsymbol{h}_{i,0} - \boldsymbol{x}_\theta(\boldsymbol{h}_{i,t}, t, \boldsymbol{y}_i)}{t} \right\|^2, \tag{8}$$

where $\boldsymbol{h}_{i,0}$ means the clean feature $\boldsymbol{h}_{i,0} \coloneqq \boldsymbol{h}_i$. In practice, we clamp $t$ to a minimum of 0.05, following [33], to prevent numerical instability.

**Text-to-image generation architecture.** As shown in Fig. 4, we mainly focus on using representation autoencoder with strong reconstruction performance for text-to-image generation. The high-dimensional visual features produced from it are used as latents for diffusion. For the generation framework, we adopt an architecture similar to MetaQuery [43] and BLIP3-o [9], where we use a frozen pretrained vision language model (VLM) for the autoregressive model. Learnable query tokens are employed to extract text information as conditioning for randomly initialized diffusion transformer (DiT).




(a) Tokenizer (b) Text-to-image generation framework

**Fig. 4: Method.** (a) We unfreeze pretrained visual encoder when training the decoder for reconstruction. Then we use high-dimensional representation from finetuned encoder as latent. (b) We use autoregressive model and query tokens to extract text embedding as conditioning for DiT. Due to the low-dimensional manifold property, we employ $\boldsymbol{x}_0$ prediction to train text-to-image DiT more efficiently.

## 4 Experiments

### 4.1 Implementation Details

We use Qwen3-VL-2B-Instruct [1] as our (frozen) autoregressive model and Lumina-Next [86] with 1B size as our DiT. The DiT consists of 24 layers and 24 attention heads per layer, with a hidden dimension of 1536. The number of query tokens is set to be 64. We use AdamW [37] with the learning rate of $10^{-4}$. Global batch size is 1024. We use a 34M subset of the public pretraining dataset used in BLIP-3o [9]. All text-to-image generation models are trained for 90k steps. The whole size of the public BLIP-3o [9] pretraining dataset is 39.3M. It combines mostly webdata like CC12M [5], SA-1B [29], and JourneyDB [62], and recaptions many images. VLM is frozen during training. DiT and learnable query tokens are trained from scratch. Following prior work [64, 85], we adopt the uniform time schedule with a shift for DiT training. Concretely, given base $t_n$, the shifted $t_m$ is defined as follows:

$$t_m = \frac{\alpha t_n}{1 + (\alpha - 1)t_n}, \tag{9}$$

where $\alpha = \sqrt{\frac{m}{n}}$, $m$ = number of image tokens $\times$ token dimension. We use $n = 4096$ following RAE [64, 85]. We use two tokenizers: 1) frozen MAE pretrained

**Table 2: Evaluation of finetuned DINOv2-L.** We compare the text-to-image generation performance of the original DINOv2-L [42] against our finetuned variants on three benchmarks: GenEval [19], DPG-Bench [25], and COCO-30k FID [34]. Following RAE [85], we use standard $\boldsymbol{v}$-prediction for DINOv2-L. For the fine-tuned models, we evaluate $\boldsymbol{v}$ and $\boldsymbol{x}_0$ prediction targets. The FT-DINOv2-L-low-dim variant utilizes a trained adapter to down-project high-dimensional features, performing standard flow matching in this lower-dimensional space. Best are highlighted in **bold**.

| **Tokenizer** | Dimension | Pred. type | GenEval↑ | DPG-Bench↑ | COCO-30k FID↓ |
|---|---|---|---|---|---|
| FT-DINOv2-L-low-dim | 32 | $\boldsymbol{v}$ | 40.99 | 73.04 | 17.64 |
| DINOv2-L | 1024 | $\boldsymbol{v}$ | 37.63 | 70.21 | 18.34 |
| FT-DINOv2-L | 1024 | $\boldsymbol{v}$ | 30.96 | 67.47 | 29.97 |
| FT-DINOv2-L | 1024 | $\boldsymbol{x}_0$ | **39.48** | **71.83** | **16.80** |

on ImageNet [13] paired with a ViT-XL [15] decoder from RAE [85], which is also trained on IN-1k [13], and 2) we unfreeze DINOv2-L [42] during reconstruction training on IN-1k [13], which is paired with a decoder with the similar architecture as VAE used in stable diffusion [47] following [7].

When we finetune encoder of DINOv2-L [42], we also leverage semantic preservation loss besides reconstruction loss to prevent collapse following [7]. Specifically, for each image $I_i$, feature $\boldsymbol{h}_i$ is extracted by original DINOv2-L [42] $g$ for semantic preservation. The final loss to finetune encoder DINOv2-L $g\prime$ is defined as follows:

$$\mathcal{L}_{FT} = \|g(I) - g\prime(I)\|^2 + \mathcal{L}_{\text{recon}}, \tag{10}$$

where $\mathcal{L}_{\text{recon}}$ consists of pixel-level $L1$, perceptual, and adversarial losses. All experiments are conducted at a resolution of $256 \times 256$. The model training is using either 64 NVIDIA A100 or 32 H200 GPUs.

### 4.2 Evaluation

We evaluate FID [23] on COCO-30k [34] for image quality. Following prior work [9, 64], we also use two widely adopted metrics: the GenEval score [19] and the DPG-Bench score [25] for evaluation of text-image alignment. We employ a 100-step Euler sampler for generation.

