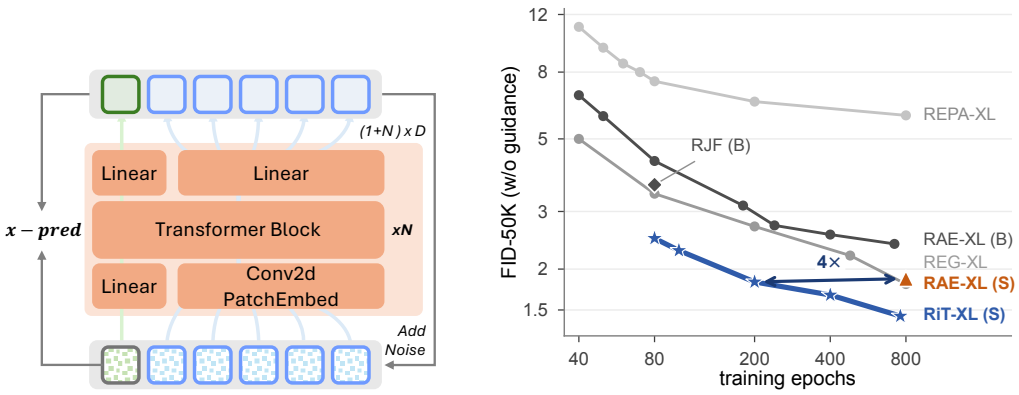
# RiT: A Simple Recipe for Standard Diffusion Transformers in Representation Space

**Anonymous authors**
Paper under double-blind review

## Abstract

Diffusion models trained in the feature space of a frozen representation encoder such as DINOv2 now rival those trained in VAE latents, but existing methods add machinery specific to such spaces: a wide prediction head, or Riemannian transport on the hypersphere. We show that a standard Diffusion Transformer reaches strong results without either. The *Representation Image Transformer* (RiT) is a DiT trained with $x$-prediction on element-wise standardized DINOv2 features, with a dimension-aware time shift and joint diffusion of the encoder's `[CLS]` token. With the same frozen encoder, decoder and parameter count as RAE's DiT-XL, RiT improves ImageNet $256\times256$ FID from 1.87 to 1.45 without guidance and from 1.41 to 1.14 with guidance, matches the baseline's 800-epoch FID after 200 epochs, and samples at FID 2.0 with 5 Heun steps without distillation. On frozen SigLIP2 features it reaches FID 1.52 and 1.22. We also compare pixels, VAE latents and representation features. Pixels and DINOv2 have similar intrinsic-dimension estimates, yet DINOv2's covariance has a normalized spectral condition number about $35\times$ smaller, and its coordinates have lower absolute excess kurtosis. On three tokenizers that differ only in their alignment loss, these covariance statistics order generation quality. Code and models will be released.




(a) Architecture (b) FID against training epochs

**Figure 1: RiT is a standard DiT that trains fast in representation space.** (a) The noised `[CLS]` token and $N$ patch tokens of a frozen DINOv2-S encoder (each of dimension $D$) share self-attention in the DiT blocks, and linear heads predict $\hat{\mathbf{z}}_0$ and $\hat{\mathbf{z}}_{\text{cls},0}$. (b) FID-50K without guidance; (S) and (B) denote DINOv2-S and DINOv2-B encoders. With the same encoder, decoder and parameter count as RAE-XL (S), RiT-XL reaches its 800-epoch FID after 200 epochs (arrow). RiT-XL is evaluated with 50 Heun steps and baselines use their published results; one epoch is one pass over ImageNet-1K ($\approx$834 steps for RiT-XL).

## 1 Introduction

Diffusion in the feature space of a frozen, pretrained representation encoder has become a strong alternative to diffusion in VAE latents. RAE (Zheng et al., 2025) pairs a frozen DINOv2 encoder (Oquab et al., 2024) with a trained pixel decoder and generates directly in the encoder's feature space, with results competitive with latent diffusion on ImageNet. Such spaces are high-dimensional, and LayerNorm places every token on a sphere. Existing methods adapt the generator to this setting. RAE



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

widens the denoiser with a DDT prediction head (Wang et al., 2025b) on top of $v$-prediction, and concurrent work (Kumar & Patel, 2026) keeps a standard DiT but replaces Euclidean transport with Riemannian flow matching on the sphere. In scaling RAE to text-to-image generation, Tong et al. (2026) describe the wide head as a patch for capacity-constrained models: it gives a large gain for a 0.5B DiT, whose width only slightly exceeds the token dimension, and a diminishing gain beyond 2.4B parameters, so they use a standard DiT at scale.

We study the class-conditional ImageNet setting in which these additions were introduced. There the backbone is already wider than the tokens (1152 against 384 channels for DINOv2-S), and RAE's own DiT-XL on DINOv2-S omits the head. We therefore ask how far a standard DiT (Peebles & Xie, 2023), by which we mean a modernized DiT without a representation-specific wide head or non-Euclidean transport, can be improved by its training recipe. With the same frozen encoder, decoder and parameter count as RAE's DiT-XL, our recipe improves ImageNet $256\times256$ FID from 1.87 to 1.45 without guidance and from 1.41 to 1.14 with guidance, and matches the baseline's 800-epoch FID after 200 epochs (Section 4; Appendix B lists how the two recipes differ). With guidance it is on par with $\text{DiT}^{\text{DH}}$-XL (1.16 in our reproduction), which uses the larger DINOv2-B encoder, a DDT head and $24\%$ more parameters. Few steps suffice for sampling: without distillation, 5 Heun steps give FID 2.0 and 10 steps give 1.25 with guidance.

The recipe consists of existing components, and we state the problem each one addresses in representation space (Section 3). *Element-wise standardization* removes a $1414\times$ spread of per-element variances; a global scale factor leaves this spread unchanged. *$x$-prediction* (Li & He, 2025) has the network regress the clean feature, whose per-sample target is always an encoder output, while the linear path from noise to data leaves the set of encoder outputs; it improves FID over $v$-prediction at every checkpoint we evaluated. A *dimension-aware time shift* (Zheng et al., 2025; Esser et al., 2024) adapts the noise schedule to tokens with 384 channels; replacing RiT's time distribution with JiT's has the largest effect among our ablations. Finally, representation space offers a component that VAE latents lack: the encoder's global `[CLS]` token, which we diffuse jointly with the patch tokens. We call the resulting model the **Representation Image Transformer (RiT)**.

We also analyze the spaces themselves (Section 5). Pixels and DINOv2-B features have the same ambient dimension and similar TwoNN intrinsic-dimension estimates ($33.6 \pm 1.3$ against $32.6 \pm 0.8$), yet DINOv2's covariance has higher effective rank and a normalized condition number about $35\times$ smaller along the transport path, and its coordinates have lower excess kurtosis. On three tokenizers that share architecture, data and reconstruction quality and differ only in their alignment loss (Yao et al., 2025), the covariance statistics order generation quality. These diagnostics describe how the spaces differ; they are correlational and do not by themselves explain why the recipe works.

Our contributions are:

- **A simple recipe for a standard DiT in representation space** (Sections 3 and 4.4). The four components above need no representation-specific head or non-Euclidean transport. Our ablations measure the role of each: the time distribution has the largest effect, and joint `[CLS]` diffusion and $x$-prediction each give a consistent gain.
- **A strong model that also samples well in few steps** (Section 4). RiT-XL is among the best-performing models of its size in Table 1 (FID 1.45 without guidance, 1.14 with guidance). It improves on RAE's DiT-XL at the same size, is on par with the DDT-head model at $19\%$ fewer parameters, samples well with 5 to 10 Heun steps without distillation, and transfers to SigLIP2 (FID 1.52 and 1.22).
- **Geometric diagnostics of representation spaces** (Section 5). At similar intrinsic-dimension estimates, DINOv2 differs from pixels in effective rank, conditioning and marginal kurtosis; SigLIP2 falls in the same range, and on a controlled family of tokenizers the covariance statistics order generation quality.

## 2 Related Work

**Diffusion in learned latent spaces.** Latent diffusion (Rombach et al., 2022) moves diffusion into a VAE latent, and DiT and SiT (Peebles & Xie, 2023; Ma et al., 2024) replace the U-Net with a transformer. A line of work makes such latents easier to model: VA-VAE, EQ-VAE and Diffusability regularize the autoencoder (Yao et al., 2025; Kouzelis et al., 2025; Skorokhodov et al., 2025), REPA-E

tunes it end to end (Leng et al., 2025), REPA and REG align the denoiser with DINOv2 features (Yu et al., 2024; Wu et al., 2025), and MAETok, SVG and UniFlow train the tokenizer against a vision-foundation-model objective (Chen et al., 2025a; Shi et al., 2025; Yue et al., 2026). FAE compresses frozen DINOv2 features into a compact latent with a single trained attention layer (Gao et al., 2025). RiT does not train or adapt the tokenizer; it diffuses in the output space of a frozen encoder.

**Diffusion in representation space.** RAE (Zheng et al., 2025) generates in the space of a frozen DINOv2 encoder and adds a DDT head (Wang et al., 2025b) for wide tokens. Tong et al. (2026) scale this approach to text-to-image generation; there the head's gain is largest for a 0.5B DiT whose width only slightly exceeds the token dimension and diminishes beyond 2.4B parameters, and a standard DiT with the dimension-dependent shift is used at scale. Concurrent RJF (Kumar & Patel, 2026) replaces Euclidean transport with Riemannian flow matching (Chen & Lipman, 2023). ReDi and Latent Forcing (Kouzelis et al., 2026; Baade et al., 2026) generate VAE latents jointly with representation features, so a non-semantic stream remains in the diffusion state; RiT's state consists of encoder features only. RiT keeps a standard DiT and Euclidean transport.

**Prediction targets and preconditioning.** JiT (Li & He, 2025) shows that $x$-prediction succeeds in pixel space where $\epsilon$- and $v$-prediction fail, because only the clean-image target lies on a low-dimensional manifold. EDM (Karras et al., 2022) preconditions network inputs and outputs per noise level, and preconditioned flow matching (Ahamed et al., 2026) relates the conditioning of the data covariance to optimization; we use the latter's covariance $\Sigma_t$ as a diagnostic across spaces.

## 3 RiT: A Standard DiT for Representation-Space Diffusion

RiT (Figure 2) generates in the feature space of a frozen encoder–decoder pair. We reuse RAE's (Zheng et al., 2025) frozen DINOv2-with-registers encoder (Oquab et al., 2024), which maps an image to patch tokens $\mathbf{z} \in \mathbb{R}^{d\times h\times w}$ and a `[CLS]` token $\mathbf{z}_{\text{cls}} \in \mathbb{R}^d$, and RAE's ViT decoder, which maps patch tokens back to pixels; neither is adapted for generation. The denoiser is the modernized DiT used by JiT and LightningDiT (Li & He, 2025; Yao et al., 2025), with SwiGLU (Shazeer, 2020), RMSNorm (Zhang & Sennrich, 2019), 2D RoPE (Su et al., 2024), QK-normalization (Henry et al., 2020) and in-context class tokens. We call it *standard* because it contains no component introduced for representation-space diffusion: its output heads are linear layers and the transport is Euclidean. The one design addition is a second input and output for the `[CLS]` token (§3.3). Details are in Appendix J.

**Flow matching.** Flow matching (Liu et al., 2022; Esser et al., 2024) learns a velocity field that transports noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to data $\mathbf{z}_0$ along $\mathbf{z}_t = t\,\mathbf{z}_0 + (1{-}t)\,\boldsymbol{\epsilon}$, with $t{=}0$ pure noise and $t{=}1$ clean data. The velocity along the path is $\mathbf{v} = \mathbf{z}_0 - \boldsymbol{\epsilon}$, and $v$-prediction trains a network $\mathbf{v}_\theta(\mathbf{z}_t, t, c)$ with class label $c$ to regress it, $\mathcal{L}_v = \mathbb{E}\|\mathbf{v}_\theta(\mathbf{z}_t, t, c) - \mathbf{v}\|^2$. Samples are obtained by integrating $\dot{\mathbf{z}}_t = \mathbf{v}_\theta$ from $t{=}0$ to $t{=}1$, here with Heun's method.

### 3.1 Element-wise Standardization

Latent diffusion models rescale their latents toward the unit-variance source: latent diffusion multiplies VAE latents by a global constant (Rombach et al., 2022), EDM preconditions network inputs and outputs per noise level (Karras et al., 2022), and RAE standardizes DINOv2 features per element (Zheng et al., 2025), as we do. We keep this step unchanged and explain why representation features need it at this granularity. DINOv2's per-element variances range from $0.004$ to $5.72$ on ImageNet, a factor of $1414$. A global scale $c$ leaves the ratio between per-element variances unchanged; it maps the data covariance $\mathbf{H}$ to $c^2\mathbf{H}$, and $\kappa(c^2\mathbf{H}) = \kappa(\mathbf{H})$. Under an MSE loss the high-variance elements would then dominate the gradient. We therefore standardize every element of the patch tokens and

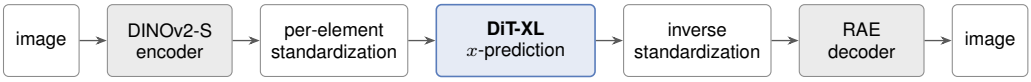



**Figure 2: RiT pipeline.** Gray boxes are frozen and the blue box is trained. The encoder is used during training and the decoder during sampling; patch tokens and the `[CLS]` token are standardized per element and denoised jointly (Figure 1a).

of the [CLS] token, $\tilde{z}_{c,h,w} = (z_{c,h,w} - \mu_{c,h,w})/\sqrt{\sigma^2_{c,h,w} + \epsilon}$, with statistics precomputed on the training set, and invert the map before decoding. From here on, $\mathbf{z}$ denotes the standardized feature. Section 5 measures how standardization changes the covariance spectrum of the space RiT is trained in.

