Improved Baselines with Representation Autoencoders

Jaskirat Singh[1,2] **Boyang Zheng**[3] **Zongze Wu**[1] **Richard Zhang**[1]
Eli Shechtman[1] **Saining Xie**[3]
[1]Adobe Research [2]ANU [3]New York University

Abstract

Representation Autoencoders (RAE) replace traditional VAE with pretrained vision encoders. In this paper, we systematically investigate several design choices and find three insights which simplify and improve RAE. First, we study a generalized formulation where the representation is defined as sum of the last k encoder layers rather than solely the final layer. This simple change greatly improves reconstruction without encoder finetuning or specialized data (e.g., text, faces). Second, we study the prevalent assumption that RAE (using pretrained representation as encoder) replaces representation alignment (REPA), which distills *same representation* to intermediate layers instead. Through large-scale empirical analysis, we uncover a surprising finding: RAE and REPA exhibit *complementary* working mechanisms, allowing same representation to be used as both encoder and target for intermediate diffusion layers. Finally, the original RAE struggles with classifier-free guidance (CFG) and requires training a second, weaker diffusion model for AutoGuidance (AG). We show that REPA itself can be viewed as x-prediction in RAE latent space. By simply re-parameterizing the output of DiT model, it can provide guidance for "free". Overall, RAEv2 leads to more than $10\times$ faster convergence over original RAE, achieving a state-of-the-art gFID of 1.06 in just 80 epochs on ImageNet-256. On FDr^k RAEv2 achieves state-of-art 2.17 at just 80 epochs compared to previous best 3.26 (800 epochs) without any post-training. This motivates $\text{EP}_{\text{FID@}k}$ (epochs to reach unguided gFID $\leq k$) as a measure of training efficiency. RAEv2 attains an $\text{EP}_{\text{FID@2}}$ of 35 epochs, versus 177 for the original RAE. We also validate our approach across diverse settings for text-to-image generation and navigation world models, showing consistent improvements. We hope that this work provides insights for practical adoption of representation autoencoders. Project page: https://raev2.github.io.

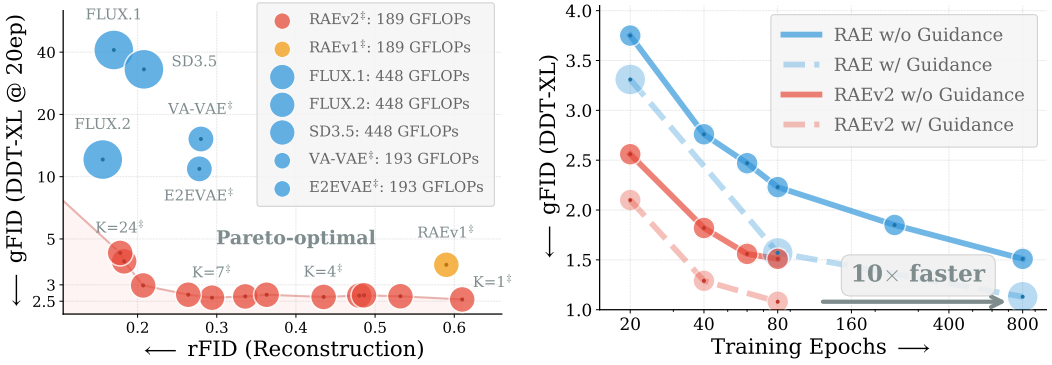
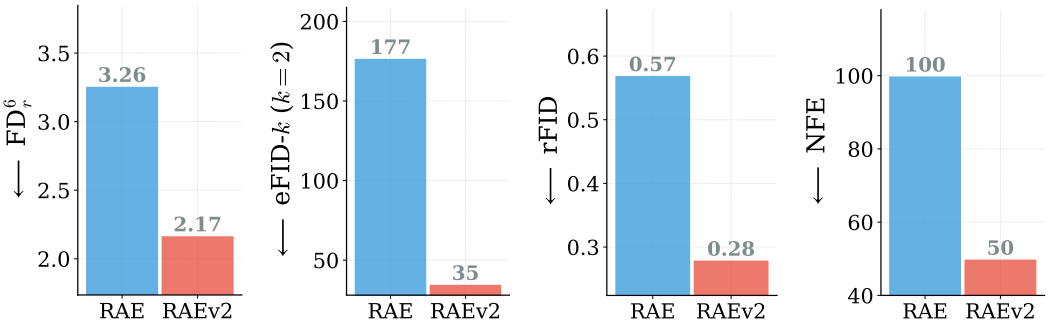