### 4.3 Text-to-image Generation Comparison

**$\boldsymbol{x}_0$-prediction better than $\boldsymbol{v}$-prediction for strong-reconstruction representation autoencoders.** To evaluate the finetuned DINOv2-L tokenizer, we compare four configurations: (1) the original DINOv2-L [42] tokenizer using $\boldsymbol{v}$-prediction, following RAE [85]; (2) the finetuned DINOv2-L tokenizer using standard $\boldsymbol{v}$-prediction; (3) the finetuned DINOv2-L tokenizer using $\boldsymbol{x}_0$-prediction; and (4) a low-dimensional variant. For this fourth setup, we train

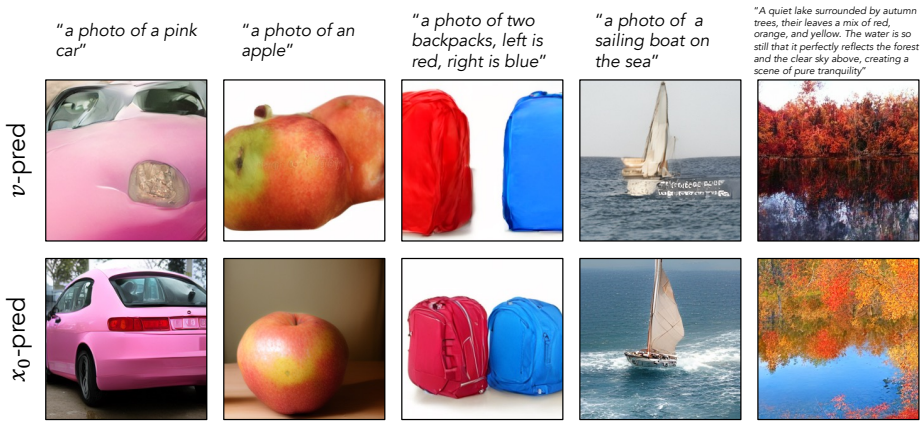



**Fig. 5: Qualitative comparison of $\boldsymbol{x}_0$ and $\boldsymbol{v}$ predictions for finetuned DINOv2-L.** We present text-to-image generation examples comparing the $\boldsymbol{x}_0$ and $\boldsymbol{v}$ prediction targets using the finetuned DINOv2-L tokenizer. The results demonstrate that $\boldsymbol{x}_0$ prediction yields better visual quality.

an adapter to down-project the high-dimensional (1024-dim) features into a 32-dimensional space, perform standard flow matching ($\boldsymbol{v}$-prediction) in this compressed space, and train the decoder specifically on these 32-dimensional features rather than the full 1024 dimensions following [7]. We present result in Tab. 2. When we finetune DINOv2-L [42] by reconstruction objective and perform standard $\boldsymbol{v}$-prediction, we observe performance deterioration on three benchmarks: GenEval [19], DPG-Bench [25], and COCO-30k FID [34] as suggested by our analysis in our analysis in Sec. 3.2. Even though unfreezing encoder should let encoder capture more information about image, diffusion model seems to struggle to denoise finetuned features correctly. This suggests that finetuning decoder by reconstruction makes the diffusibility of representation space decrease for standard flow matching $\boldsymbol{v}$-prediction.

However, switching to $\boldsymbol{x}_0$-prediction boosts performance by a relatively large margin and surpasses original DINOv2 [42] with $\boldsymbol{v}$-prediction. This solidifies our analysis in Sec. 3.2, when effective dimensionality of visual representation is low (e.g., finetuned DINOv2 in Tab. 1), using $\boldsymbol{x}_0$-prediction leads to faster convergence for generative models. We present some qualitative results in Fig. 5 to compare $\boldsymbol{x}_0$-prediction and $\boldsymbol{v}$-prediction for finetuned DINOv2 tokenizer. Fig. 5 shows that $\boldsymbol{x}_0$-prediction leads to better overall visual quality, such as object structure and text alignment. For example, given text prompt "`a photo of a pink car`", sampled image by $\boldsymbol{v}$-prediction struggles to capture the global geometry of the object, resulting in amorphous, blob-like structures. In contrast, $\boldsymbol{x}_0$-prediction maintains strong structural integrity and generates geometrically coherent objects.

Additionally, we present results for tokenizer MAE-RAE [85] in Tab. 3, where encoder MAE is pretrained on IN-1k [13] and kept frozen during training of

**Table 3: Evaluation of MAE-RAE.** We compare the text-to-image generation performance of $\boldsymbol{x}_0$ and $\boldsymbol{v}$ predictions for MAE-RAE [85] across three benchmarks: GenEval [19], DPG-Bench [25], and COCO-30k FID [34]. Best are highlighted in **bold**.

| **Tokenizer** | Dimension | Prediction type | GenEval↑ | DPG-Bench↑ | COCO-30k FID↓ |
|---|---|---|---|---|---|
| MAE-B-RAE | 768 | $\boldsymbol{v}$ | 36.17 | 67.54 | 22.20 |
| MAE-B-RAE | 768 | $\boldsymbol{x}_0$ | **40.89** | **73.90** | **17.24** |

decoder. The pretraining objective for encoder MAE is reconstruction, it has low effective dimensionality compared with its own full feature dimension as suggested in Tab. 1. As shown in Tab. 3, $\boldsymbol{x}_0$-prediction improves text-to-image generation performance on three benchmarks in comparison with $\boldsymbol{v}$-prediction. The qualitative results in Fig. 6 show that sampled images by $\boldsymbol{x}_0$-prediction have better object structure and richer details. These results suggest that when high-dimensional visual feature/latent resides on a low-dimensional manifold, using $\boldsymbol{x}_0$-prediction can make diffusion models on this high-dimensional representation space converge faster and yield better sampled results.

**Reconstruction still matters for generation.** The semantic space formed by pretrained visual encoders like DINOv2 [42] can lift a lot for generation. Semantics help models converge faster [64, 85]. To investigate whether reconstruction can help generation, we compare tokenizer of original DINOv2-L [42] with tokenizer of finetuned DINOv2-L. For original DINOv2 [42], we use $\boldsymbol{v}$-prediction. $\boldsymbol{x}_0$-prediction is employed for finetuned DINOv2-L since previous section suggests that $\boldsymbol{v}$-prediction struggles to diffuse efficiently for finetuned DINOv2-L. Quantitative and qualitative results are presented in Tab. 2 and Fig. 7, respectively. Tab. 2 shows that finetuned DINOv2-L achieves better results on three benchmarks: GenEval [19], DPG-Bench [25], and COCO-30k FID [34], which indicates that tuning encoder by reconstruction can help generation. As presented in Fig. 7, diffusion model based on original DINOv2-L [42] fails to capture correct color information (e.g., `"a photo of a purple backpack"`) or the object integrity when texture is complicated (e.g., `"a photo of a bicycle"`). Though Scale-RAE [64] presents that scaling the size of training dataset can alleviate this problem, our experiments show the certain limitation of using features from pretrained semantic encoders without reconstruction tuning. We will explore more regarding building visual encoder for generative and unified models (understanding and generation) in the future.