### 3.2 $x$-PREDICTION

Because RAE removes the affine parameters of the encoder's final LayerNorm, every raw token has zero mean over channels and norm $\sqrt{d}$ (up to the LayerNorm $\epsilon$). Encoder outputs therefore lie on a product of $hw$ spheres, and after the element-wise affine standardization on the image of that set, a lower-dimensional subset of the ambient space. The linear path from noise to data leaves this subset: for independent $\mathbf{z}_0$ and $\boldsymbol{\epsilon}$ with equal second moments, per token in raw space or over the whole map in standardized space,

$$\mathbb{E}\|\mathbf{z}_t\|^2 = \left(t^2 + (1-t)^2\right) \mathbb{E}\|\mathbf{z}_0\|^2, \tag{1}$$

which is half the endpoint value at $t=\frac{1}{2}$ (Appendix E). Kumar & Patel (2026) identify the resulting mismatch between linear transport and the spherical geometry of the features and replace the path with a geodesic. We keep the linear path: with per-element standardization, Euclidean flow matching trains stably in our experiments with either $x$- or $v$-prediction (Table 2). $x$-prediction does not constrain the path or the network output to the set of encoder outputs; it changes which function the network represents.

With $x$-prediction (Li & He, 2025), the network outputs the clean feature, $\hat{\mathbf{z}}_0 = f_\theta(\mathbf{z}_t, t, c)$, and the velocity follows in closed form, $\hat{\mathbf{v}}_\theta = (\hat{\mathbf{z}}_0 - \mathbf{z}_t)/(1-t)$. We train with the velocity loss $\mathcal{L}_{\text{fm}} = \mathbb{E}\|\hat{\mathbf{v}}_\theta - \mathbf{v}\|^2$, which equals the $x$-loss $\|\hat{\mathbf{z}}_0 - \mathbf{z}_0\|^2$ reweighted by $(1-t)^{-2}$ (Appendix C). The two parameterizations share this objective but define different functions for the network to fit. The per-sample $x$-target $\mathbf{z}_0$ is always an encoder output, and the network regresses its posterior mean $\mathbb{E}[\mathbf{z}_0 \mid \mathbf{z}_t, t, c]$. The $v$-target $\mathbf{z}_0 - \boldsymbol{\epsilon}$ carries the $D$-dimensional noise, half of its expected energy, so a $v$-predictor has to output an estimate of $\mathbb{E}[\boldsymbol{\epsilon} \mid \mathbf{z}_t, t, c]$, a component that no encoder output contains; under $x$-prediction this component is supplied analytically by the $-\mathbf{z}_t/(1-t)$ term.[1]

**Effect.** In our ablations, $x$-prediction improves FID over $v$-prediction at every checkpoint, with and without [CLS] (Section 4.4). Since $v$-prediction also trains stably under the same objective, the gain comes from the parameterization.

### 3.3 JOINT [CLS]–PATCH DIFFUSION

A VAE latent has no global token, but a representation encoder produces one alongside the patches. REG (Wu et al., 2025) shows that diffusing a DINOv2 [CLS] token together with SD-VAE latents improves generation; since its two token types come from different encoders, it runs a second encoder during training and connects the two spaces with a REPA alignment loss (Yu et al., 2024). We adapt this idea to representation space. In RiT, the [CLS] and the patch tokens are two outputs of the same frozen encoder, so the [CLS] is one more token of the diffusion state and neither the second encoder nor the alignment loss is needed. The noised [CLS] is projected and prepended to the patch sequence (Figure 1a), all tokens attend to each other, a separate linear head predicts $\hat{\mathbf{z}}_{\text{cls},0}$, and the objective is $\mathcal{L} = \mathcal{L}_{\text{fm}} + \lambda\mathcal{L}_{\text{cls}}$ with the same parameterization and $\lambda$=0.2. During training, the [CLS] noise is drawn independently of the patch noise. At inference both are integrated jointly and only the patch tokens are decoded. For the reported results we initialize the [CLS] noise as the spatial mean of the patch noise (*coupled noise*), which correlates it with the patch noise and lowers its per-coordinate variance to $1/(hw) = 1/256$. With the time-shift grid used for all reported results, this inference-time choice changes FID by at most 0.06 (at most 0.12 over the six grids we evaluate): with independent [CLS] noise the 25-step FIDs are 1.47 without and 1.15 with guidance, against 1.45 and 1.14 (Tables 11 and 12). Joint diffusion improves FID from 1.63 to 1.44 (Table 2); Appendix O visualizes the [CLS]–patch attention.

[1] These are the unclamped forms. In the implementation, $1-t$ is replaced by $a(t) = \max(1-t, 0.05)$ in the target and in the $x$-parameterized prediction, and the $v$-predictor regresses the same clamped target, so both parameterizations share one objective; for $t > 0.95$ this target is a rescaled velocity of the linear path (Appendix C).

**Table 1: Class-conditional generation on ImageNet 256×256.** RiT-XL uses a frozen DINOv2-S encoder and 25 Heun steps. DiT$^{DH}$-XL adds a DDT head and uses DINOv2-B; FAE compresses frozen DINOv2-G features into a 32-channel latent. *Reproduced by us, with AutoGuidance (Karras et al., 2024) for the guided results. RiT uses classifier-free guidance. Bold: best per column.

| Method | Tokenizer | Epochs | #Params | w/o guidance | | | | w/ guidance | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | FID↓ | IS↑ | Prec.↑ | Rec.↑ | FID↓ | IS↑ | Prec.↑ | Rec.↑ |
| *Pixel space* | | | | | | | | | | | |
| ADM (Dhariwal & Nichol, 2021) | – | 400 | 554M | 10.94 | 101.0 | 0.69 | 0.63 | 3.94 | 215.8 | **0.83** | 0.53 |
| PixelFlow-XL (Chen et al., 2025b) | – | 320 | 677M | – | – | – | – | 1.98 | 282.1 | 0.81 | 0.60 |
| PixNerd-XL (Wang et al., 2025a) | – | 320 | 700M | – | – | – | – | 1.93 | 298.0 | 0.80 | 0.60 |
| PixelDiT-XL (Yu et al., 2025) | – | 320 | 797M | – | – | – | – | 1.61 | 292.7 | 0.78 | 0.64 |
| JiT-G (Li & He, 2025) | – | 600 | 2B | – | – | – | – | 1.82 | 292.6 | 0.79 | 0.62 |
| *VAE latent space* | | | | | | | | | | | |
| DiT-XL (Peebles & Xie, 2023) | SD-VAE | 1400 | 675M | 9.62 | 121.5 | 0.67 | 0.67 | 2.27 | 278.2 | **0.83** | 0.57 |
| SiT-XL (Ma et al., 2024) | SD-VAE | 1400 | 675M | 8.61 | 131.7 | 0.68 | 0.67 | 2.06 | 270.3 | 0.82 | 0.59 |
| MaskDiT (Zheng et al., 2023) | SD-VAE | 1600 | 675M | 5.69 | 177.9 | 0.74 | 0.60 | 2.28 | 276.6 | 0.80 | 0.61 |
| MDTv2-XL (Gao et al., 2023) | SD-VAE | 1080 | 675M | – | – | – | – | 1.58 | **314.7** | 0.79 | 0.65 |
| REPA-XL (Yu et al., 2024) | SD-VAE | 800 | 675M | 5.90 | 158.3 | 0.70 | 0.67 | 1.29 | 306.3 | 0.79 | 0.64 |
| DDT-XL (Wang et al., 2025b) | SD-VAE | 400 | 675M | 6.27 | 154.7 | 0.68 | **0.69** | 1.26 | 310.6 | 0.79 | 0.65 |
| REG-XL (Wu et al., 2025) | SD-VAE | 800 | 675M | 1.80 | – | – | – | 1.36 | 299.4 | 0.77 | 0.66 |
| LightningDiT (Yao et al., 2025) | VA-VAE | 800 | 675M | 2.17 | 205.6 | 0.77 | 0.65 | 1.35 | 295.3 | 0.79 | 0.65 |
| *Representation space* | | | | | | | | | | | |
| SVG-XL (Shi et al., 2025) | SVGTok | 1400 | 675M | 3.36 | 181.2 | – | – | 1.92 | 264.9 | – | – |
| RAE-XL (Zheng et al., 2025) | RAE-DINOv2-S | 800 | 676M | 1.87 | 209.7 | 0.80 | 0.63 | 1.41 | 309.4 | 0.80 | 0.63 |
| DiT$^{DH}$-XL* (Zheng et al., 2025) | RAE-DINOv2-B | 800 | 839M | 1.51 | **242.9** | 0.79 | 0.63 | 1.16 | 261.0 | 0.77 | **0.67** |
| FAE-XL (Gao et al., 2025) | FAE-DINOv2-G | 800 | 675M | 1.48 | 239.8 | **0.81** | 0.63 | 1.29 | 268.0 | 0.80 | 0.64 |
| RiT-XL (ours) | RAE-DINOv2-S | 760 | 676M | **1.45** | 231.6 | **0.81** | 0.62 | **1.14** | 299.7 | 0.80 | 0.63 |

### 3.4 Dimension-Aware Time Shift

The effective signal-to-noise ratio of a token grows with its dimension: the signal within a token is redundant while the noise is independent across its $d$ coordinates (Hoogeboom et al., 2023). A DINOv2-S token has $d$=384 coordinates against 3 for a pixel, so a pixel-space schedule spends too little training on high-noise states. We apply the time shift of SD3 and RAE (Esser et al., 2024; Zheng et al., 2025) without modification: $X' = sX/(1+(s-1)X)$ with $X \sim \text{logit-}\mathcal{N}(0,1)$, $t = 1 - X'$ and $s = \sqrt{hwd/4096} \approx 4.9$, which lowers the median training $t$ from ≈0.31 under JiT's schedule to ≈0.17 (Appendix I); Tong et al. (2026) also find this shift essential for text-to-image training in RAE space. Replacing RiT's time distribution with JiT's pixel-space distribution raises FID from 1.44 to 3.17 at 760 epochs (Section 4.4).

## 4 Experiments

**Setup.** RiT-XL (28 layers, width 1152, 16 heads, 676M parameters) is trained on ImageNet-1K at 256×256 in the space of a frozen DINOv2-S encoder ($d$=384, 16×16 tokens) with RAE's pretrained decoder (Zheng et al., 2025), for 760 epochs at batch size 1536 on 8 B200 GPUs (≈12 minutes per epoch). We report FID-50K with class-balanced sampling (50 images per class), following RAE. Unless stated otherwise, sampling uses 25 Heun steps (24 Heun steps and a final Euler step, 49 function evaluations) with the time-shift discretization and coupled `[CLS]` noise, and guided results use classifier-free guidance with scale 3.7 on $t \in [0.1, 0.98]$; within this interval each function evaluation requires a conditional and an unconditional forward pass. Hyperparameters and pseudocode are in Appendix J.

### 4.1 Comparison with Prior Methods

The closest comparison is RAE-XL, which shares RiT's encoder, decoder and parameter count: RiT-XL improves FID from 1.87 to 1.45 without guidance and from 1.41 to 1.14 with guidance (Table 1). RAE-XL follows RAE's training recipe, so this compares two complete recipes (Appendix B lists the differences). Within our code, $v$-prediction without `[CLS]` gives 1.78, $x$-prediction 1.63 and the full recipe 1.44 (Table 2); the gap between 1.87 and 1.78 combines differences in backbone, optimization and sampler that we did not separate. Against DiT$^{DH}$-XL, which uses the larger DINOv2-B encoder and a DDT head for 839M parameters, RiT-XL is lower without guidance (1.45 against 1.51) and on par with guidance (1.14 with classifier-free guidance, against 1.16 in our AutoGuidance reproduction).