Figure 1: **Improved Representation Autoencoders.** *Left:* RAEv2 exhibits pareto-optimal reconstruction-generation performance at half the encoder FLOPs. ‡ denotes VAE / RAE / RAEv2 trained only on ImageNet. Training on more data (e.g., text) can further help reconstruction [66]. *Right:* Over $10\times$ faster convergence, achieving state-of-the-art gFID of 1.06 in just 80 epochs.

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



(a) Better generation (b) Faster convergence (c) Better reconstruction (d) Efficient inference

Figure 2: **Improved performance.** RAEv2 improves over RAE on (a) generation performance: achieving FDr^6 [76] of 2.17 in just 80 epochs over RAE 3.26 (800 epochs) without any post-training. (b) faster convergence: improving $\mathrm{EP}_{\mathrm{FID@2}}$ (epochs to reach unguided gFID $\leq$ 2) from 177 to 35 (see §3.2, Tab. 7). (c) better reconstruction (d) efficient inference: reusing the REPA head for guidance, eliminating need for separate model (AutoGuidance) and extra forward pass (CFG).

1 Introduction

Representation Autoencoders (RAE) [85] have emerged as a powerful framework for replacing traditional VAEs in diffusion transformer training [79, 85, 41, 60, 20], moving a step closer towards a unified tokenization for both understanding and generation. However, several problems persist towards practical adoption: 1) reconstruction performance lags behind specialized VAEs; 2) RAE is incompatible with traditional classifier-free guidance (CFG) [85], requiring training a secondary, weaker diffusion model for AutoGuidance [35], adding compute and complexity; and 3) the encoder representations themselves remain underexplored, with prior work defaulting to final-layer features.

In this paper, we systematically investigate several design choices and find three key insights which significantly simplify and accelerate RAE training.

Generalized Representation Autoencoder. Prior works typically consider only the final layer output of a pretrained vision encoder as the representation for RAE. However, the representation from a pretrained encoder is not just its final layer; rich and diverse abstractions exist across all layers. We propose a generalized, training-free formulation that simply defines the encoder output as the sum of its last k layers. We find that simply varying k allows easy control over reconstruction quality, leading to Pareto-optimal performance for both reconstruction and generation (Fig. 1, 2, 3).

RAE and REPA exhibit complementary working mechanisms. We next study the prevailing assumption [85, 73, 8] that RAE (using pretrained representation as latent space encoder) eliminates the need for REPA [79], which distills the *same representation* to intermediate diffusion layers. Since RAE already uses encoder features as input, distilling them again to intermediate layers appears to be a wasteful skip connection. We perform large-scale empirical analysis across 27 vision encoders studying the working mechanism of RAE and REPA. The results are surprising: RAE and REPA operate through complementary mechanisms. RAE provides a more semantically rich latent space, while REPA improves the spatial structure of intermediate diffusion features [60]. This encourages using the same representation as both encoder (RAE) and target for intermediate layers (REPA). Furthermore, the complementary mechanism enables stronger encoders (e.g., DINOv3-L) good in both global and spatial performance [60, 59] to also exhibit better generation performance (§2.2).

REPA is x-prediction in RAE latent space. The original RAE struggles with traditional classifier-free guidance (CFG), instead relying on AutoGuidance [35], which requires training a secondary weaker diffusion model, adding compute and complexity. We observe a key property: when used with RAE, the REPA prediction head performs x-prediction in the target representation space. By simply reformulating the output head as also x-prediction [43], we find that the REPA head itself can be used as the weaker baseline for internal-guidance [88]. This eliminates the need for a separate model entirely (AG). Also unlike CFG, which requires an additional unconditional forward pass (doubling the number of function evaluations at inference), internal-guidance [88] with REPA head in x-prediction space is computed within the same forward pass, effectively halving the NFEs.