**High-dimensional diffusion is inherently harder than the low-dim.** Following [7], we train another variant for finetuned DINOv2-L. Concretely, we set a two-layer MLP after encoder to down project high-dimensional feature (1024-dim) to be low-dimensional (32-dim), the subsequent decoder is also based on 32-dim features to do pixel decoding. We use $\boldsymbol{v}$-prediction following standard practice for this low-dimensional latent space. We present results in Tab. 2. While the low-dimensional bottleneck (FT-DINOv2-L-low-dim) achieves

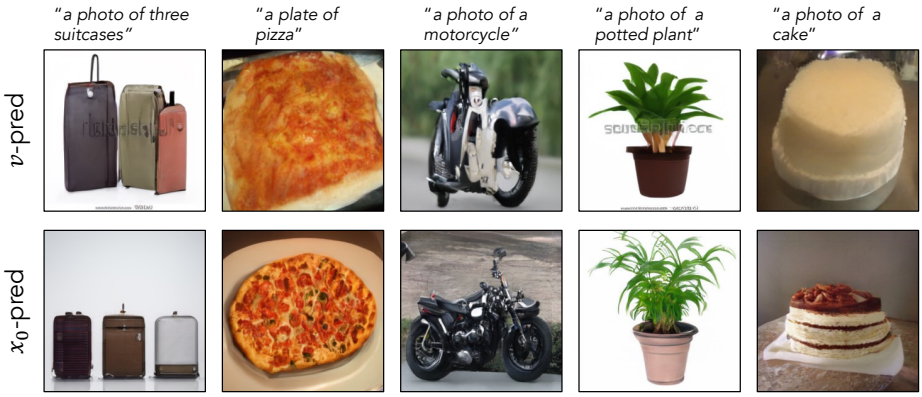



**Fig. 6: Qualitative comparison of $\boldsymbol{x}_0$ and $\boldsymbol{v}$ predictions for MAE.** We present generated examples comparing the $\boldsymbol{x}_0$ and $\boldsymbol{v}$ prediction targets using the MAE-RAE tokenizer [85]. The results demonstrate that $\boldsymbol{x}_0$ prediction yields better visual quality.

stronger text-alignment metrics (GenEval [19] and DPG-Bench [25]), this compression comes at the cost of image quality, yielding a worse FID score (17.64). In contrast, our unbottlenecked high-dimensional approach with $\boldsymbol{x}_0$-prediction avoids this dimensionality reduction, achieving better visual fidelity (FID 16.80) while maintaining competitive semantic alignment. This demonstrates that $\boldsymbol{x}_0$-prediction successfully mitigates the optimization difficulty of high-dimensional spaces, preserving the rich, fine-grained details of strong-reconstruction encoders.

**Table 4: Scaling to a larger dataset.** We train our method on a larger-scale dataset and perform supervised fine-tuning (SFT) at resolutions of 256 and 512. The asterisk (*) denotes our own SFT implementation.

| **Method** | DiT size | Training dataset | Latent dim. | Res. | GenEval↑ | DPG-Bench↑ |
|---|---|---|---|---|---|---|
| Ours | 1B | 90M | 1024 | 256 | 43.84 | 76.15 |
| Ours | 1B | 90M + SFT 60k | 1024 | 256 | 78.62 | 80.13 |
| Ours (512) | 1B | 90M (256) + SFT 60k (512) | 1024 | 512 | **79.69** | **81.15** |
| Scale-RAE* [64] | 2.4B | 64M + SFT 60k | 1152 | 224 | 77.43 | 78.47 |

**Scaling to a Larger Dataset.** We further train a diffusion model with $\boldsymbol{x}_0$-prediction using the finetuned DINOv2 tokenizer on a larger internal 90M dataset, while keeping the remaining hyperparameters unchanged. Following Scale-RAE [64], we then finetune the pretrained model on BLIP3o-60k [9] at 256 resolution, and additionally finetune a high-resolution variant at 512 resolution. We include Scale-RAE [64] as a reference. Results are shown in Tab. 4. These results suggest that our method has potential scalability and could achieve competitive performance for text-to-image generation.


**Fig. 7: Qualitative comparison of DINOv2-L and finetuned DINOv2-L.** We present text-to-image generation examples comparing the tokenizers of the original DINOv2-L [42] and our finetuned DINOv2-L variant. The results indicate that the fine-tuned encoder leads to better structural integrity and color alignment.

## 5 Conclusion

In this paper, we investigate text-to-image generation in strong-reconstruction, high-dimensional representation spaces. We show that after tuning the encoder by reconstruction objective, the standard velocity prediction becomes difficult to optimize in the resulting high-dimensional representation space. Empirically, reconstruction-tuned representation autoencoders exhibit effective-dimensionality collapse. We provide an analysis explaining this failure mode and propose a simple remedy: training the diffusion model to predict the clean representation $\boldsymbol{x}_0$ directly. Across two strong-reconstruction tokenizers, $\boldsymbol{x}_0$-prediction consistently improves text-to-image generation quality.

**Acknowledgements.** We thank Sicheng Mo, Xichen Pan, Eli Shechtman, Krishna Kumar Singh, Phillip Isola, and Nicolas Dufour for their valuable discussions and help. This work was supported in part by NSF CAREER #2339071.