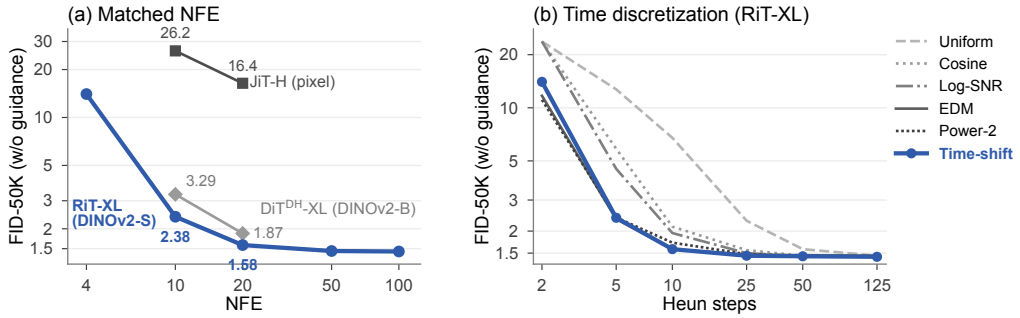



**Figure 3: Few-step FID.** (a) FID-50K without guidance at matched NFE, counting $K$ Heun steps as $2K$ evaluations (our sampler takes the last step with Euler, so it uses $2K-1$); baseline values are measured by us with each method's released checkpoint, code and sampler. (b) RiT-XL under six time discretizations (Appendix M), without guidance and with coupled `[CLS]` noise.

FAE-XL (1.48 without guidance) compresses the features of the largest DINOv2 variant into a compact latent with a trained attention layer, whereas RiT diffuses the features of the smallest variant directly. RiT's unguided FID is below the guided FID of every pixel-space model in the table and of DiT, SiT, MaskDiT and MDTv2.

### 4.2 Convergence in Training Epochs

Figure 1b plots FID against training epochs. The matched comparison is RAE-XL (S), a $v$-prediction DiT-XL with the same encoder, decoder and parameter count, trained with its own recipe: RiT-XL reaches its 800-epoch FID (1.87) after 200 epochs, a $4\times$ reduction, and ends at 1.44. At 100 epochs, RiT-XL (2.28) is better than RAE-XL (B), which uses a different encoder, at 720 epochs (2.39). RiT-XL passes REPA's 800-epoch FID (5.90) by epoch 80 and REG's between epochs 200 and 400, and at 80 epochs it reaches 2.48 against 3.62 for concurrent RJF. Epochs count passes over the data; because encoders, model sizes and throughput differ across these runs, we do not convert them into compute.

### 4.3 Few-Step Generation without Distillation

Few-step sampling is usually obtained by distillation or consistency training (Salimans & Ho, 2022; Song et al., 2023). RiT uses neither; we measure how its FID changes with the number of Heun steps.

**FID at matched NFE.** Without guidance, at 10 function evaluations RiT-XL reaches FID 2.38, against 3.29 for DiT$^{\text{DH}}$-XL and 26.2 for JiT-H; at 20 NFE the values are 1.58, 1.87 and 16.4 (Figure 3a). With guidance, 5 Heun steps give FID 1.99, 10 steps give 1.25, and 25 steps match the 50-step value of 1.14 (Table 12), whereas most VAE-latent models in Table 1 sample with $\sim$250 steps. We measured the baseline values with each method's released checkpoint and its own sampler, and did not apply our time grids to them. Figure 3a therefore shows that RiT samples well with few steps; it does not show that RiT is better suited to few-step sampling than the baselines.

**Time discretization.** The time grid matters at few steps (Figure 3b). With 5 steps, the three grids that place the most steps at high noise reach FID 2.34 (EDM), 2.39 (power-2) and 2.38 (time-shift), against 12.7 for uniform spacing, so the result does not depend on one tuned grid; the time-shift grid reuses the training shift $s$. At 50 steps all non-uniform grids agree to within 0.04 FID. Appendix F reports how much the decoded samples change with the number of steps.

### 4.4 Ablations

Table 2 changes one component at a time. Replacing the **time distribution** with JiT's has the largest effect (3.17 against 1.44); this row changes the base distribution together with the shift, so it does not isolate the shift. **Joint `[CLS]` diffusion** improves FID from 1.63 to 1.44, and doubling its weight to $\lambda$=0.4 gives 1.50. The **prediction target** gives a smaller but consistent gain. The paired no-`[CLS]` rows share the objective and differ only in the parameterization: $x$-prediction reaches 2.63 at 80 epochs and 1.63 at 760, against 3.17 and 1.78 for $v$-prediction. With `[CLS]`, $x$-prediction also leads at 80 epochs (2.48 against 2.86), so joint `[CLS]` diffusion improves both parameterizations. Removing the 32 **in-context class tokens** inherited from JiT, with class conditioning retained through adaLN, gives 1.50 without guidance and 1.20 with guidance (1.14 for the default).

**Table 2: Ablations.** FID-50K without guidance, 50 Heun steps. Each row changes one component of the default; the two no-`[CLS]` rows form a paired comparison of the prediction target, and the $v$-prediction row with `[CLS]` pairs with the default. †Replaces RiT's time distribution with JiT's (Appendix I).

| | 80 | 200 | 400 | 760 ep |
|---|---|---|---|---|
| RiT-XL default | 2.48 | 1.83 | 1.67 | **1.44** |
| JiT time distrib.† | – | 4.45 | 4.11 | 3.17 |
| $x$-pred., no `[CLS]` | 2.63 | 1.89 | 1.70 | 1.63 |
| $v$-pred., no `[CLS]` | 3.17 | 2.11 | 1.86 | 1.78 |
| $v$-pred., with `[CLS]` | 2.86 | – | – | – |
| `[CLS]` weight $\lambda$=0.4 | – | 1.86 | 1.64 | 1.50 |
| DINOv2-B encoder | – | 2.20 | 1.78 | 1.56 |
| w/o in-context tokens | – | – | – | 1.50 |

**Encoder size.** DINOv2-S outperforms DINOv2-B (1.44 against 1.56) with half as many channels. We reuse RAE's decoders unchanged, and the better reconstruction of DINOv2-B (rFID 0.57 against 0.64 for the noise-augmented decoders we use; Table 15b of Zheng et al., 2025) does not explain the gap, although other decoder-related factors are not excluded. DINOv2-B doubles the ambient dimension at a similar estimated intrinsic dimension ($\hat{d}$ between 30 and 35). Whether the larger target or the fixed denoiser width accounts for the gap has not been tested (Appendix K).

### 4.5 A Second Representation Space

We train RiT on frozen SigLIP2 features (Tschannen et al., 2025). On ImageNet 256×256 it reaches FID 1.52 without guidance and 1.22 with guidance. SigLIP2 falls in DINOv2's range on the covariance and marginal quantities of Table 3.

## 5 Geometric Diagnostics of Representation Spaces

Flow matching transports $\mathcal{N}(\mathbf{0}, \mathbf{I})$ to the data distribution. This section describes how the data distribution of several spaces relates to that source; the analysis is descriptive, and the recipe of Section 3 is not derived from it. We compare raw pixels (3×256×256), DINOv2-B features (768×16×16), which have the same ambient dimension $D$=196,608, and SD-VAE latents (4×32×32) (Rombach et al., 2022), a compression autoencoder trained without a representation objective. We add SigLIP2 (Tschannen et al., 2025), a representation encoder trained with a contrastive image–text objective. All quantities are computed on 10,000 ImageNet training images in the space each encoder produces; we also report them in the standardized DINOv2-S space in which RiT is trained. Table 3 collects the results and Figure 4 shows three of them in full.

**Table 3: Geometry of four spaces.** Raw features of 10,000 ImageNet images. Effective rank and $\tilde{\kappa}_{512}$ use the top 512 principal components (Figure 4); $\hat{d}$ is estimated on one random subsample of 5,000 points, with means over 10 subsamples in the text. Arrows give the direction closer to the isotropic Gaussian source.

| Quantity | Pixel | SD-VAE | DINOv2-B | SigLIP2 | DINOv2-B vs. pixel |
|---|---|---|---|---|---|
| Ambient dimension $D$ | 196,608 | 4,096 | 196,608 | 294,912 | equal |
| Intrinsic dimension $\hat{d}$ (TwoNN) | 35.3 | 70.9 | 33.5 | 80.4 | similar |
| Effective rank ↑ | 44.9 | 98.4 | **326.6** | 316.8 | 7.3× higher |
| Condition number $\tilde{\kappa}_{512}(\Sigma_{0.9})$ ↓ | 1,925 | 447 | **55.6** | 63.9 | 35× lower |
| Median $\lvert\gamma_j\rvert$ (absolute excess kurtosis) ↓ | 0.958 | 0.228 | **0.083** | 0.093 | 11.5× lower |
| Coordinates with $\lvert\gamma_j\rvert < 0.5$ ↑ | 0.0% | 74.2% | **98.7%** | 95.9% | |

**Table 4: Controlled tokenizers.** Same architecture, data, budget and latent shape (16×16×32); only the alignment loss differs (Yao et al., 2025). gFID: published FID-10K of LightningDiT-B trained on each (Table 2 of Yao et al., 2025); "none" is the LDM baseline there.

| Alignment | rFID | gFID↓ | erank↑ | $\tilde{\kappa}(\Sigma_{0.5})\downarrow$ | $\tilde{\kappa}(\Sigma_{0.9})\downarrow$ |
|---|---|---|---|---|---|
| none | 0.26 | 22.62 | 70.0 | 99.8 | 888.8 |
| MAE | 0.28 | 19.89 | 105.2 | 72.8 | 552.0 |
| DINOv2 | 0.28 | **15.82** | **117.8** | **42.0** | **336.3** |

**Table 5: Standardization on DINOv2-S**, the encoder RiT is trained on (10,000 images, $k$=512).

| | Raw | Standardized |
|---|---|---|
| Effective rank ↑ | 252.1 | **289.4** |
| $\tilde{\kappa}_{512}(\Sigma_{0.9})\downarrow$ | 143.4 | **81.1** |
| $\tilde{\kappa}_{512}(\Sigma_{0.5})\downarrow$ | 22.0 | **14.8** |
| Median $\lvert\gamma_j\rvert$ | 0.0879 | 0.0879 |

**Intrinsic dimension.** The TwoNN estimator (Facco et al., 2017) gives $\hat{d} = 33.6 \pm 1.3$ for pixels and $32.6 \pm 0.8$ for DINOv2-B (mean and standard deviation over 10 random subsamples of 5,000 points; Appendix F). At this sample size the estimates do not separate the two spaces, so the intrinsic-dimension estimate alone does not account for how differently they are modeled. We use it as a control variable: SD-VAE and SigLIP2 both have higher estimates than pixels and DINOv2-B (70.9 and 80.4), yet SigLIP2 falls in DINOv2's range on every quantity below while SD-VAE lies between pixels and DINOv2.

**Effective rank** erank $= \exp(-\sum_i \hat{\lambda}_i \log \hat{\lambda}_i)$, with $\hat{\lambda}_i$ the normalized PCA eigenvalues (Roy & Vetterli, 2007), measures how evenly variance spreads over directions; the isotropic source maximizes it. Pixel variance is concentrated in few directions: the first 50 components carry 72% of the total variance, against 18% for DINOv2-B (Figure 4a). The effective ranks are 45, 98 and 327 for pixels, SD-VAE and DINOv2-B. The pixel-to-DINOv2 ratio increases over the PCA budgets we tested, from 3.8× at $k$=128 to 7.3× at $k$=512 (Appendix G).

**Conditioning.** For any data distribution with covariance $\mathbf{H}$, independence of $\mathbf{z}_0$ and $\boldsymbol{\epsilon}$ gives the network input at time $t$ the covariance $\Sigma_t = (1-t)^2\mathbf{I} + t^2\mathbf{H}$, which moves from the identity at $t$=0 (noise) to the data covariance at $t$=1 (data). For a linear regression on these inputs, the condition number of $\Sigma_t$ sets the ratio of convergence rates across directions (Ahamed et al., 2026); we use it as a diagnostic. We report $\tilde{\kappa}_{512}(\Sigma_t)$, computed from the top 512 eigenvalues rescaled to unit mean, which makes it invariant to a global scale of the features (Appendix F). In all three spaces it starts at 1 and grows with $t$ (Figure 4b); at $t$=0.9 it is 1,925 for pixels, 447 for SD-VAE and 55.6 for DINOv2-B, a 35× gap between pixels and DINOv2.

**Marginal kurtosis.** The source is Gaussian along every coordinate. We summarize the tails of each coordinate by its excess kurtosis $\gamma_j = \mathbb{E}[(z_j - \mu_j)^4]/\sigma_j^4 - 3$, which is 0 for a Gaussian; a small value does not by itself imply Gaussianity. Of the DINOv2-B coordinates, 98.7% have $\lvert\gamma_j\rvert < 0.5$, against 74.2% for SD-VAE and none for pixels, and the median $\lvert\gamma_j\rvert$ is 11.5× lower than for pixels (Figure 4c). Joint statistics give mixed results: DINOv2's multivariate kurtosis is also closer to the Gaussian value, while its multivariate skewness is larger than that of pixels (Appendix H).

**Linear interpolation.** Flow matching is trained on linear paths. Linear interpolants between the DINOv2 features of two images decode to coherent images, whereas linear blends of the images themselves show ghosting (Figure 5 in Appendix F).