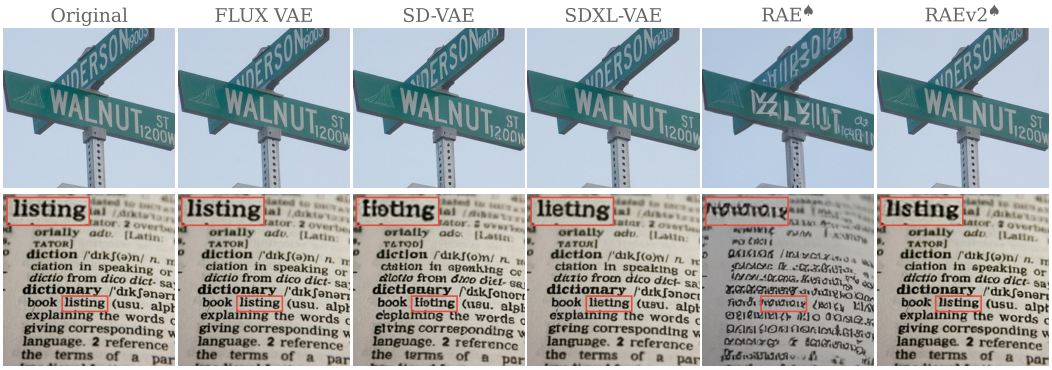


Figure 3: **Qualitative reconstruction comparison.** ♠ denotes trained only on ImageNet. RAEv2 despite only being trained on Imagenet performs competitively with proprietary VAEs. Training on more data (e.g., text) can further help reconstruction [66] (see Fig. 10). Results use DINOv3-L (K=23) for RAEv2.

We combine these insights into an improved baseline RAEv2, which exhibits over $10\times$ faster convergence over original RAE, achieving state-of-art gFID of 1.06 in just 80 epochs. On recently proposed FDr^k metric [76] RAEv2 achieves 2.17 in just 80 epochs as opposed to previous best 3.26 (800 epochs). With improved convergence speed of RAEv2, we believe that incremental improvements in the gFID metric might provide little signal for practical applications. Instead the training efficiency of a given method, provides much more useful signal for practical applications. Motivated by recent speedrun in language domain [34], we therefore report $\text{EP}_{\text{FID@}k}$ metric (epochs to reach unguided gFID $\leq k$) as a measure of training efficiency (Tab. 7). Notably, RAE marks a huge jump over prior works reducing $\text{EP}_{\text{FID@2}}$ from 480 to 177. RAEv2 further boosts the training efficiency achieving $\text{EP}_{\text{FID@2}}$ of just 35 epochs [0]. We also validate our approach across diverse settings including text-to-image generation and navigation world models [4] (§3.3), showing consistent improvements.

2 Improved Representation Autoencoders

We next discuss the improved baseline analyzing three insights for improving and simplifying RAE. First, in §2.1 we generalize the RAE formulation to treat the encoder representation not as a single final-layer feature but as a signal distributed across all layers. This simple change greatly improves reconstruction without encoder finetuning or specialized data (e.g., text, faces) [66]. Next, in §2.2 we perform large-scale empirical analysis finding that RAE and REPA exhibit complementary working mechanisms. As a result, using the same representation as both encoder and intermediate target consistently improves generation. We then in §2.3 show that REPA when applied with RAE can be viewed as performing x-prediction [43] in the target latent space. We therefore propose a simple reformulation, which allows the REPA prediction head itself to be used for guidance.

2.1 Generalized Representation Encoder

Prior work on RAE usually consider the encoder output as the final-layer feature of a pretrained vision encoder. However, different layers of a pretrained encoder capture complementary features [6]. As shown in Fig. 11, feature visualizations and spatial self-similarity patterns vary substantially across depth, with later layers emphasizing global semantics and earlier-to-middle layers retaining finer spatial structure. The final layer alone is therefore not always the most informative signal for generation. A natural question arises: *"instead of just relying on the final layer features, can we leverage features* across all layers *without introducing additional parameters or training cost?"*

Naive concatenation is impractical. A direct way to use multi-layer features is to concatenate them along the sequence or channel dimension. For an encoder with L layers producing N tokens of dimension d each, this yields an $LN \times d$ latent sequence. While lossless, this causes an explosion in the latent sequence length, making the resulting latent space substantially more expensive for

[0] RAEv2 trains within $\sim$10.5 hours on our setup, compared to $>$1 week for 800 epochs in RAE [85].