## References


1. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025)
2. Cai, H., Cao, S., Du, R., Gao, P., Hoi, S., Hou, Z., Huang, S., Jiang, D., Jin, X., Li, L., et al.: Z-image: An efficient image generation foundation model with single-stream diffusion transformer. arXiv preprint arXiv:2511.22699 (2025)
3. Carlsson, G.: Topology and data. Bulletin of the American Mathematical Society **46**(2), 255–308 (2009)
4. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9650–9660 (2021)
5. Changpinyo, S., Sharma, P., Ding, N., Soricut, R.: Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3558–3568 (2021)
6. Chapelle, O., Schölkopf, B., Zien, A. (eds.): Semi-Supervised Learning. MIT Press, Cambridge, MA, USA (2006)
7. Chen, B., Bi, S., Tan, H., Zhang, H., Zhang, T., Li, Z., Xiong, Y., Zhang, J., Zhang, K.: Aligning visual foundation encoders to tokenizers for diffusion models. arXiv preprint arXiv:2509.25162 (2025)
8. Chen, H., Han, Y., Chen, F., Li, X., Wang, Y., Wang, J., Wang, Z., Liu, Z., Zou, D., Raj, B.: Masked autoencoders are effective tokenizers for diffusion models. In: Forty-second International Conference on Machine Learning (2025)
9. Chen, J., Xu, Z., Pan, X., Hu, Y., Qin, C., Goldstein, T., Huang, L., Zhou, T., Xie, S., Savarese, S., et al.: Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. arXiv preprint arXiv:2505.09568 (2025)
10. Chen, L., Li, J., Dong, X., Zhang, P., Zang, Y., Chen, Z., Duan, H., Wang, J., Qiao, Y., Lin, D., et al.: Are we on the right way for evaluating large vision-language models? arXiv preprint arXiv:2403.20330 (2024)
11. Chen, X., Wang, X., Changpinyo, S., Piergiovanni, A.J., Padlewski, P., Salz, D., Goodman, S., Grycner, A., Mustafa, B., Beyer, L., et al.: Pali: A jointly-scaled multilingual language-image model. arXiv preprint arXiv:2209.06794 (2022)
12. Delbracio, M., Milanfar, P.: Inversion by direct iteration: An alternative to denoising diffusion for image restoration. arXiv preprint arXiv:2303.11435 (2023)
13. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
14. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in neural information processing systems **34**, 8780–8794 (2021)
15. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
16. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-first international conference on machine learning (2024)

17. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12873–12883 (2021)
18. Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576 (2015)
19. Ghosh, D., Hajishirzi, H., Schmidt, L.: Geneval: An object-focused framework for evaluating text-to-image alignment. Advances in Neural Information Processing Systems **36**, 52132–52152 (2023)
20. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020)
21. Hansen-Estruch, P., Yan, D., Chung, C.Y., Zohar, O., Wang, J., Hou, T., Xu, T., Vishwanath, S., Vajda, P., Chen, X.: Learnings from scaling visual tokenizers for reconstruction and generation. arXiv preprint arXiv:2501.09755 (2025)
22. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16000–16009 (2022)
23. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
24. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
25. Hu, X., Wang, R., Fang, Y., Fu, B., Cheng, P., Yu, G.: Ella: Equip diffusion models with llm for enhanced semantic alignment. arXiv preprint arXiv:2403.05135 (2024)
26. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European conference on computer vision. pp. 694–711. Springer (2016)
27. Kingma, D., Gao, R.: Understanding diffusion objectives as the elbo with simple data augmentation. Advances in Neural Information Processing Systems **36**, 65484–65516 (2023)
28. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. CoRR **abs/1312.6114** (2013), `https://api.semanticscholar.org/CorpusID:216078090`
29. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4015–4026 (2023)
30. Kouzelis, T., Kakogeorgiou, I., Gidaris, S., Komodakis, N.: Eq-vae: Equivariance regularized latent space for improved generative image modeling. arXiv preprint arXiv:2502.09509 (2025)
31. Labs, B.F.: Flux. `https://github.com/black-forest-labs/flux` (2024)
32. Lai, B., Wang, X., Rambhatla, S., Rehg, J.M., Kira, Z., Girdhar, R., Misra, I.: Toward diffusible high-dimensional latent spaces: A frequency perspective. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 43450–43460 (2026)
33. Li, T., He, K.: Back to basics: Let denoising generative models denoise. arXiv preprint arXiv:2511.13720 (2025)
34. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
35. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022)

36. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. arXiv preprint arXiv:2209.03003 (2022)
37. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
38. Lu, P., Bansal, H., Xia, T., Liu, J., Li, C., Hajishirzi, H., Cheng, H., Chang, K.W., Galley, M., Gao, J.: Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. arXiv preprint arXiv:2310.02255 (2023)
39. Ma, N., Goldstein, M., Albergo, M.S., Boffi, N.M., Vanden-Eijnden, E., Xie, S.: Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In: European Conference on Computer Vision. pp. 23–40. Springer (2024)
40. Milanfar, P., Delbracio, M.: Denoising: a powerful building block for imaging, inverse problems and machine learning. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **383**(2299) (2025)
41. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: International conference on machine learning. pp. 8162–8171. PMLR (2021)
42. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
43. Pan, X., Shukla, S.N., Singh, A., Zhao, Z., Mishra, S.K., Wang, J., Xu, Z., Chen, J., Li, K., Juefei-Xu, F., et al.: Transfer between modalities with metaqueries. arXiv preprint arXiv:2504.06256 (2025)
44. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4195–4205 (2023)
45. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 **1**(2), 3 (2022)
46. Ridnik, T., Ben-Baruch, E., Noy, A., Zelnik-Manor, L.: Imagenet-21k pretraining for the masses. In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1) (2021), https://openreview.net/forum?id=Zkj_VcZ6ol
47. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
48. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
49. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. Advances in neural information processing systems **35**, 36479–36494 (2022)
50. Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. arXiv preprint arXiv:2202.00512 (2022)
51. Shazeer, N.: Glu variants improve transformer. arXiv preprint arXiv:2002.05202 (2020)
52. Shi, M., Wang, H., Zheng, W., Yuan, Z., Wu, X., Wang, X., Wan, P., Zhou, J., Lu, J.: Latent diffusion model without variational autoencoder. arXiv preprint arXiv:2510.15301 (2025)
53. Silvestri, G., Ambrogioni, L.: Covae: Consistency training of variational autoencoders. arXiv preprint arXiv:2507.09103 (2025)