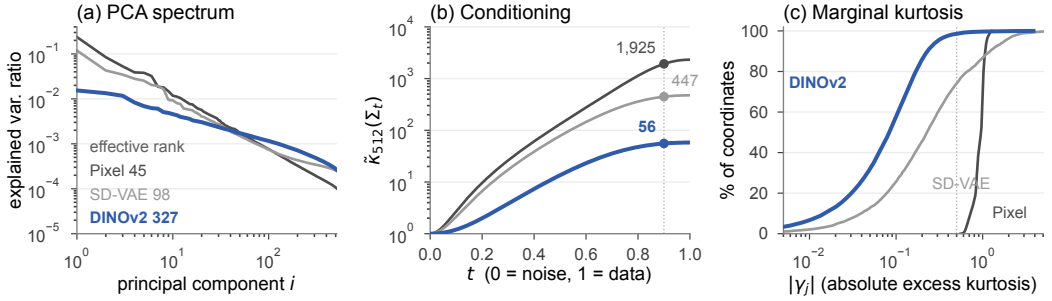



**Figure 4: Pixels, SD-VAE and DINOv2-B relative to the Gaussian source** (raw features, 10,000 ImageNet images). (a) Explained-variance ratio of the principal components. (b) Diagnostic $\tilde{\kappa}_{512}(\Sigma_t)$: condition number of $(1-t)^2\mathbf{I} + t^2\mathbf{H}$ computed from the top 512 eigenvalues of $\mathbf{H}$ rescaled to unit mean; values at $t$=0.9 are marked. (c) Cumulative distribution of the per-coordinate excess kurtosis; the dotted line is $\lvert\gamma_j\rvert$=0.5.

**Do these quantities track generation?** The spaces above differ in many respects at once. For a controlled test we use three tokenizers that share architecture, data, training budget and latent shape and differ only in the vision-foundation-model alignment loss (Table 4). Their reconstruction quality is nearly identical, while generation FID falls by 30% from no alignment to DINOv2 alignment, and effective rank and both condition numbers improve in the same order. The ordering over three tokenizers is consistent, its statistical power is limited, and the alignment loss also changes other properties of the latent, such as its semantic organization. All three remain far from DINOv2-B ($\tilde{\kappa}_{512}(\Sigma_{0.9})$ of 336 at best, against 55.6).

**The standardized space.** Element-wise standardization (§3.1) equalizes the diagonal of $\mathbf{H}$ exactly, but its effect on the spectrum is an empirical question: van der Sluis (1969) bounds the condition number after unit-diagonal scaling only relative to the best diagonal scaling, within a factor equal to the number of nonzeros per row, and this does not guarantee an improvement over the unscaled matrix. In the space RiT is trained in, it improves both covariance quantities (Table 5) and leaves the excess kurtosis unchanged, as any per-coordinate affine map must. Applying the same standardization to pixels and SD-VAE preserves the ordering of Table 3: the pixel-to-DINOv2-S gap in $\tilde{\kappa}_{512}(\Sigma_{0.9})$ is $23.7\times$ after standardization (Appendix G).

**Relation to the $x$-target.** Under a Gaussian surrogate $p(\mathbf{z}_0) \approx \mathcal{N}(\boldsymbol{\mu}, \mathbf{H})$ with $\mathbf{H} \succ 0$, the Bayes-optimal $x$-target is affine in $\mathbf{z}_t$, and the condition number of its Jacobian $t\,\mathbf{H}\Sigma_t^{-1}$ is at most $\kappa(\mathbf{H})$ at every $t$ (Appendix D). This links the covariance diagnostics to the $x$-target of §3.2; it is a property of the target map under the surrogate and does not describe the optimization dynamics.

**Summary.** At similar intrinsic-dimension estimates, DINOv2 differs from pixels in covariance spread, conditioning and marginal kurtosis; a second representation encoder, SigLIP2, falls in the same range and also trains to a strong FID (Section 4.5), and within a controlled family of tokenizers the covariance statistics order generation quality. These measurements are consistent with the results of Section 4. They do not rank two widths of one encoder (raw DINOv2-S is less well conditioned than DINOv2-B yet trains to a better FID; Section 4.4), and they do not isolate why the recipe works.

## 6 Conclusion

With element-wise standardization, $x$-prediction, a dimension-aware time shift and joint `[CLS]` diffusion, a standard DiT trained on frozen DINOv2 features needs neither a representation-specific head nor non-Euclidean transport. RiT-XL reaches FID 1.45 without guidance and 1.14 with guidance on ImageNet $256\times256$, samples at FID 2.0 with 5 Heun steps, and transfers to SigLIP2. Our geometric diagnostics show how DINOv2 differs from pixels at similar intrinsic-dimension estimates, in effective rank, conditioning and marginal kurtosis; they are consistent with the empirical results but do not isolate why the recipe works (Appendix A). Our experiments are limited to class-conditional ImageNet at $256\times256$, frozen encoders and denoisers wider than the token dimension. Whether such diagnostics can rank candidate representation spaces before a generator is trained in them remains open.

## AI Use Statement

In this work, we used generative AI tools (large language models) for the following tasks with required disclosure: implementing methods and analysis code; providing feedback on the research methodology and experiments, including reviews of the evaluation and analysis protocols and the suggestion of an ablation; refining how the hypotheses and claims of the paper are stated; assisting in formulating and writing the derivations of Appendices D and E; and interpreting results. We have not used generative AI tools to generate synthetic data sets or to clean or reformat data sets, and translation and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for the following tasks with recommended disclosure: creating and editing software code, creating and modifying scientific figures, drafting parts of the paper, editing the paper to improve readability and grammar, suggesting a structure for the paper, and searching for information in existing code and documentation. We have reviewed all AI-assisted work. LLM-assisted code was checked and tested by the authors, the derivations were checked by the authors, every numerical result reported in the paper comes from experiments run by the authors, and every

edit to the text was read and approved by the authors. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

## Reproducibility Statement

Appendix J lists the architecture, all training and sampling hyperparameters, and PyTorch-style pseudocode for training and sampling. The encoder and decoder are the released RAE checkpoints (Zheng et al., 2025), used frozen; the standardization statistics are computed on the ImageNet training set as described in Section 3.1. Appendices F–H specify each geometric measurement, including sample sizes, PCA budget and estimators, and Appendix M defines every sampling schedule. We will publicly release our code and the trained model weights.

## References

Shadab Ahamed, Eshed Gal, Simon Ghyselincks, Md Shahriar Rahim Siddiqui, Moshe Eliasof, and Eldad Haber. Preconditioned score and flow matching. *arXiv preprint arXiv:2603.02337*, 2026.

Alan Baade, Eric Ryan Chan, Kyle Sargent, Changan Chen, Justin Johnson, Ehsan Adeli, and Li Fei-Fei. Latent forcing: Reordering the diffusion trajectory for pixel-space image generation. *arXiv preprint arXiv:2602.11401*, 2026.

Hao Chen, Yujin Han, Fangyi Chen, Xiang Li, Yidong Wang, Jindong Wang, Ze Wang, Zicheng Liu, Difan Zou, and Bhiksha Raj. Masked autoencoders are effective tokenizers for diffusion models. In *Forty-second International Conference on Machine Learning*, 2025a.

Ricky TQ Chen and Yaron Lipman. Flow matching on general geometries. *arXiv preprint arXiv:2302.03660*, 2023.

Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. Pixelflow: Pixel-space generative models with flow. *arXiv preprint arXiv:2504.07963*, 2025b.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *NeurIPS*, 2021.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific reports*, 7(1):12140, 2017.

Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Mdtv2: Masked diffusion transformer is a strong image synthesizer. *arXiv preprint arXiv:2303.14389*, 2023.

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

Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. Query-key normalization for transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4246–4253, 2020.

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pp. 13213–13232. PMLR, 2023.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. *Advances in Neural Information Processing Systems*, 37:52996–53021, 2024.

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.

Theodoros Kouzelis, Efstathios Karypidis, Ioannis Kakogeorgiou, Spyridon Gidaris, and Nikos Komodakis. Boosting generative image modeling via joint image-feature synthesis. *Advances in Neural Information Processing Systems*, 38:16685–16714, 2026.

Amandeep Kumar and Vishal M Patel. Learning on the manifold: Unlocking standard diffusion transformers with representation encoders. *arXiv preprint arXiv:2602.10099*, 2026.

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

Elizaveta Levina and Peter Bickel. Maximum likelihood estimation of intrinsic dimension. *Advances in neural information processing systems*, 17, 2004.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

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*, pp. 23–40. Springer, 2024.

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. *Transactions on Machine Learning Research*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023. URL https://arxiv.org/abs/2212.09748.

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*, pp. 10684–10695, 2022.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*, pp. 606–610. IEEE, 2007.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

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.

Ivan Skorokhodov, Sharath Girish, Benran Hu, Willi Menapace, Yanyu Li, Rameen Abdal, Sergey Tulyakov, and Aliaksandr Siarohin. Improving the diffusability of autoencoders. *arXiv preprint arXiv:2502.14831*, 2025.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. 2023.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Shengbang Tong, Boyang Zheng, Ziteng Wang, Bingda Tang, Nanye Ma, Ellis Brown, Jihan Yang, Rob Fergus, Yann LeCun, and Saining Xie. Scaling text-to-image diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2601.16208*, 2026.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

A. van der Sluis. Condition numbers and equilibration of matrices. *Numerische Mathematik*, 14(1): 14–23, 1969.

Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. Pixnerd: Pixel neural field diffusion. *arXiv preprint arXiv:2507.23268*, 2025a.

Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. Ddt: Decoupled diffusion transformer, 2025b.

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.

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*, pp. 15703–15712, 2025.

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.

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, Anima Anandkumar, and Arash Vahdat. Pixeldit: Pixel diffusion transformers for image generation. *arXiv preprint arXiv:2511.20645*, 2025.

Zhengrong Yue, Haiyu Zhang, Xiangyu Zeng, Boyu Chen, Chenting Wang, Shaobin Zhuang, Lu Dong, Yi Wang, Limin Wang, and Yali Wang. Uniflow: A unified pixel flow tokenizer for visual understanding and generation. In *International Conference on Learning Representations*, volume 2026, pp. 84741–84772, 2026.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

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

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

## A LIMITATIONS

**Encoder bias.** RiT inherits the inductive biases of the frozen DINOv2 encoder. DINOv2's self-supervised objective emphasizes semantic content over photometric detail, and details that the encoder represents poorly, such as fine textures, thin structures and small objects, may also be generated less faithfully, since the decoder operates on the same features. Fine-tuning the encoder could mitigate them at the cost of the frozen-encoder setup.

**Scope.** All RiT results are class-conditional on ImageNet at $256\times256$. We have not evaluated text-to-image generation, higher resolutions or other modalities. All denoisers we train are much wider than the token dimension (width 1152 against $d$=384 for DINOv2-S). We have not tested denoisers whose width is close to the token dimension, the regime in which RAE motivates its wide head and in which Tong et al. (2026) find its largest gain, nor whether adding a DDT head to RiT improves it. We applied the time grids of Appendix M to RiT only; baselines in the few-step comparison use their own samplers. The source of the `[CLS]` gain (the semantic content of the token, the additional token, or the auxiliary loss) has not been separated. The geometric analysis of Section 5 uses ImageNet images only; whether the same geometry holds at larger scale or under text conditioning is untested.

**Analysis.** The covariance $\Sigma_t$ is exact, but its condition number measures regression difficulty only for a linear model, and our diagnostic uses a truncated, rescaled spectrum. The Jacobian statement of Appendix D holds under a Gaussian surrogate. All quantities are aggregate scalars that cannot exclude regions of the data distribution where the favorable properties fail. The generative results agree with the geometric analysis but do not establish any single quantity as necessary for them.

## B COMPARISON WITH THE RAE RECIPE

Table 6 lists how RiT-XL differs from the RAE recipe. RAE has released the configuration of DiT$^{\text{DH}}$-XL on DINOv2-B, which the RAE column follows; a configuration for RAE-XL on DINOv2-S has not been released, and we do not assume that it matches in every entry. The released RAE configuration integrates with 49 Euler steps (50 time points), and RiT with 24 Heun steps followed by a final Euler step; both use 49 function evaluations.

**Table 6: RAE and RiT recipes.** RAE column: released configuration of DiT$^{\text{DH}}$-XL on DINOv2-B.

| | RAE (DiT$^{\text{DH}}$-XL) | RiT-XL |
|---|---|---|
| Encoder | DINOv2-B with registers, frozen | DINOv2-S with registers, frozen |
| Output head | DDT head (2 layers, width 2048) | linear |
| Backbone blocks | SwiGLU, RMSNorm, 2D RoPE, fixed sin-cos position embedding | SwiGLU, RMSNorm, 2D RoPE, fixed sin-cos position embedding, QK-normalization, 32 in-context class tokens |
| Prediction | velocity | clean feature ($x$) |
| Loss | velocity MSE | velocity MSE with $a(t) = \max(1-t, 0.05)$ |
| Time distribution | logit-normal$(0, 1)$ with dimension shift | logit-normal$(0, 1)$, same shift rule |
| `[CLS]` token | not used | diffused jointly, $\lambda$=0.2 |
| Optimizer | AdamW, $\beta$=$(0.9, 0.95)$, lr $2\times10^{-4}$ for 40 epochs, then linear decay to $2\times10^{-5}$ at epoch 800 | AdamW, $\beta$=$(0.9, 0.95)$, lr $3\times10^{-4}$ after a 5-epoch warmup, constant |
| Batch size, EMA | 1024, 0.9995 | 1536, 0.9999 |
| Sampler | Euler, 49 steps, shifted time grid | Heun, 25 steps, time-shift grid |
| Guidance | AutoGuidance (scale 1.42); otherwise none | CFG (scale 3.7, $t \in [0.1, 0.98]$) |

Within our code (Table 2), $v$-prediction without `[CLS]` reaches 1.78, $x$-prediction without `[CLS]` 1.63, and the full recipe 1.44. We did not train RiT's backbone with RAE's optimizer and sampler, so the difference between RAE-XL's 1.87 and our $v$-prediction run is not attributed to a single entry of Table 6.