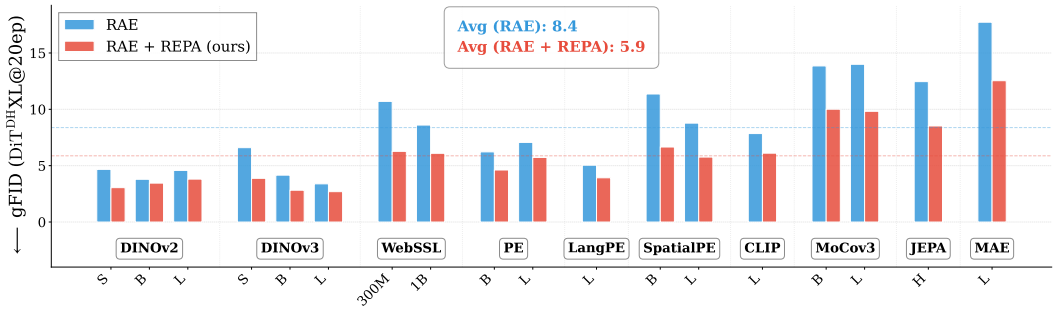


Figure 4: **RAE does not eliminate need for REPA.** Prevailing assumptions say that using the pretrained representation (*e.g.*, DINOv2) as both encoder and target of intermediate representations wastes model capacity by introducing a skip connection. Surprisingly, we instead find that RAE and REPA when used together work through complementary working mechanisms (§2.2). This leads to consistent improvements in generation performance across all pretrained representations.

the diffusion model. On the other hand, concatenation along the channel dimension yields $N \times Ld$ significantly increasing the latent space dimension, making it harder to learn the diffusion model [77].

We instead consider two approaches that combine features across the last K layers while preserving original latent shape $N \times d$. Let $\boldsymbol{z}_\ell \in \mathbb{R}^{N\times d}$ denote the feature map at layer ℓ of an L-layer encoder.

- **Simple addition.** The encoder output is defined as the sum of the last K layer features. In high-dimensional spaces, addition preserves the geometric structure of the underlying subspaces [70]:

$$\boldsymbol{x} \;=\; \sum_{\ell=L-K+1}^{L} \boldsymbol{z}_\ell \;\in\; \mathbb{R}^{N\times d}. \tag{1}$$

- **Random-matrix projection.** We concatenate the last K layer features along the channel dimension and project back to d with a fixed random matrix $\boldsymbol{R} \in \mathbb{R}^{Kd\times d}$ (sampled once at initialization, *e.g.* i.i.d. Gaussian, and held fixed). Random projections are a standard tool in dimensionality reduction [70] and preserve pairwise distances in expectation:

$$\boldsymbol{x} \;=\; \left[\, \boldsymbol{z}_{L-K+1} \,\middle\|\, \cdots \,\middle\|\, \boldsymbol{z}_L \,\right] \boldsymbol{R} \;\in\; \mathbb{R}^{N\times d}. \tag{2}$$

The original RAE is thus a special case in this generalized formulation with $K = 1$ i.e., just the final layer. Both approaches keep the latent footprint identical to the original RAE and add no extra learned parameters. We defer a head-to-head empirical comparison of the two to §3.

Finding 1: Generalized Representation Encoders. Pretrained vision encoders are more than their final layer. Aggregating features across layers of a pretrained vision encoder greatly improves reconstruction without encoder finetuning or specialized data (e.g., text, faces).

2.2 RAE and REPA exhibit Complementary Working Mechanisms

Empirical results. We next study the prevailing assumption [85, 73, 8] that RAE eliminates need for REPA. Since RAE already uses encoder features as input, distilling them again to intermediate layers appears to be a wasteful skip connection. To this end, we first perform large-scale empirical analysis, using the same representation as both encoder and target at intermediate diffusion layers (refer Fig. 4). Results are surprising. Across all encoders, instead of hurting performance, the use of REPA with RAE consistently leads to better generation performance. This suggests a fundamental difference in how representation alignment (REPA) and RAE benefit diffusion training.