54. Singh, J., Leng, X., Wu, Z., Zheng, L., Zhang, R., Shechtman, E., Xie, S.: What matters for representation alignment: Global information or spatial structure? arXiv preprint arXiv:2512.10794 (2025)
55. Singh, J., Zheng, B., Wu, Z., Zhang, R., Shechtman, E., Xie, S.: Improved baselines with representation autoencoders. arXiv preprint arXiv:2605.18324 (2026)
56. Singh, M., Duval, Q., Alwala, K.V., Fan, H., Aggarwal, V., Adcock, A., Joulin, A., Dollár, P., Feichtenhofer, C., Girshick, R., et al.: The effectiveness of mae pre-pretraining for billion-scale pretraining. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5484–5494 (2023)
57. Skorokhodov, I., Girish, S., Hu, B., Menapace, W., Li, Y., Abdal, R., Tulyakov, S., Siarohin, A.: Improving the diffusability of autoencoders. arXiv preprint arXiv:2502.14831 (2025)
58. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International conference on machine learning. pp. 2256–2265. pmlr (2015)
59. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
60. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. Advances in neural information processing systems **32** (2019)
61. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)
62. Sun, K., Pan, J., Ge, Y., Li, H., Duan, H., Wu, X., Zhang, R., Zhou, A., Qin, Z., Wang, Y., et al.: Journeydb: A benchmark for generative image understanding. Advances in neural information processing systems **36**, 49659–49678 (2023)
63. Tang, H., Xie, C., Bao, X., Weng, T., Li, P., Zheng, Y., Wang, L.: Unilip: Adapting clip for unified multimodal understanding, generation and editing. arXiv preprint arXiv:2507.23278 (2025)
64. Tong, S., Zheng, B., Wang, Z., Tang, B., Ma, N., Brown, E., Yang, J., Fergus, R., LeCun, Y., Xie, S.: Scaling text-to-image diffusion transformers with representation autoencoders. arXiv preprint arXiv:2601.16208 (2026)
65. Tschannen, M., Gritsenko, A., Wang, X., Naeem, M.F., Alabdulmohsin, I., Parthasarathy, N., Evans, T., Beyer, L., Xia, Y., Mustafa, B., et al.: Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. arXiv preprint arXiv:2502.14786 (2025)
66. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. Advances in neural information processing systems **30** (2017)
67. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)
68. Wang, K., Pan, J., Shi, W., Lu, Z., Ren, H., Zhou, A., Zhan, M., Li, H.: Measuring multimodal mathematical reasoning with math-vision dataset. In: The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track (2024), https://openreview.net/forum?id=QWTCcxMpPA
69. Wang, Z., Zhao, W., Zhou, Y., Li, Z., Liang, Z., Shi, M., Zhao, X., Zhou, P., Zhang, K., Wang, Z., et al.: Repa works until it doesn't: Early-stopped, holistic alignment supercharges diffusion training. arXiv preprint arXiv:2505.16792 (2025)
70. Wang, Z., Xia, M., He, L., Chen, H., Liu, Y., Zhu, R., Liang, K., Wu, X., Liu, H., Malladi, S., Chevalier, A., Arora, S., Chen, D.: Charxiv: Charting gaps in realistic chart understanding in multimodal llms. arXiv preprint arXiv:2406.18521 (2024)

71. Wen, X., Zhao, B., Elezi, I., Deng, J., Qi, X.: " principal components" enable a new language of images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16641–16651 (2025)
72. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., Yin, S.m., Bai, S., Xu, X., Chen, Y., et al.: Qwen-image technical report. arXiv preprint arXiv:2508.02324 (2025)
73. Wu, G., Zhang, S., Shi, R., Gao, S., Chen, Z., Wang, L., Chen, Z., Gao, H., Tang, Y., Yang, J., et al.: Representation entanglement for generation: Training diffusion transformers is much easier than you think. arXiv preprint arXiv:2507.01467 (2025)
74. Wu, P., Zhu, K., Liu, Y., Tang, L., Yang, J., Peng, Y., Zhai, W., Cao, Y., Zha, Z.J.: Towards sequence modeling alignment between tokenizer and autoregressive model. arXiv preprint arXiv:2506.05289 (2025)
75. Xie, Y., Yuan, M., Dong, B., Li, Q.: Diffusion model for generative image denoising. arXiv preprint arXiv:2302.02398 (2023)
76. Yao, J., Yang, B., Wang, X.: Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 15703–15712 (2025)
77. Yu, S., Kwak, S., Jang, H., Jeong, J., Huang, J., Shin, J., Xie, S.: Representation alignment for generation: Training diffusion transformers is easier than you think. arXiv preprint arXiv:2410.06940 (2024)
78. Yue, X., Ni, Y., Zhang, K., Zheng, T., Liu, R., Zhang, G., Stevens, S., Jiang, D., Ren, W., Sun, Y., et al.: Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9556–9567 (2024)
79. Yue, X., Zheng, T., Ni, Y., Wang, Y., Zhang, K., Tong, S., Sun, Y., Yu, B., Zhang, G., Sun, H., et al.: Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 15134–15186 (2025)
80. Yue, Z., Zhang, H., Zeng, X., Chen, B., Wang, C., Zhuang, S., Dong, L., Du, K., Wang, Y., Wang, L., et al.: Uniflow: A unified pixel flow tokenizer for visual understanding and generation. arXiv preprint arXiv:2510.10575 (2025)
81. Zhang, R., Jiang, D., Zhang, Y., Lin, H., Guo, Z., Qiu, P., Zhou, A., Lu, P., Chang, K.W., Qiao, Y., et al.: Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In: European Conference on Computer Vision. pp. 169–186. Springer (2024)
82. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
83. Zhang, S., Zhang, H., Zhang, Z., Ge, C., Xue, S., Liu, S., Ren, M., Kim, S.Y., Zhou, Y., Liu, Q., et al.: Both semantics and reconstruction matter: Making representation encoders ready for text-to-image generation and editing. arXiv preprint arXiv:2512.17909 (2025)
84. Zheng, A., Wen, X., Zhang, X., Ma, C., Wang, T., Yu, G., Zhang, X., Qi, X.: Vision foundation models as effective visual tokenizers for autoregressive image generation. arXiv preprint arXiv:2507.08441 (2025)
85. Zheng, B., Ma, N., Tong, S., Xie, S.: Diffusion transformers with representation autoencoders. arXiv preprint arXiv:2510.11690 (2025)
86. Zhuo, L., Du, R., Xiao, H., Li, Y., Liu, D., Huang, R., Liu, W., Zhu, X., Wang, F.Y., Ma, Z., et al.: Lumina-next: Making lumina-t2x stronger and faster with next-dit. Advances in Neural Information Processing Systems **37**, 131278–131315 (2024)