## C Equivalence of Velocity Loss and Reweighted $x$-Prediction Loss

We show that the velocity MSE under the $x$-prediction parameterization is a reweighted $x$-prediction loss. Given the forward process $\mathbf{z}_t = t\,\mathbf{z}_0 + (1-t)\,\boldsymbol{\epsilon}$, the target velocity is

$$\mathbf{v} = \mathbf{z}_0 - \boldsymbol{\epsilon} = \frac{\mathbf{z}_0 - \mathbf{z}_t}{1-t}, \tag{2}$$

where the second equality substitutes $\boldsymbol{\epsilon} = (\mathbf{z}_t - t\,\mathbf{z}_0)/(1-t)$. Under $x$-prediction, the network outputs $\hat{\mathbf{z}}_0 = f_\theta(\mathbf{z}_t, t, c)$ and the predicted velocity is $\hat{\mathbf{v}}_\theta = (\hat{\mathbf{z}}_0 - \mathbf{z}_t)/(1-t)$. Substituting both into the velocity loss gives

$$\mathcal{L}_{\text{fm}} = \mathbb{E}_{t,\mathbf{z}_0,\boldsymbol{\epsilon}} \left\|\hat{\mathbf{v}}_\theta - \mathbf{v}\right\|^2 = \mathbb{E}_{t,\mathbf{z}_0,\boldsymbol{\epsilon}} \left\| \frac{\hat{\mathbf{z}}_0 - \mathbf{z}_t}{1-t} - \frac{\mathbf{z}_0 - \mathbf{z}_t}{1-t} \right\|^2 = \mathbb{E}_{t,\mathbf{z}_0,\boldsymbol{\epsilon}} \frac{1}{(1-t)^2} \left\|\hat{\mathbf{z}}_0 - \mathbf{z}_0\right\|^2. \tag{3}$$

The velocity loss is therefore the $x$-prediction loss reweighted by $(1-t)^{-2}$, which upweights states near clean data. In the implementation the denominator is clamped, $1-t \to \max(1-t,\, 0.05)$, in both the prediction and the target, so the weight is capped at 400 and, for $t > 0.95$, the target differs from $\mathbf{z}_0 - \boldsymbol{\epsilon}$ by the factor $(1-t)/0.05$. Under the shifted training distribution $P(t > 0.95) \approx 3 \times 10^{-6}$, and the default 25-step time-shift grid evaluates the network only at $t \le 0.831$, so the clamp is almost never active during training and is not active during default sampling. The $v$-prediction runs of Table 2 regress this same clamped target with the network output directly, so the $x$/$v$ comparison differs only in the parameterization. Matching the objectives makes the comparison fair; it does not make the clamped target equal to the velocity of the linear path for $t > 0.95$. The analysis in the main text uses the unclamped forms. The velocity loss and the reweighted $x$-loss are the same objective; what differs between $x$- and $v$-prediction is the function the network is asked to represent (§3.2). Under $x$-prediction the network output is $\hat{\mathbf{z}}_0$, an estimate of an encoder output, and the $-\mathbf{z}_t/(1-t)$ term is applied analytically; under $v$-prediction the network must itself produce $(\mathbf{z}_0 - \mathbf{z}_t)/(1-t)$, including its noise component.

## D Conditioning of the $x$-Target under a Gaussian Surrogate

This appendix derives the conditioning statement of Section 5 ("Relation to the $x$-target") and a companion statement about the $v$-target. Take a Gaussian surrogate $p(\mathbf{z}_0) \approx \mathcal{N}(\boldsymbol{\mu}, \mathbf{H})$ with $\mathbf{H} \succ 0$ (if $\mathbf{H}$ is singular, the statements below hold on its range), and the forward process $\mathbf{z}_t = t\,\mathbf{z}_0 + (1-t)\,\boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Then $(\mathbf{z}_0, \mathbf{z}_t)$ is jointly Gaussian with $\mathrm{Cov}(\mathbf{z}_t) = \Sigma_t = (1-t)^2\mathbf{I} + t^2\mathbf{H}$ and $\mathrm{Cov}(\mathbf{z}_0, \mathbf{z}_t) = t\,\mathbf{H}$, so the posterior mean is affine:

$$\mathbb{E}[\mathbf{z}_0 \mid \mathbf{z}_t] \;=\; \boldsymbol{\mu} + t\,\mathbf{H}\,\Sigma_t^{-1}\,(\mathbf{z}_t - t\boldsymbol{\mu}), \qquad \mathbf{J}_x \;\equiv\; \frac{\partial\,\mathbb{E}[\mathbf{z}_0 \mid \mathbf{z}_t]}{\partial \mathbf{z}_t} \;=\; t\,\mathbf{H}\,\Sigma_t^{-1}. \tag{4}$$

$\mathbf{H}$ and $\Sigma_t$ share eigenvectors, so in the eigenbasis of $\mathbf{H}$ with eigenvalues $\lambda_i$ the Jacobian is diagonal with entries

$$(\mathbf{J}_x)_{ii} \;=\; \frac{t\,\lambda_i}{(1-t)^2 + t^2\lambda_i}. \tag{5}$$

These entries increase with $\lambda_i$, so the extreme entries come from $\lambda_{\max}$ and $\lambda_{\min}$ and

$$\kappa(\mathbf{J}_x) \;=\; \frac{\lambda_{\max}}{\lambda_{\min}} \cdot \frac{(1-t)^2 + t^2\lambda_{\min}}{(1-t)^2 + t^2\lambda_{\max}} \;=\; \kappa(\mathbf{H}) \cdot \frac{(1-t)^2 + t^2\lambda_{\min}}{(1-t)^2 + t^2\lambda_{\max}}. \tag{6}$$

The second factor is below 1 for every $t \in (0,1)$ whenever $\lambda_{\max} > \lambda_{\min}$ and tends to $\lambda_{\min}/\lambda_{\max}$ as $t \to 1$, where $\kappa(\mathbf{J}_x) \to 1$. Under the surrogate, the map an $x$-predictor fits is therefore never worse conditioned than the data covariance, and $\kappa(\mathbf{H})$ bounds its condition number at every noise level. The model is class-conditional, so the relevant target is $\mathbb{E}[\mathbf{z}_0 \mid \mathbf{z}_t, c]$; the same derivation applies with a class-conditional surrogate $\mathcal{N}(\boldsymbol{\mu}_c, \mathbf{H}_c)$, and our diagnostics use the unconditional covariance $\mathbf{H}$.

The $v$-target is $\mathbf{v} = \mathbf{z}_0 - \boldsymbol{\epsilon} = (\mathbf{z}_0 - \mathbf{z}_t)/(1-t)$, so $\mathbb{E}[\mathbf{v} \mid \mathbf{z}_t]$ has Jacobian $\mathbf{J}_v = (\mathbf{J}_x - \mathbf{I})/(1-t)$ with entries

$$(\mathbf{J}_v)_{ii} \;=\; \frac{t\lambda_i - (1-t)}{(1-t)^2 + t^2\lambda_i}, \tag{7}$$

which vanish at $t_i^* = 1/(1+\lambda_i)$, at a different point of the schedule for each eigendirection. A vanishing eigenvalue of the target map does not by itself make the regression harder, and we do not use this property as evidence about optimization. Both statements are exact under the surrogate. How well the surrogate fits is mixed: DINOv2 is closer to the Gaussian value on Mardia's multivariate kurtosis, but not on multivariate skewness (Appendix H).

## E Norm of Intermediate States and Composition of the Targets

Let $\mathbf{z}_0$ and $\boldsymbol{\epsilon}$ be independent with $\mathbb{E}\|\mathbf{z}_0\|^2 = \mathbb{E}\|\boldsymbol{\epsilon}\|^2 = m$. This holds per token in raw DINOv2 space, where LayerNorm without affine parameters gives $\|\mathbf{z}_0\|^2 = d$ for every token (up to the LayerNorm $\epsilon$) and $\mathbb{E}\|\boldsymbol{\epsilon}\|^2 = d$, and, up to the standardization $\epsilon$, for the full feature map in the standardized space, where both equal $D$. Since $\boldsymbol{\epsilon}$ has zero mean and is independent of $\mathbf{z}_0$, the cross term vanishes and

$$\mathbb{E}\|\mathbf{z}_t\|^2 = t^2\,\mathbb{E}\|\mathbf{z}_0\|^2 + (1-t)^2\,\mathbb{E}\|\boldsymbol{\epsilon}\|^2 = \big(t^2 + (1-t)^2\big)\,m, \tag{8}$$

which equals $m$ only at the endpoints and $m/2$ at $t = \frac{1}{2}$. In raw space every encoder output token has norm $\sqrt{d}$, and the noise norm concentrates at $\sqrt{d}$ with relative fluctuation of order $1/\sqrt{2d}$ (about $4\%$ for $d$=384). The squared norm of an intermediate token concentrates in the same way around $(t^2 + (1-t)^2)\,d$, so away from the endpoints an intermediate token typically has a norm below $\sqrt{d}$, the norm of every encoder output token.

The two regression targets decompose as follows; we write $\mathbb{E}[\,\cdot \mid \mathbf{z}_t]$ for $\mathbb{E}[\,\cdot \mid \mathbf{z}_t, t, c]$. The $x$-target is $\mathbf{z}_0$ itself. The $v$-target is $\mathbf{v} = \mathbf{z}_0 - \boldsymbol{\epsilon}$ with $\mathbb{E}\|\mathbf{v}\|^2 = 2m$, of which $m$ is the noise vector. Conditioning on $\mathbf{z}_t$, $\mathbb{E}[\mathbf{v} \mid \mathbf{z}_t] = \mathbb{E}[\mathbf{z}_0 \mid \mathbf{z}_t] - \mathbb{E}[\boldsymbol{\epsilon} \mid \mathbf{z}_t]$, and since $\mathbb{E}[\boldsymbol{\epsilon} \mid \mathbf{z}_t] = (\mathbf{z}_t - t\,\mathbb{E}[\mathbf{z}_0 \mid \mathbf{z}_t])/(1-t)$, the second term is determined by the first and by $\mathbf{z}_t$. An $x$-predictor learns only $\mathbb{E}[\mathbf{z}_0 \mid \mathbf{z}_t]$ and obtains the rest in closed form; a $v$-predictor learns the difference of the two terms as a single function.

## F Details of the Geometric and Sampling Measurements

All geometric measurements use 10,000 ImageNet training images drawn uniformly without replacement and center-cropped to 256×256. Table 3 (except the SigLIP2 column) and Figure 4 use one random draw; the SigLIP2 column, Table 4, Table 5 and the measurements of Appendices G and H use a second draw from a separate run. SD-VAE latents are posterior samples and the latents of the tokenizers in Table 4 are posterior modes.

**PCA spectrum, effective rank and conditioning.** We fit PCA with 512 components (centered, randomized SVD solver) on the flattened feature vectors of each space. Let $\lambda_1 \geq \cdots \geq \lambda_k$ be the retained eigenvalues of the sample covariance and $\hat{\lambda}_i = \lambda_i / \sum_j \lambda_j$. The effective rank (Roy & Vetterli, 2007) is

$$\text{erank} = \exp\Big(-\sum_{i=1}^{k} \hat{\lambda}_i \log \hat{\lambda}_i\Big), \tag{9}$$

which equals 1 when all variance lies in one direction and $k$ when variance is uniform over the retained directions. For the conditioning diagnostic we rescale the retained eigenvalues to unit mean, $\bar{\lambda}_i = \lambda_i / (\frac{1}{k}\sum_j \lambda_j)$, and report

$$\tilde{\kappa}_k(\Sigma_t) = \frac{\max_i \big((1-t)^2 + t^2\bar{\lambda}_i\big)}{\min_i \big((1-t)^2 + t^2\bar{\lambda}_i\big)}, \qquad k = 512. \tag{10}$$

This is a truncated, normalized diagnostic and differs from $\kappa\big((1-t)^2\mathbf{I} + t^2\mathbf{H}\big)$ for the full covariance: the rescaling removes the dependence on a global scale of the features, which the untruncated quantity has at fixed $t$, and the truncation discards the smallest eigenvalues. Table 3 and Figure 4 evaluate $\tilde{\kappa}$ at $t$=0.5 and 0.9 exactly; the measurements from the second run evaluate it on a 200-point grid, at $t$=100/199 and 179/199. On the pixel and DINOv2-B spectra of Table 3, the grid changes $\tilde{\kappa}_{512}(\Sigma_{0.9})$ by 0.2% and $\tilde{\kappa}_{512}(\Sigma_{0.5})$ by at most 2%.