Working mechanism. We next analyze how REPA impacts diffusion features when combined with RAE. As shown in Fig. 5, adding REPA on top of RAE has minimal impact on the peak *global semantic information* (measured through linear probing) of diffusion features. Instead, we observe that REPA improves the spatial self-similarity structure of the learned diffusion features (i.e., how different tokens pay attention to each other) - an intriguing phenomenon recently identified in



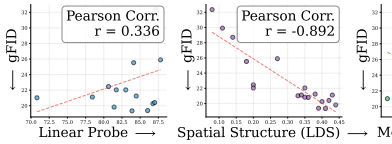
(a) Impact of REPA on global semantics with RAE (DINOv2-B)

(b) Impact of REPA on spatial structure with RAE (DINOv2-B)

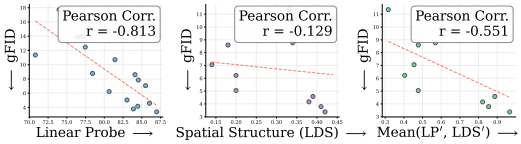
Figure 5: **Working mechanism of REPA with RAE.** While REPA applied with RAE has minimal impact on global semantics, it significantly improves spatial structure [60] of diffusion features.

iREPA [60]. This suggests complementary working mechanisms for REPA and RAE: RAE provides a semantically rich latent space for diffusion, while REPA regularizes the token-token similarity structure in the intermediate diffusion features.

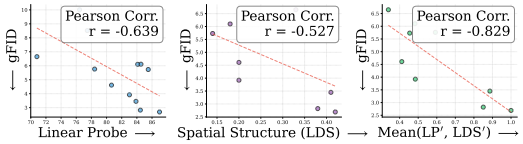
Correlation analysis. To further validate the complementary mechanisms of RAE and REPA, we follow the practice in iREPA [60], analyzing the Imagenet linear probing accuracy (LP) and local distance similarity score (LDS) [60] across 27 vision encoders, and report their Pearson correlation r with generation quality (gFID). As shown in Fig. 6, for REPA alone (with VAE), LDS is highly predictive ($|r|$=0.89) while LP is actually anticorrelated (r=+0.34), consistent with findings in [60]. In contrast, when using RAE alone, LP dominates ($|r|$=0.81) while LDS barely correlates (r=|0.13|). When combining RAE with REPA, neither metric alone is strongly predictive, but the average of LP (global semantics) and LDS (spatial structure) achieves the highest correlation ($|r|$=0.83). This confirms that RAE and REPA operate through complementary mechanisms: RAE leverages global semantics while REPA regularizes spatial structure.



(a) REPA alone (SD-VAE)



(b) RAE alone



(c) RAE + REPA

Method	LP (r)↓	LDS (r)↓	Avg (r)↓
REPA alone	+0.34	**-0.89**	-0.56
RAE alone	**-0.81**	-0.13	-0.55
RAE + REPA	-0.64	-0.53	**-0.83**

(d) Pearson correlation r with gFID

Figure 6: **RAE and REPA leverage complementary encoder properties.** Correlation analysis with gFID across 27 vision encoders for (a) REPA alone, (b) RAE alone, (c) RAE + REPA. (a) Similar to [60], performance with REPA alone correlates more with spatial structure (LDS) of a representation. (b) RAE alone benefits much more from higher global semantics (LP). (c) Together, RAE and REPA benefit from encoders strong in both global semantics (LP) and spatial structure (LDS). This explains why stronger encoders (e.g., DINOv3-L) which perform well in both metrics yield the best generation with RAEv2 (Tab. 2). All results: DDT-XL, 20 epochs, without guidance.

Selecting the best representation. The above complementarity also enables stronger representations (e.g., DINOv3-L) that perform well for both global (LP) and spatial (LDS) performance, to also exhibit better generation with RAEv2 We defer the detailed encoder-selection study to §3.1.

Finding 2: RAE and REPA exhibit complementary working mechanisms. RAE leverages semantic quality while REPA regularizes spatial structure. This complementary nature allows using same pretrained representation as both encoder (RAE) and target for intermediate diffusion features (REPA). This also explains why stronger representations like DINOv3-L, which excel in both metrics, achieve the best generation with RAEv2 (see Tab. 2).

2.3 Reformulating REPA as x-prediction with RAE

We next show that when used with RAE, the REPA head itself can be used for guidance, eliminating need for training second weaker diffusion model (AutoGuidance) or additional forward pass (CFG).

RAE struggles with traditional CFG. As shown in Tab. 1, we confirm that original RAE [85] struggles with standard classifier-free guidance [29]. RAE therefore relies on AutoGuidance [35], a separate, smaller model trained to serve as the "weaker" baseline, adding compute and complexity.