## A.1 Implementation Details

**Tokenizer training.** Following [7], we use three stages to finetune DINOv2-L [42] and train the decoder by the reconstruction objective. Specifically, we use four losses: 1) semantic preservation; 2) L1 loss; 3) perceptual loss; and 4) adversarial loss [15,17,18,20,21,26,30,47,53,57,66,71,74,82]. For each image $I_i$, feature $\boldsymbol{h}_i$ is extracted by original encoder $g$ for semantic preservation $\boldsymbol{h}_i = g(I_i)$. The final loss for the finetuned encoder $g\prime$ and decoder $D$ is defined as follows:

$$\mathcal{L}_{FT} = \omega_{sp}\|\boldsymbol{h}_i' - \boldsymbol{h}_i\|^2 + \mathcal{L}_{\text{L1}}(I_i', I_i) + \omega_p \mathcal{L}_{\text{perceptual}}(I_i', I_i) + \omega_g \mathcal{L}_{\text{GAN}}(I_i', I_i), \quad (11)$$

where $\boldsymbol{h}_i' = g\prime(I_i)$ and $I_i' = D(\boldsymbol{h}_i')$. We set $\omega_{sp} = 1, \omega_p = 1,$ and $\omega_g = 0.5$ following [7].

**Table 5: Optimization hyperparameters for the 1B Diffusion Transformer**.

| Hyperparameter | Value |
|---|---|
| Optimizer | AdamW |
| Max learning rate | $1 \times 10^{-4}$ |
| Min learning rate | $1 \times 10^{-5}$ |
| Learning rate schedule | Cosine |
| Warmup ratio | 0.003 |
| $\beta_1, \beta_2$ | 0.9, 0.999 |
| Global batch size | 1024 |
| Gradient norm clip | 1.0 |

**DiT Architecture.** The DiT backbone consists of 24 layers with 24 attention heads each, and a hidden dimension of 1536. We utilize the SwiGLU [51] activation function. The Classifier-Free Guidance (CFG) dropout probability is set to 0.1. The input latent shapes for the DiT are $1024 \times 16 \times 16$ when using the finetuned DINOv2 encoder and $768 \times 16 \times 16$ for the MAE encoder [22].

**Optimization.** Detailed optimization hyperparameters are provided in Tab. 5.

## A.2 Analysis of Effective Dimensionality on More Datasets

Effective dimensionality for tokenizers/encoders is evaluated on three additional larger datasets (ImageNet-21k [46], BLIP3o [9], and CC12M [5]) shown in Tab. 6, and the phenomenon is consistent with Tab. 1.

## A.3 Visual Understanding of Frozen VLM Backbone

To verify that the frozen VLM backbone (Qwen3-VL-2B-Instruct [1]) provides a sufficiently strong foundation for text-conditioned generation, we evaluate its vi-

**Table 6: Effective dimensionality analysis on ImageNet-21k [46], BLIP3o [9], and CC12M [5].**

| Tokenizer/Encoder | Full dim | ImageNet-21k | | | BLIP3o | | | CC12M | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | $R_{90}$ | $R_{95}$ | $R_{99}$ | $R_{90}$ | $R_{95}$ | $R_{99}$ | $R_{90}$ | $R_{95}$ | $R_{99}$ |
| DINOv1-B [4] | 768 | 499 | 594 | 701 | 483 | 582 | 697 | 481 | 581 | 696 |
| DINOv2-B-RAE [42, 85] | 768 | 505 | 610 | 726 | 491 | 601 | 723 | 494 | 602 | 724 |
| SigLIP2-B-RAE [65, 85] | 768 | 466 | 583 | 716 | 470 | 585 | 717 | 488 | 598 | 721 |
| MAE-B-RAE [22, 85] | 768 | 92 | 228 | 562 | 77 | 200 | 536 | 74 | 196 | 533 |
| MAE-B-IG-3B* [56] | 768 | 185 | 335 | 588 | 204 | 356 | 600 | 201 | 355 | 600 |
| DINOv2-L [42] | 1024 | 671 | 811 | 966 | 659 | 804 | 964 | 660 | 804 | 963 |
| Finetuned DINOv2-L | 1024 | 124 | 293 | 720 | 114 | 276 | 707 | 121 | 287 | 716 |

sual reasoning and understanding capabilities for reference. We assess its performance on seven representative multimodal benchmarks, including MMMU [78], MMMU-Pro [79], MathVista [38], MathVerse [81], MathVision [68], MMStar [10], and Charxiv-RQ [70]. The results are shown in Tab. 7.

**Table 7:** Performance of frozen VLM backbone across seven visual reasoning and understanding benchmarks.

| | MMMU | MMMU-Pro | MathVista | MathVerse | MathVision | MMStar | Charxiv-RQ |
|---|---|---|---|---|---|---|---|
| Frozen VLM Backbone | 52.9 | 37.2 | 61.9 | 53.2 | 30.9 | 58.5 | 25.8 |

## A.4 Extended Derivations for High-Dimensional Flow Matching

**Notation.** In Sec. 3.2 of the main paper, the learned predictors $\boldsymbol{v}_\theta$ and $\boldsymbol{x}_\theta$ are trained to approximate the corresponding Bayes-optimal predictors under squared loss. More precisely, we define these optimal high-dimensional predictors as

$$\boldsymbol{v}^{h,*}(\boldsymbol{z}_t^h, t) := \mathbb{E}[\boldsymbol{\epsilon}_h - \boldsymbol{x}_0 \mid \boldsymbol{z}_t^h], \qquad \boldsymbol{x}^{h,*}(\boldsymbol{z}_t^h, t) := \mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{z}_t^h], \tag{12}$$

and similarly in the low-dimensional space,

$$\boldsymbol{v}^{l,*}(\boldsymbol{z}_t^l, t) := \mathbb{E}[\boldsymbol{\epsilon}_l - \boldsymbol{c}_0 \mid \boldsymbol{z}_t^l], \qquad \boldsymbol{x}^{l,*}(\boldsymbol{z}_t^l, t) := \mathbb{E}[\boldsymbol{c}_0 \mid \boldsymbol{z}_t^l]. \tag{13}$$