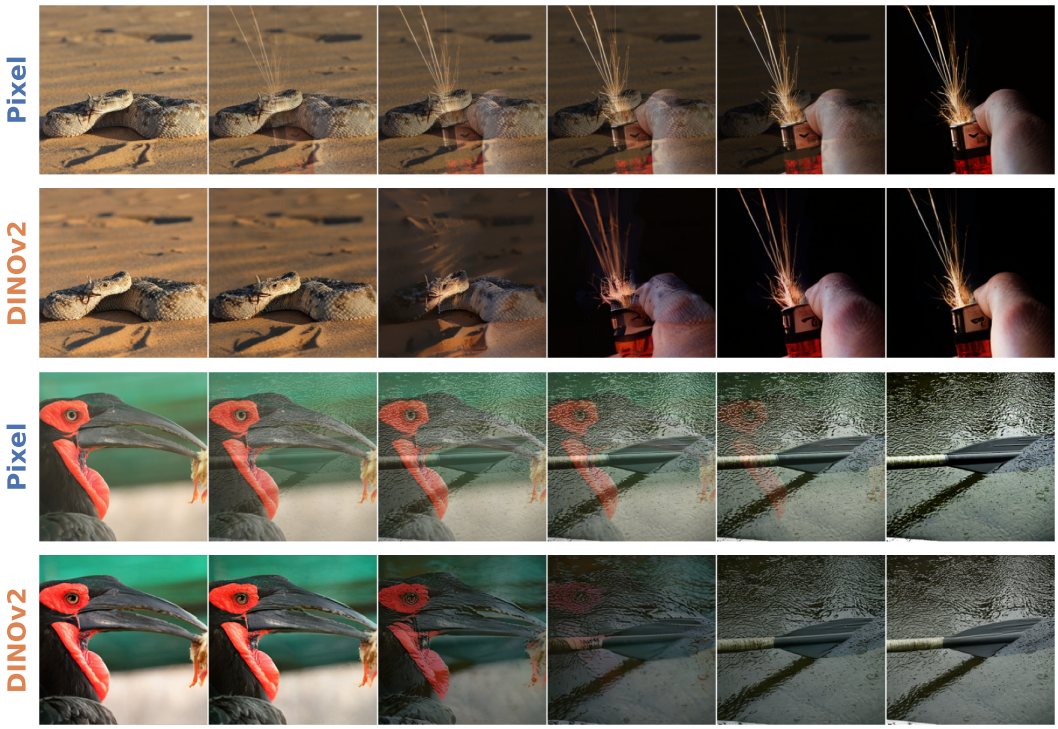



**Figure 5: Cross-class interpolation.** Top row of each pair: linear blending of two ImageNet training images in pixel space. Bottom row: linear interpolation of their DINOv2-S patch features, decoded by the RAE decoder; the endpoints of this row are RAE reconstructions.

**Intrinsic dimensionality (TwoNN).** The TwoNN estimator (Facco et al., 2017) estimates intrinsic dimensionality from the ratio of first- and second-nearest-neighbor distances. For each point $\mathbf{x}_i$, let $r_1^{(i)}$ and $r_2^{(i)}$ be the distances to its nearest and second-nearest neighbors and $\mu_i = r_2^{(i)}/r_1^{(i)}$. Under local uniformity on a $d$-dimensional manifold, the maximum-likelihood estimate is

$$\hat{d} = \frac{N}{\sum_{i=1}^{N} \log \mu_i}, \tag{11}$$

where $N$ is the number of samples with $\mu_i > 1$. Pairwise Euclidean distances are computed in chunks to control memory. Each estimate uses a random subsample of 5,000 points.

**Robustness of $\hat{d}$ at $D{\sim}10^5$.** At this ambient dimension nearest-neighbor distances concentrate, and a single estimate can be noisy. We repeat TwoNN on 10 random subsamples of 5,000 points, each drawn without replacement:

| Space | $\hat{d}$ (mean $\pm$ std over 10 subsamples) |
|---|---|
| Pixel | $33.6 \pm 1.3$ |
| DINOv2-B | $32.6 \pm 0.8$ |

The pixel–DINOv2 gap of 1.0 is below the combined standard deviation $\sqrt{1.3^2 + 0.8^2} \approx 1.5$, so the subsamples do not separate the two estimates; this is not an equivalence test, and similar estimates do not imply that the two distributions have the same local structure. TwoNN requires the density to be locally uniform only up to the second-neighbor distance, which reduces the effect of curvature and density variation relative to estimators that use larger neighborhoods, such as the $k$-nearest-neighbor maximum-likelihood estimator (Levina & Bickel, 2004; Facco et al., 2017); we therefore use TwoNN as the primary estimator. We compute the maximum-likelihood estimate under the TwoNN model given above, rather than the least-squares fit to the empirical distribution function used by Facco

et al. (2017). On the single subsample used for Table 3, the estimator gives 35.3 for pixels, 33.5 for DINOv2-B and 70.9 for SD-VAE.

**Marginal Gaussianity.** For each coordinate $j$, the excess kurtosis is

$$\gamma_j = \frac{\mathbb{E}[(z_j - \mu_j)^4]}{\sigma_j^4} - 3, \tag{12}$$

with $\mu_j$ and $\sigma_j$ the per-coordinate mean and standard deviation. A Gaussian has $\gamma_j = 0$; positive values indicate heavier tails. We summarize by the median of $|\gamma_j|$ over coordinates and by the fraction with $|\gamma_j| < 0.5$.

**Pixel-space endpoint discrepancy.** For RiT-XL and pixel-space JiT-H we draw 128 noise–label pairs per model and integrate each pair without guidance with $K \in \{2, 5, 10, 25, 50\}$ Heun steps and with a reference $K{=}125$, reusing the same initial noise for all $K$; RiT uses the time-shift grid and coupled `[CLS]` noise, and JiT the uniform grid of its released code. We measure the Frobenius distance $\|x^{(K)} - x^{(125)}\|_F$ between the resulting images (Figure 6). Each model is compared with its own reference. For RiT the images pass through the frozen decoder, which JiT does not have, so the measurement reflects how much the final image changes with $K$, including the effect of the decoder, rather than the latent ODE error alone; we do not interpret the decay in $K$ as an asymptotic convergence order, since the reference itself uses a finite number of steps. RiT's discrepancy falls $12.9\times$ from $K{=}2$ to $K{=}50$, against $3.6\times$ for JiT; it is larger than JiT's at $K \leq 5$, smaller from $K{=}10$ on, and $2.7\times$ smaller at $K{=}50$.

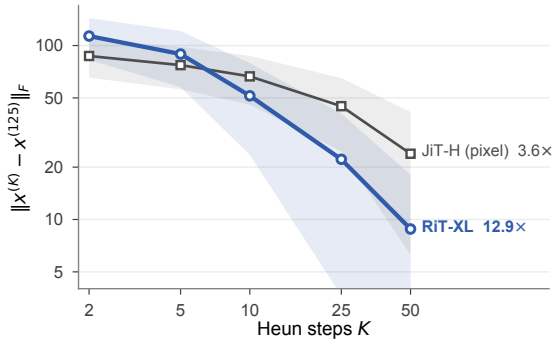



**Figure 6: Pixel-space endpoint discrepancy** between $K$-step and 125-step Heun samples from the same initial noise within each model (mean $\pm 1\sigma$ over 128 trajectories; the band is clipped at the bottom of the axis). Numbers at the right give the reduction from $K{=}2$ to $K{=}50$.

## G Geometry in the Standardized Space, and the PCA Budget

Section 5 measures DINOv2-B so that the ambient dimension matches pixels. RiT uses DINOv2-S and is trained on standardized features; this appendix reports the same quantities in that setting, on 10,000 ImageNet images at the same PCA budget $k{=}512$.

| Quantity | DINOv2-S (raw) | DINOv2-S (standardized) | DINOv2-B (raw) |
|---|---|---|---|
| Effective rank | 252.1 | 289.4 | 326.6 |
| $\tilde{\kappa}_{512}(\Sigma_t)$ at $t{=}0.9$ | 143.4 | 81.1 | 55.6 |
| $\tilde{\kappa}_{512}(\Sigma_t)$ at $t{=}0.5$ | 22.0 | 14.8 | 13.6 |
| Median $\|\gamma_j\|$ | 0.0879 | 0.0879 | 0.083 |
| TwoNN $\hat{d}$ | 30.3 | 35.1 | 33.5 |

The DINOv2-B column is taken from Table 3, which uses a different draw of 10,000 images. Excess kurtosis is unchanged up to floating-point error (the medians differ by $1.2 \times 10^{-8}$), as it must be

under a per-coordinate affine map. Effective rank rises and conditioning improves at both times. Raw DINOv2-S is less well conditioned than raw DINOv2-B yet trains to a better FID (Section 4.4); Section 5 uses these quantities to compare spaces produced by different training objectives; for two widths of one encoder they do not predict the FID ordering (Appendix K).

**Matched standardization across spaces.** Section 5 compares the spaces as their encoders produce them. The table below applies element-wise standardization to all three spaces, with the training-set statistics that RiT uses for DINOv2-S and with statistics fitted on the same 10,000 images for pixels and SD-VAE. Its raw values come from the second draw of images and differ slightly from Table 3; the ratios below use this table only.

| | Effective rank | | $\tilde{\kappa}_{512}(\Sigma_t)$ at $t$=0.9 | | $\tilde{\kappa}_{512}(\Sigma_t)$ at $t$=0.5 | |
|---|---|---|---|---|---|---|
| Space | raw | std. | raw | std. | raw | std. |
| Pixel | 43.7 | 45.0 | 1995.5 | 1921.6 | 138.0 | 134.7 |
| SD-VAE | 96.4 | 104.9 | 466.3 | 384.9 | 81.9 | 71.9 |
| DINOv2-S | 252.1 | **289.4** | 143.4 | **81.1** | 22.0 | **14.8** |

Standardization lowers $\tilde{\kappa}_{512}(\Sigma_{0.9})$ by 3.7% on pixels, 17% on SD-VAE and 43% on DINOv2-S. The ordering of the three spaces is unchanged, and the pixel-to-DINOv2-S gap after standardization is 23.7× (1921.6 against 81.1); the 35× of Section 5 compares raw pixels with raw DINOv2-B. Median excess kurtosis is identical before and after standardization in all three spaces.

**Sensitivity to the PCA budget.** Effective rank and $\tilde{\kappa}_k(\Sigma_t)$ are computed from the top-$k$ eigenvalues, so both depend on $k$, and values at different budgets are not comparable. All numbers in this paper use $k$=512. The pixel-to-DINOv2 effective-rank ratio increases with the budget over the values we tested and has not stabilized at $k$=512; the conditioning ratio is 32× at $k$=256 and 35× from $k$=384 on.

| $k$ | 128 | 256 | 384 | 512 |
|---|---|---|---|---|
| Effective rank, DINOv2-B / pixel | 3.8× | 5.4× | 6.4× | 7.3× |
| $\tilde{\kappa}_k(\Sigma_t)$ at $t$=0.9, pixel / DINOv2-B | 24× | 32× | 35× | 35× |

## H Other Encoders and Tokenizers

This appendix asks whether the pattern of Section 5 is particular to DINOv2, and whether the quantities track generation quality when everything else is held fixed. Both measurements use encoders only.

**A second representation encoder.** SigLIP2 (Tschannen et al., 2025) is trained with a contrastive image–text objective, unrelated to DINOv2's self-distillation. Measured in the same way on the second draw of images ($D = 1152 \times 16 \times 16$, raw features, $k$=512), it falls in DINOv2's range and far from pixels: effective rank 316.8 (7.0× pixels, 97% of DINOv2-B), $\tilde{\kappa}_{512}(\Sigma_{0.9})$ of 63.9 (30× lower than pixels, within 1.15× of DINOv2-B), median $|\gamma_j| = 0.093$ with 95.9% of coordinates below 0.5, and TwoNN $\hat{d} = 80.4$. The mean $|\gamma_j|$ is 0.495, higher than the median because a small fraction of coordinates is heavy-tailed. Section 4.5 trains a generative model in this space.

**Controlled tokenizers.** Table 4 reports the comparison. Effective rank is comparable within this trio because the latent shape is fixed, and it is not comparable with the spaces of Table 3, which have different ambient dimensions. Median $|\gamma_j|$ is 0.199, 0.330 and 0.167 for the three tokenizers and does not follow the generation ordering.