Guidance	gFID↓	IS↑
w/o Guidance	3.75	198.7
CFG [29]	3.86	276.4
Autoguidance (AG) [35]	**3.31**	219.1

Table 1: RAE DiT^{DH}-XL, 20 epochs.

REPA is x-prediction in RAE latent space. We observe a key connection. In RAE, the clean latent *is* the encoder representation: $\boldsymbol{x} = E(\boldsymbol{I})$. The REPA projection head h_ϕ maps early-layer intermediate features $\boldsymbol{h}$ to predict the clean latent $\hat{\boldsymbol{x}}_{\text{repa}} = h_\phi(\boldsymbol{h})$. This is exactly x-prediction [43] in the RAE latent space. Importantly, because h_ϕ is a lightweight MLP that only accesses early-layer features, its prediction is inherently weaker than the full model's, playing the same role as the separately trained smaller model in AutoGuidance [35].

Reformulating REPA head for guidance. If we reformulate the full model output to also give x-prediction instead of velocity [49], both outputs live in the same space. Let $\hat{\boldsymbol{x}}_{\text{full}}$ denote the full model's x-prediction (all layers) and $\hat{\boldsymbol{x}}_{\text{repa}}$ the REPA head's x-prediction (early layers only). We can then apply internal-guidance [88] directly as,

$$\hat{\boldsymbol{x}}_{\text{guided}} = \hat{\boldsymbol{x}}_{\text{full}} + w \cdot (\hat{\boldsymbol{x}}_{\text{full}} - \hat{\boldsymbol{x}}_{\text{repa}}), \tag{3}$$

and convert back to velocity for sampling or loss computation: $\boldsymbol{v} = (\boldsymbol{x}_t - \hat{\boldsymbol{x}}_{\text{guided}})/t$. The REPA head runs during the same forward pass as main model, so this eliminates the need for training a separate weaker model (AG [35]) and no additional forward pass (CFG [29]).

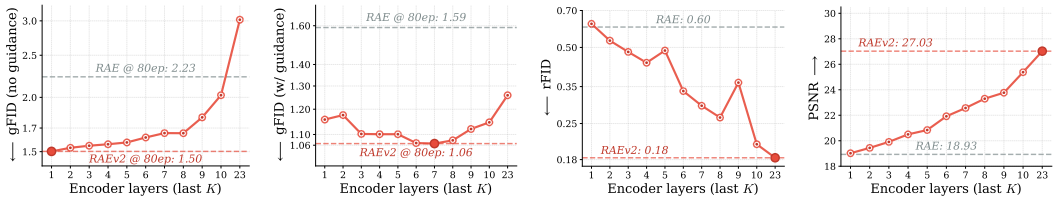
Thus, *when used with RAE* our formulation is equivalent to a deep supervised network [39] or internal-guidance [88], with additional reparameterization to x-prediction. The reparameterization to x-prediction [43] is important as it allows use of REPA-head for both supervising spatial structure of intermediate layers (§2.2) and act as a weaker baseline for guidance after reparametrization.

Finding 3: REPA enables self-guidance. REPA is x-prediction in RAE latent space. By reformulating output head also as x-prediction, REPA head itself can be used for internal-guidance; eliminating need for a separate model (AG) or extra forward pass (CFG).

3 Experiments

We validate the performance of our approach through extensive experiments on ImageNet, text-to-image generation and world models. In particular, we investigate the following research questions:

- Does the improved training recipe consistently improve convergence speed with representation autoencoders across diverse settings, model scales etc? (Fig. 1, 2, 4, 8, 9; Tab. 2, 4, 6, 7)
- Can we use generalized RAE formulation for improving reconstruction performance of representation autoencoders in a training free manner? (Fig. 1, 3, 7, 9; Tab. 3, 13)
- Does the proposed approach generalize to diverse training settings including text-to-image generation and world models? (Fig. 12, 13, 14; Tab. 9, 19)



(a) gFID ↓ (no guidance) (b) gFID ↓ (with guidance) (c) rFID ↓ (d) PSNR ↑

Figure 7: **Ablation on choice of K for generalized RAE.** (a, b) Stage-2 generation quality without and with guidance. (c, d) Stage-1 reconstruction (rFID and PSNR). All results with DINOv3-L (24 layers), DDT-XL and 80 epochs. Stage-1 reconstruction (rFID, PSNR) improves monotonically with K. Interestingly, the generalized RAE not only improves reconstruction performance but also leads to better generation performance with guidance (best at $K = 7$).