Accordingly, the notation $\boldsymbol{x}_\theta^l(\boldsymbol{z}_t^h, t)$ and $\boldsymbol{v}_\theta^l(\boldsymbol{z}_t^h, t)$ introduced in the main paper (e.g., Eq. 7) are formulated such that their dependence on the observed state $\boldsymbol{z}_t^h$ occurs entirely through its projected manifold component $\boldsymbol{z}_t^l = Q^\top \boldsymbol{z}_t^h$. We formalize this below.

**Setup.** Let the observed high-dimensional feature be $\boldsymbol{x}_0 \in \mathbb{R}^h$ and the true latent variable be $\boldsymbol{c}_0 \in \mathbb{R}^l$. For simplicity, we assume the relationship is linear: $\boldsymbol{x}_0 = Q\boldsymbol{c}_0$, where the columns of $Q \in \mathbb{R}^{h\times l}$ form an orthonormal basis for the subspace (i.e., $Q^\top Q = I_l$). We denote the orthogonal projectors $P_Q = QQ^\top$ and $P_\perp = I_h - QQ^\top$.

**Orthogonal decomposition of the forward process.** In standard flow matching, the forward process constructs the noisy state $\boldsymbol{z}_t^h = (1-t)\boldsymbol{x}_0 + t\boldsymbol{\epsilon}_h$, where $\boldsymbol{\epsilon}_h \sim \mathcal{N}(0, I_h)$ and $t \in [0,1]$ ($t$=0 corresponds to clean data, $t$=1 to pure Gaussian noise).

Because $\boldsymbol{x}_0$ is assumed to lie in the column space of $Q$, we can decompose the isotropic Gaussian noise $\boldsymbol{\epsilon}_h$ into two orthogonal components: a component parallel to the data subspace, $\boldsymbol{\epsilon}_l = Q^\top\boldsymbol{\epsilon}_h \sim \mathcal{N}(0, I_l)$, and a component normal to it, $\boldsymbol{\epsilon}_\perp = P_\perp\boldsymbol{\epsilon}_h$. These two components are independent, since $\mathrm{Cov}(\boldsymbol{\epsilon}_l, \boldsymbol{\epsilon}_\perp) = Q^\top P_\perp = Q^\top - Q^\top QQ^\top = 0$ and both are Gaussian. Moreover, $\boldsymbol{c}_0$ (drawn from the data distribution) is independent of $\boldsymbol{\epsilon}_h$, and hence independent of both $\boldsymbol{\epsilon}_l$ and $\boldsymbol{\epsilon}_\perp$.

Substituting into the forward process:

$$\boldsymbol{z}_t^h = Q\underbrace{((1-t)\boldsymbol{c}_0 + t\boldsymbol{\epsilon}_l)}_{\boldsymbol{z}_t^l} + t\boldsymbol{\epsilon}_\perp, \tag{14}$$

where $\boldsymbol{z}_t^l \in \mathbb{R}^l$ is the manifold component. Crucially, $\boldsymbol{z}_t^l$ is a function of $(\boldsymbol{c}_0, \boldsymbol{\epsilon}_l)$ only, and is therefore independent of $\boldsymbol{\epsilon}_\perp$. Applying $P_\perp$ to both sides isolates the normal component deterministically for $t > 0$:

$$P_\perp\boldsymbol{z}_t^h = t\boldsymbol{\epsilon}_\perp, \quad \text{i.e.,} \quad \boldsymbol{\epsilon}_\perp = \tfrac{1}{t}P_\perp\boldsymbol{z}_t^h. \tag{15}$$

This means $\boldsymbol{\epsilon}_\perp$ is fully determined by $\boldsymbol{z}_t^h$—no statistical estimation is required to recover it. We also note that $Q^\top\boldsymbol{z}_t^h = \boldsymbol{z}_t^l$ (since $Q^\top\boldsymbol{\epsilon}_\perp = Q^\top P_\perp\boldsymbol{\epsilon}_h = 0$), so the model can recover the manifold state from $\boldsymbol{z}_t^h$ via projection.

**The inefficiency of high-dimensional velocity ($\boldsymbol{v}$) prediction.** The velocity target is $\boldsymbol{v} = \boldsymbol{\epsilon}_h - \boldsymbol{x}_0$. Using the decomposition $\boldsymbol{\epsilon}_h = Q\boldsymbol{\epsilon}_l + \boldsymbol{\epsilon}_\perp$ and $\boldsymbol{x}_0 = Q\boldsymbol{c}_0$:

$$\boldsymbol{v}^{h,*}(\boldsymbol{z}_t^h, t) = \mathbb{E}[\boldsymbol{v} \mid \boldsymbol{z}_t^h] = \mathbb{E}[Q(\boldsymbol{\epsilon}_l - \boldsymbol{c}_0) \mid \boldsymbol{z}_t^h] + \mathbb{E}[\boldsymbol{\epsilon}_\perp \mid \boldsymbol{z}_t^h]. \tag{16}$$

By Eq. (15), the second term equals $\frac{1}{t}P_\perp\boldsymbol{z}_t^h$. For the first term, since $\boldsymbol{\epsilon}_l - \boldsymbol{c}_0$ depends only on $(\boldsymbol{c}_0, \boldsymbol{\epsilon}_l)$, which are jointly independent of $\boldsymbol{\epsilon}_\perp$, conditioning on $\boldsymbol{z}_t^h = (Q\boldsymbol{z}_t^l, t\boldsymbol{\epsilon}_\perp)$ reduces to conditioning on $\boldsymbol{z}_t^l$ alone. Therefore:

$$\boldsymbol{v}^{h,*}(\boldsymbol{z}_t^h, t) = Q\boldsymbol{v}^{l,*}(Q^\top\boldsymbol{z}_t^h, t) + \frac{1}{t}P_\perp\boldsymbol{z}_t^h, \tag{17}$$

where $\boldsymbol{v}^{l,*}(\boldsymbol{z}_t^l, t) = \mathbb{E}[\boldsymbol{\epsilon}_l - \boldsymbol{c}_0 \mid \boldsymbol{z}_t^l]$ is the optimal velocity predictor in the $l$-dimensional latent space.