**Is the Gaussian surrogate supported?** Marginal Gaussianity is a per-coordinate statement, while the Gaussian surrogate of Section 5 concerns the joint distribution, so we measure the joint. In the top-64 PCA subspace of each space (10,000 images) we compute Mardia's multivariate kurtosis $b_2$, normalized by its Gaussian value $k(k+2)$, and Mardia's multivariate skewness $b_1$, computed on a

random subset of 2,000 images, whose population value under a Gaussian is 0; on Gaussian data of the same dimensions the same computation gives $b_1 \approx 141$ and $b_2/k(k{+}2) \approx 1.00$. We also test how affine the conditional mean is: at $t \in \{0.1, 0.25, 0.5, 0.7, 0.9\}$ we fit a ridge map and an MLP with two hidden layers of width 1024 from $\mathbf{z}_t$ to $\mathbf{z}_0$ on an 80% training split and compare $R^2$ on the remaining 20%. DINOv2-S enters with raw features.

| Space | $b_2/k(k{+}2)$ | $b_1$ | Median $\lvert\gamma_j\rvert$ (top-64 PCA) |
|---|---|---|---|
| Pixel | 1.277 | 291.5 | 1.340 |
| SD-VAE | 1.225 | 287.3 | 1.229 |
| DINOv2-S | **1.056** | 736.8 | **0.198** |

On Mardia's kurtosis DINOv2 is the closest of the three to the Gaussian value, in the same order as conditioning. Mardia's skewness is largest for DINOv2 (736.8 against about 290, all above the Gaussian reference of about 141), so the joint distribution is not closer to Gaussian on every moment. The MLP improves on the ridge map by at most 0.036 in held-out $R^2$ (DINOv2-S at $t{=}0.1$) and is otherwise no better than it, so this diagnostic does not separate the spaces at this subspace dimension. Standardization changes the two Mardia statistics by at most 3.3% (DINOv2-S $b_1$: 736.8 to 760.6).

**Granularity of standardization.** The same three tokenizers compare per-element and per-channel standardization statistics. $\tilde{\kappa}_{512}(\Sigma_{0.9})$ is 402.1 per-channel against 392.3 per-element for the unaligned tokenizer, 377.4 against 318.9 for the MAE-aligned tokenizer, and 344.6 against 316.0 for the DINOv2-aligned one. Per-element is better in all three cases, and per-channel does not always improve on no standardization: on the DINOv2-aligned tokenizer it raises $\tilde{\kappa}_{512}$ from 336.3 to 344.6.

## I Noise Schedule

Figure 7 compares three time distributions. *JiT's schedule* samples $\mathrm{logit}(t) \sim \mathcal{N}(-0.8, 0.8^2)$, with median $t \approx 0.31$; it is the "JiT time distribution" row of Table 2. That row therefore changes both the base distribution and the shift. The *unshifted* distribution, shown for reference and not used to train any model, samples $\mathrm{logit}(X) \sim \mathcal{N}(0, 1)$ truncated to $X \in [10^{-3}, 1 - 10^{-3}]$ and sets $t = 1 - X$. RiT's *shifted* distribution applies $X' = sX/(1 + (s{-}1)X)$ to the same truncated $X$, with $s = \sqrt{16 \cdot 16 \cdot d/4096}$ ($\approx 4.9$ for $d{=}384$), and sets $t = 1 - X'$, which lowers the median $t$ to $\approx$0.17. Figure 7 shows the three densities (top) and the per-coordinate signal-to-noise ratio $(t/(1{-}t))^2$ of the interpolant, which does not depend on $d$, with the median $t$ of each distribution marked (bottom).

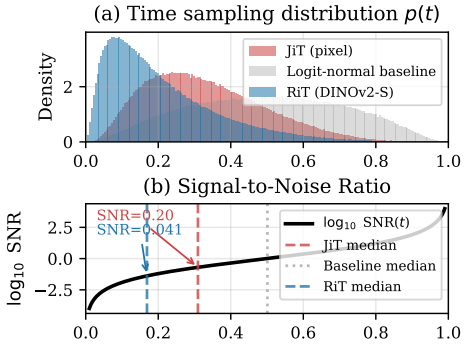



**Figure 7: Time sampling density $p(t)$ (top) and per-coordinate SNR $(t/(1{-}t))^2$ with the median $t$ of each distribution (bottom).**

# J Architecture and Hyperparameters

## J.1 Model Architecture

RiT uses the modernized DiT backbone of JiT (Li & He, 2025) and LightningDiT (Yao et al., 2025), operating on the $16\times16$ grid of DINOv2 features. The main experiments use DINOv2-S ($d$=384); the encoder ablation uses DINOv2-B ($d$=768). Table 7 summarizes the model variants, and Tables 8 and 9 list the training and sampling hyperparameters.

**Table 7: RiT model variants.** All models use patch size 1, a SwiGLU FFN with hidden width $\lfloor \frac{2}{3} \cdot 4 \cdot \text{width} \rfloor$, QK-normalization and 2D RoPE. The input is $384 \times 16 \times 16$ for DINOv2-S or $768 \times 16 \times 16$ for DINOv2-B.

| Model | Layers | Hidden dim | Heads | FFN dim | Params |
|---|---|---|---|---|---|
| RiT-L | 24 | 1024 | 16 | 2730 | 459M |
| RiT-XL | 28 | 1152 | 16 | 3072 | 676M |

Each DiT block consists of:

1. **adaLN modulation**: timestep and class embeddings are summed ($\mathbf{c} = \text{Emb}(t) + \text{Emb}(y)$) and projected by a zero-initialized SiLU–Linear layer in each block to shift, scale and gate vectors for the attention and FFN branches.
2. **Multi-head self-attention** with QK-normalization (RMSNorm on Q and K) and 2D RoPE. The `[CLS]` token and the in-context tokens receive no rotary embedding.
3. **SwiGLU FFN**: $\text{FFN}(\mathbf{x}) = (\text{SiLU}(\mathbf{x}W_1) \odot \mathbf{x}W_3)W_2$.

The final layer applies RMSNorm modulated by its own SiLU–Linear shift and scale, and a linear projection to $d$ output channels (384 for DINOv2-S, 768 for DINOv2-B). A separate linear head predicts the `[CLS]` token. The code supports attention and projection dropout in the middle 50% of layers; the main RiT-XL run sets both rates to 0.

Following JiT (Li & He, 2025), we insert 32 *in-context tokens* before the ninth of the 28 blocks of RiT-XL. At every forward pass they are formed as copies of the class embedding plus learnable positional embeddings; they take part in self-attention in all subsequent blocks and are discarded before the final layer. Removing them (`in_context_len`=0, class conditioning still carried by adaLN) changes FID from 1.44 to 1.50 at Table 2's setting of 50 Heun steps without guidance, and from 1.14 to 1.20 at the main table's setting of 25 Heun steps with guidance. Their positional embeddings are the only parameters they add, 0.04M of 676M.

## J.2 Pseudocode

Figure 8 gives the training and sampling loops in PyTorch-style pseudocode.

# K Encoder Size Ablation

DINOv2-S ($d$=384) outperforms DINOv2-B ($d$=768) with half as many channels, reaching FID 1.44 against 1.56 at 760 epochs (Table 2). We consider three explanations: decoder quality, denoiser capacity and ambient dimensionality.

*Decoding.* RAE (Zheng et al., 2025) reports reconstruction rFID of 0.64, 0.57 and 0.59 for its noise-augmented ($\tau$=0.8) DINOv2-S, -B and -L decoders (Table 15b; 0.52, 0.49 and 0.52 without noise augmentation, Table 1c). RiT uses the released noise-augmented decoders, frozen and unchanged, so the first set of values applies to RiT. The DINOv2-B decoder reconstructs better, and the DINOv2-B model still generates worse, so reconstruction FID does not explain the gap; other decoder-related factors, such as sensitivity to generation errors, are not excluded.

*Denoiser capacity.* RiT-XL is held fixed across the two encoders, so a larger denoiser might suit the higher-dimensional target better. We have not run this experiment and make no claim about it.

*Ambient dimension.* DINOv2-B doubles the ambient dimension, $768 \times 16 \times 16$ against $384 \times 16 \times 16$, while the TwoNN estimates stay between 30 and 35 for both (Appendix G), so the intrinsic-dimension

**Table 8: Training hyperparameters** for RiT-XL on ImageNet $256^2$.

| Hyperparameter | Value |
|---|---|
| Hardware | $8 \times$ NVIDIA B200 GPUs |
| Throughput | $\sim$12 minutes per epoch |
| Optimizer | AdamW ($\beta_1$=0.9, $\beta_2$=0.95) |
| Base learning rate | $5 \times 10^{-5}$ (scaled by batch/256) |
| LR schedule | Constant (after warmup) |
| Warmup epochs | 5 |
| Weight decay | 0.0 |
| Gradient clipping | 1.0 (max $\ell_2$ norm) |
| Total epochs | 760 |
| Batch size | 1536 ($8 \times 192$ per GPU) |
| EMA decay | 0.9999 / 0.9996 (dual tracking) |
| Label dropout | 0.1 |
| Attention / projection dropout | 0.0 (main configuration) |
| Time distribution | $\text{logit}(X) \sim \mathcal{N}(0,1)$, $X \in [10^{-3}, 1-10^{-3}]$, then shift |
| Time shift $s$ | $\sqrt{16 \times 16 \times d/4096}$ ($\approx 4.9$ for $d$=384) |
| Noise scale $\sigma_\epsilon$ | 1.0 |
| Denominator clamp | $\max(1-t,\ 0.05)$ |
| `[CLS]` loss weight $\lambda$ | 0.2 |

**Table 9: Sampling hyperparameters.**

| Hyperparameter | Value |
|---|---|
| ODE solver | Heun (2nd order); the final step is an Euler step |
| Number of steps | 25 (49 function evaluations) |
| Time discretization | Time-shift ($s$=4.9) |
| `[CLS]` initial noise | Coupled: spatial mean of patch noise (variance 1/256) |
| CFG scale (patches) | 3.7 |
| CFG scale (`[CLS]`) | 3.7 |
| CFG interval | $[0.1, 0.98]$ |
| Denominator clamp | $\max(1-t,\ 0.05)$ |
| Generation precision | BF16 autocast for the denoiser; decoder in FP32 |
| EMA model | 0.9999 |

estimate does not increase noticeably while the regression target doubles in size. We have not tested whether this accounts for the gap. The conditioning quantities of Section 5 do not predict this comparison: raw DINOv2-S is the less well conditioned of the two encoders and trains to the better FID. Section 5 uses those quantities to separate spaces produced by different training objectives, and we do not apply them to width within one encoder family.

## L Samples

Figure 9 shows curated RiT-XL samples and Figure 10 a larger selection; both are selected from the 192 samples of Figure 12.

Figure 10 shows 28 of these samples (RiT-XL, DINOv2-S encoder, classifier-free guidance scale 3.7), spanning animals, food, landscapes, vehicles and plants.

## M Sampling Schedule Analysis

Table 10 lists the six ODE time discretizations evaluated in this work. Each maps a normalized step index $i/K$ ($i = 0, \ldots, K$) to a time $t_i \in [0, 1]$, where $t$=0 is pure noise and $t$=1 is clean data.

Figure 11 shows these schedules and their effect on FID. Panel (a) shows the schedule functions: uniform spaces steps evenly, power-2 and time-shift concentrate steps near $t$=0 (high noise), and EDM, cosine and log-SNR are denser near both ends, EDM most strongly near $t$=0. Panels (b)

**Training**

```
# z0: data [B,C,H,W], z_cls: [B,C]
# dit: DiT model, y: class labels
t = sample_logit_normal(B, shift=s)
eps = randn_like(z0) * sigma
zt = t * z0 + (1 - t) * eps
# CLS noise: independent during training
eps_cls = randn_like(z_cls) * sigma
zt_cls = t * z_cls + (1 - t) * eps_cls
# x-prediction: predict clean data
z0_hat, cls_hat = dit(zt, t, y, zt_cls)
# convert to velocity for loss
v = (z0 - zt) / (1 - t).clamp_min(eps_t)
v_hat = (z0_hat - zt) / (1 - t).clamp_min(
    eps_t)
v_cls = (z_cls - zt_cls) / (1 - t).clamp_min
    (eps_t)
v_cls_hat = (cls_hat - zt_cls) / (1 - t).
    clamp_min(eps_t)
loss = mse(v_hat, v) + lam * mse(v_cls_hat,
    v_cls)
loss.backward()
```

**Sampling (Euler)**

```
# y: class labels, K: number of steps
z = randn(B, C, H, W) * sigma
z_cls = z.mean(dim=(2, 3))  # coupled, var
    1/256
dt = 1.0 / K
for i in range(K):
    t = i / K
    # x-prediction -> velocity
    z0_hat, cls_hat = dit(z, t, y, z_cls)
    v = (z0_hat - z) / (1 - t).clamp_min(
     eps_t)
    v_cls = (cls_hat - z_cls) / (1 - t).
     clamp_min(eps_t)
    # classifier-free guidance
    # (omitted for clarity)
    z = z + dt * v
    z_cls = z_cls + dt * v_cls
# inverse standardization, then frozen
    decoder
x = rae_decoder(z * sqrt(var + eps) + mean)
```

**Figure 8: PyTorch-style pseudocode** for RiT training (left) and sampling (right). Sampling is shown with Euler steps on a uniform grid for clarity; all reported experiments use Heun's method with the time-shift grid. `eps_t`=0.05 is the denominator clamp.



**Figure 9: Curated RiT-XL samples on ImageNet** 256×256**.**

and (c) show FID against the number of Heun steps. EDM, power-2 and time-shift, which place the most steps at high noise, give much lower FID than the other grids at 2 and 5 steps, and at 50 steps all non-uniform schedules are within 0.04 FID of one another. Tables 11 and 12 give the full results for independent and coupled `[CLS]` noise.