This reveals a critical inefficiency: the second term is a deterministic, $(h-l)$-dimensional linear function of the input that lies orthogonal to the data subspace

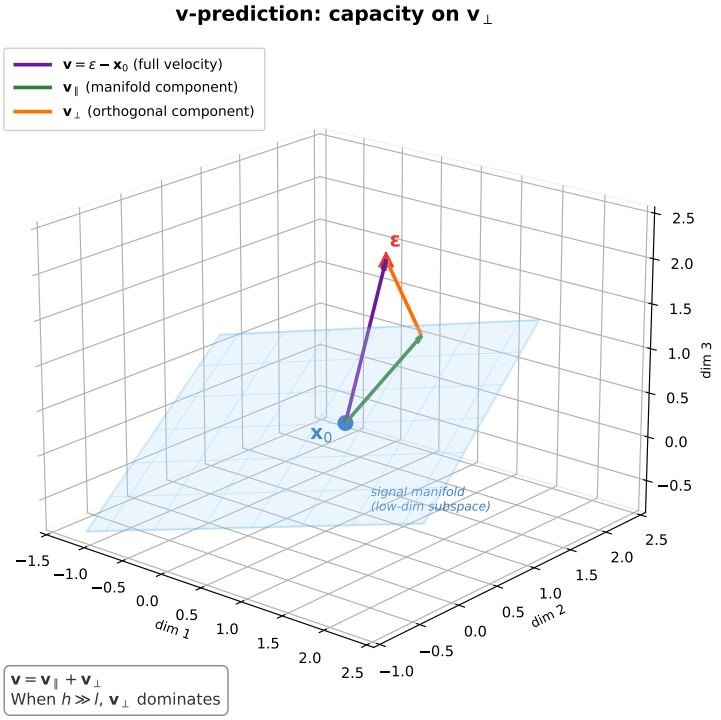



**Fig. 8: Decomposition of the velocity target under the subspace model.** The full velocity $\boldsymbol{v} = \boldsymbol{\epsilon} - \boldsymbol{x}_0$ (purple) decomposes into a manifold-aligned component $\boldsymbol{v}_\parallel$ (green) and an orthogonal component $\boldsymbol{v}_\perp$ (orange). When $h \gg l$, $\boldsymbol{v}_\perp$ dominates the target norm, forcing the model to spend capacity on an uninformative direction. In contrast, the $\boldsymbol{x}_0$-prediction target $\boldsymbol{x}_0 = Q\boldsymbol{c}_0$ lies entirely on the signal manifold, bypassing $\boldsymbol{v}_\perp$ altogether.

and carries no additional information about the clean latent $\boldsymbol{c}_0$ beyond what is already contained in $\boldsymbol{z}_t^l$, yet the model must still allocate capacity to represent it. In terms of target norm, $\mathbb{E}[\|\boldsymbol{\epsilon}_\perp\|^2] = \mathrm{tr}(P_\perp) = h - l$, while the low-dimensional velocity target has expected squared norm $l + \mathbb{E}[\|\boldsymbol{c}_0\|^2]$. Using $R_{90}$ as an empirical proxy for the intrinsic dimension, the finetuned DINOv2-L features ($h = 1024$, $R_{90} = 129$) suggest that a large fraction of the ambient space lies outside the effective signal subspace (see Fig. 8).

**The efficiency of $\boldsymbol{x}_0$-prediction.** To bypass modeling the orthogonal noise, we consider direct $\boldsymbol{x}_0$-prediction, which is also inspired by [33]:

$$\boldsymbol{x}^{h,*}(\boldsymbol{z}_t^h, t) = \mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{z}_t^h] = \mathbb{E}[Q\boldsymbol{c}_0 \mid Q\boldsymbol{z}_t^l, t\boldsymbol{\epsilon}_\perp]. \tag{18}$$

Since $(\boldsymbol{c}_0, \boldsymbol{\epsilon}_l)$ are jointly independent of $\boldsymbol{\epsilon}_\perp$, and $\boldsymbol{z}_t^l$ is a function of $(\boldsymbol{c}_0, \boldsymbol{\epsilon}_l)$ alone, conditioning on $\boldsymbol{z}_t^l$ cannot introduce dependence on $\boldsymbol{\epsilon}_\perp$, i.e., $\boldsymbol{c}_0 \perp\!\!\!\perp \boldsymbol{\epsilon}_\perp \mid \boldsymbol{z}_t^l$. Therefore, $t\boldsymbol{\epsilon}_\perp$ provides zero additional information about $\boldsymbol{c}_0$ beyond what $\boldsymbol{z}_t^l$

already contains, and the conditional expectation simplifies to:

$$\boldsymbol{x}^{h,*}(\boldsymbol{z}_t^h, t) = \mathbb{E}[Q\boldsymbol{c}_0 \mid Q\boldsymbol{z}_t^l] = Q\mathbb{E}[\boldsymbol{c}_0 \mid \boldsymbol{z}_t^l] = Q\boldsymbol{x}^{l,*}(\boldsymbol{z}_t^l, t). \tag{19}$$

Eq. (19) shows that the optimal $\boldsymbol{x}_0$-predictor always outputs vectors in $\mathrm{col}(Q)$—the orthogonal component $\boldsymbol{\epsilon}_\perp$ never appears in the target. By stripping $\boldsymbol{\epsilon}_\perp$ from the optimization objective, the network focuses its entire capacity on resolving $\boldsymbol{c}_0$ within the informative $l$-dimensional subspace, making training considerably more efficient when $h \gg l$.