## N Random Samples

Figure 12 shows 192 samples from RiT-XL: the first 8 samples generated for each of 24 hand-picked ImageNet classes, without selection within a class. Each row corresponds to one class.

## O `[CLS]`–Patch Attention Analysis

Figure 13 visualizes the attention between the `[CLS]` token and the patch tokens of one sample at seven layers and three times $t \in \{0.5, 0.7, 1.0\}$, computed with an RiT-XL model from an earlier training run.

**`[CLS]`→patch (top).** In layers 0 and 4, attention from the `[CLS]` token covers the face and body of the animal. In layer 9 it is highest on the background. In layers 13 and 18 it concentrates on the nose and mouth, and in layers 22 and 27 around the eyes.

**Patch→`[CLS]` (bottom).** Attention to the `[CLS]` token is highest on the face in layers 4, 18 and 22. In layer 13 it is highest at patches on the image border, in layer 9 at $t$=0.5 it is high over most of the

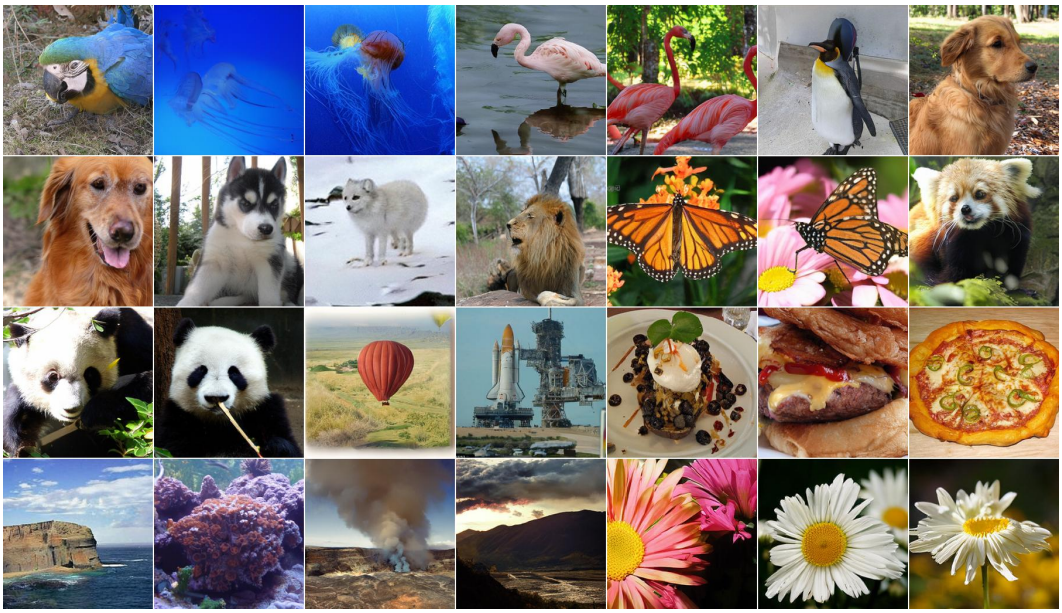


**Figure 10: Selected RiT-XL samples on ImageNet $256^2$.** 28 of the 192 samples in Figure 12; CFG scale 3.7.

**Table 10: Sampling schedule definitions.** $K$ is the number of Heun steps and $i = 0, \dots, K$. For log-SNR uniform and EDM, the endpoints are set to $t_0{=}0$ and $t_K{=}1$.

| Schedule | Formula |
|---|---|
| Uniform | $t_i = i/K$ |
| Cosine | $t_i = \frac{1}{2}\big(1 - \cos(\pi\, i/K)\big)$ |
| Log-SNR uniform | $t_i = \sigma\big(\mathrm{logit}(\epsilon) + \frac{i}{K}[\mathrm{logit}(1{-}\epsilon) - \mathrm{logit}(\epsilon)]\big), \quad \epsilon{=}10^{-3}$ |
| EDM (Karras et al., 2022) | $\sigma_i = \big(\sigma_{\max}^{1/\rho} + \frac{i}{K}(\sigma_{\min}^{1/\rho} - \sigma_{\max}^{1/\rho})\big)^{\rho}, \quad t_i = 1/(1{+}\sigma_i)$<br>$\sigma_{\min}{=}0.002,\ \sigma_{\max}{=}80,\ \rho{=}7$ |
| Power-2 | $t_i = (i/K)^2$ |
| Time-shift | $u_i = i/K, \quad t_i = u_i \,/\, \big(s - (s{-}1)\, u_i\big), \quad s{=}4.9$ |

image, and in layer 27 it is nearly uniform. Because each map is normalized by its own maximum, the figure compares spatial distributions within a map and does not compare attention strength across layers. The visualization does not establish how this attention pattern relates to the FID gain of joint diffusion in Table 2.

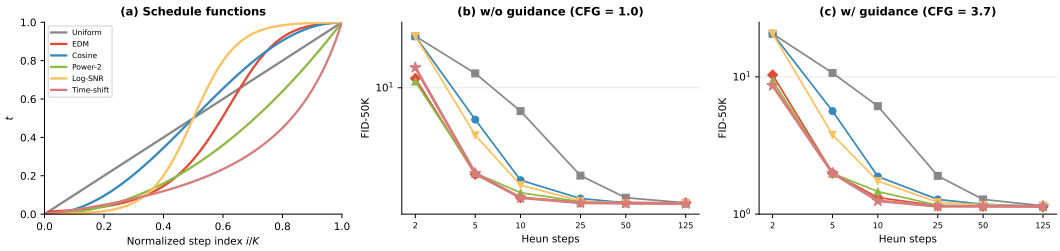



**Figure 11: Sampling schedule comparison.** (a) Schedule functions $t(i/K)$ for $K{=}50$ steps. (b, c) FID-50K against Heun steps without and with guidance. Coupled noise, RiT-XL on DINOv2-S, 760 epochs.

**Table 11: Sampling schedule ablation without guidance.** FID-50K by time discretization (rows) and number of Heun steps (columns). Each cell: independent / coupled `[CLS]` noise. RiT-XL on DINOv2-S, 760 epochs. Bold: best value per column, separately for independent and coupled noise; ties are all marked.

| **Schedule** | 2 | 5 | 10 | 25 | 50 | 125 |
|---|---|---|---|---|---|---|
| Uniform | 23.67 / 23.75 | 12.84 / 12.72 | 6.88 / 6.76 | 2.34 / 2.29 | 1.61 / 1.58 | 1.47 / 1.45 |
| EDM | 11.78 / 11.70 | **2.37** / **2.34** | 1.61 / **1.58** | 1.49 / 1.47 | 1.47 / 1.46 | 1.47 / 1.45 |
| Cosine | 23.67 / 23.75 | 5.96 / 5.86 | 2.16 / 2.12 | 1.57 / 1.56 | 1.48 / 1.45 | **1.45** / 1.44 |
| Power-2 | **11.16** / **11.09** | 2.41 / 2.39 | 1.74 / 1.72 | 1.50 / 1.48 | 1.47 / **1.44** | **1.45** / **1.43** |
| Log-SNR | 23.67 / 23.75 | 4.56 / 4.51 | 1.96 / 1.95 | 1.51 / 1.50 | **1.46** / **1.44** | **1.45** / 1.44 |
| Time-shift | 14.05 / 14.03 | 2.44 / 2.38 | **1.59** / **1.58** | **1.47** / **1.45** | **1.46** / **1.44** | **1.45** / **1.43** |

**Table 12: Sampling schedule ablation with guidance** (CFG scale 3.7). Same layout as Table 11.

| **Schedule** | 2 | 5 | 10 | 25 | 50 | 125 |
|---|---|---|---|---|---|---|
| Uniform | 20.67 / 20.63 | 10.80 / 10.68 | 6.19 / 6.12 | 1.93 / 1.90 | 1.30 / 1.28 | 1.16 / 1.15 |
| EDM | 10.43 / 10.34 | 2.01 / 1.99 | 1.33 / 1.32 | 1.17 / 1.15 | 1.16 / **1.14** | **1.14** / **1.13** |
| Cosine | 20.67 / 20.63 | 5.69 / 5.63 | 1.91 / 1.88 | 1.29 / 1.28 | 1.19 / 1.18 | 1.15 / 1.14 |
| Power-2 | 9.44 / 9.37 | **1.99** / **1.98** | 1.48 / 1.46 | 1.18 / 1.16 | 1.16 / 1.15 | 1.15 / 1.14 |
| Log-SNR | 20.67 / 20.63 | 3.79 / 3.78 | 1.73 / 1.74 | 1.23 / 1.22 | 1.18 / 1.17 | 1.15 / 1.14 |
| Time-shift | **8.59** / **8.64** | **1.99** / 1.99 | **1.27** / **1.25** | **1.15** / **1.14** | **1.15** / **1.14** | 1.15 / 1.14 |

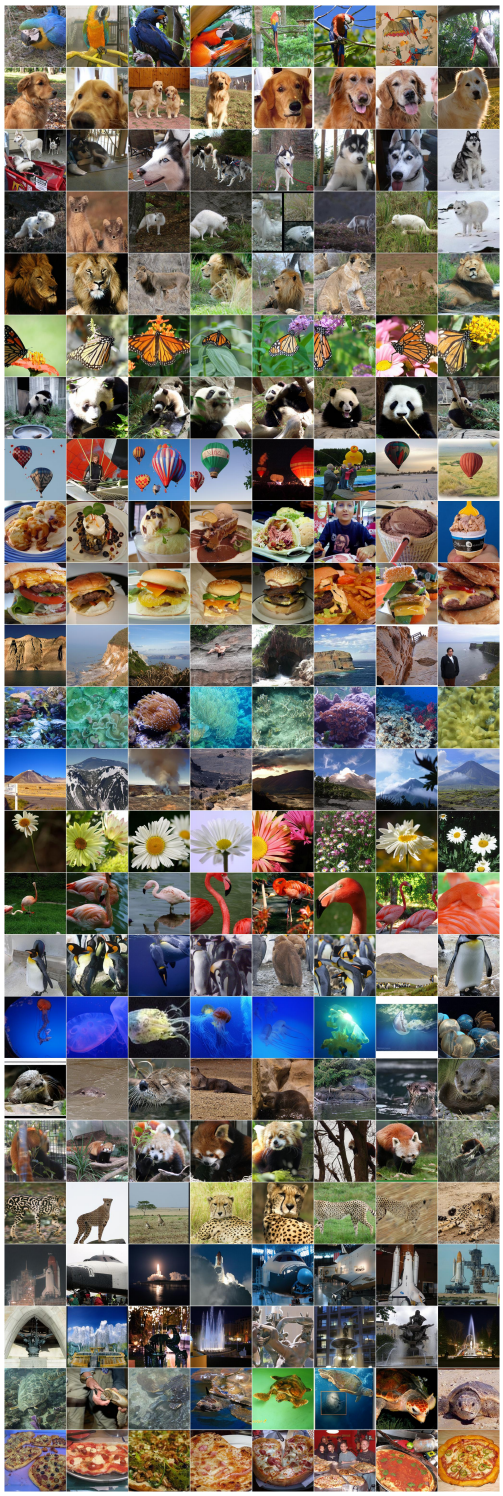


**Figure 12: RiT-XL samples without selection within a class, ImageNet $256^2$.** Each row shows the first 8 samples of one class, for 24 hand-picked classes. Heun 100 steps, CFG scale 3.7.

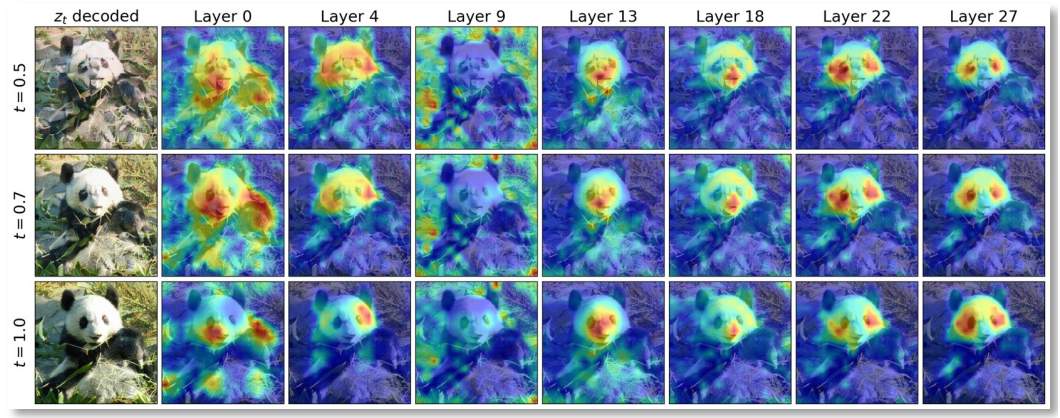



**Figure 13: Layer-wise `[CLS]`–patch attention in RiT.** Top: `[CLS]`-to-patch attention by layer. Bottom: patch-to-`[CLS]` attention by layer. Attention weights are averaged over heads, and each map is divided by its own maximum.