3.1 Ablation Studies

We first ablate different design choices for different components proposed in §2 on ImageNet-256. Unless otherwise specified we use DiT^{DH}-XL, DINOv3-L as encoder and batch size 1024.

Encoder selection. Results are shown in Tab. 2. The original RAE [85] uses DINOv2-B as its encoder because it gives the best generation among the encoders tested under the RAE recipe. With RAEv2, however, the picture changes: stronger representations such as DINOv3-B [59] yield better generation, despite performing worse than DINOv2-B under the original RAE recipe. This is consistent with correlation analysis in §2.2; stronger representations (e.g., DINOv3-L) which excel in both global semantics and spatial performance lead to best generation with RAEv2. Based on this finding, we use DINOv3-L as the default encoder for all subsequent RAEv2 experiments.

Formulation for generalized RAE. In Tab. 3, we compare the two parameter-free aggregation schemes from §2.1 on DINOv3-L: simple addition of the last K encoder layers (**MLS**) versus fixed random-matrix projection of their channel-wise concatenation (**MLR**). Interestingly, while both methods are effectively tied on Stage-1 reconstruction, MLS consistently wins on Stage-2 performance. We therefore use MLS as the default aggregation in the rest of the paper.

Encoder	Encoder properties			gFID (DiT^{DH}-XL @ 20ep) ↓	
	LP ↑	LDS ↑	Avg(LP', LDS') ↑	RAE	RAEv2 (k=1)
MoCov3-B [15]	76.4	0.15	0.46	13.84	8.35
WebSSL-1B [16]	84.1	0.18	0.51	8.60	4.16
DINOv3-B [59]	84.5	0.38	0.61	4.25	2.76
DINOv2-B [48]	83.9	0.41	0.62	3.75	2.81
DINOv3-L [59]	**87.0**	**0.42**	**0.65**	**3.30**	**2.61**

Table 2: **Ablation on choice of pretrained vision encoder.** gFID at 20 epochs (DiT^{DH}-XL). We observe that with RAEv2, stronger encoders e.g, DINOv3-L with both better global (LP) and spatial (LDS) [60] representations achieve the best performance. Please refer Tab. 12 for further results.

Choice of K for generalized RAE. We sweep $K \in \{1, \ldots, 10, 23\}$ for generalized RAE on DINOv3-L (Fig. 7). Stage-1 reconstruction (rFID, PSNR) improves monotonically with K, rFID 0.18 and PSNR 27.03 at K=23, well past the standard RAE baseline (rFID 0.60, PSNR 18.93). Stage-2 generation behaves differently: at just 80 epochs, the unguided gFID is best near K=1 (1.50), while the guided gFID performs best with K=7 (1.06). Thus, interestingly the generalized RAE not only improves reconstruction performance but also leads to better generation performance with guidance.

Guidance mechanism in RAEv2. We ablate four guidance options for RAEv2 in Tab. 4: (i) no guidance, (ii) classifier-free guidance (CFG) [29], (iii) AutoGuidance [35], and (iv) internal guidance [88] with REPA-head and x-prediction (§2.3). CFG fails to meaningfully improve gFID, AG helps but requires an additional model and forward pass. In contrast, internal guidance with REPA-head achieves the best gFID at no extra inference cost.

K	Method	rFID ↓	gFID ↓
2	MLR	0.570	3.085
	MLS	0.532	**2.586**
8	MLR	0.268	3.580
	MLS	0.264	**2.688**

Table 3: **Ablation on Generalized RAE formulation.** MLS dominates MLR for gfid (see §3.1). Full sweep in Tab. 16.

Guidance	gFID (K=7) ↓	gFID (K=23) ↓
w/o Guidance	1.65	3.01
CFG [29]	1.49	2.83
Autoguidance (AG) [35]	1.14	1.37
REPA Guidance	**1.06**	**1.25**

Table 4: **Ablation on Guidance mechanism in RAEv2.** Guidance with REPA and x-prediction achieves best results at no extra inference cost. Full results in Tab. 17.

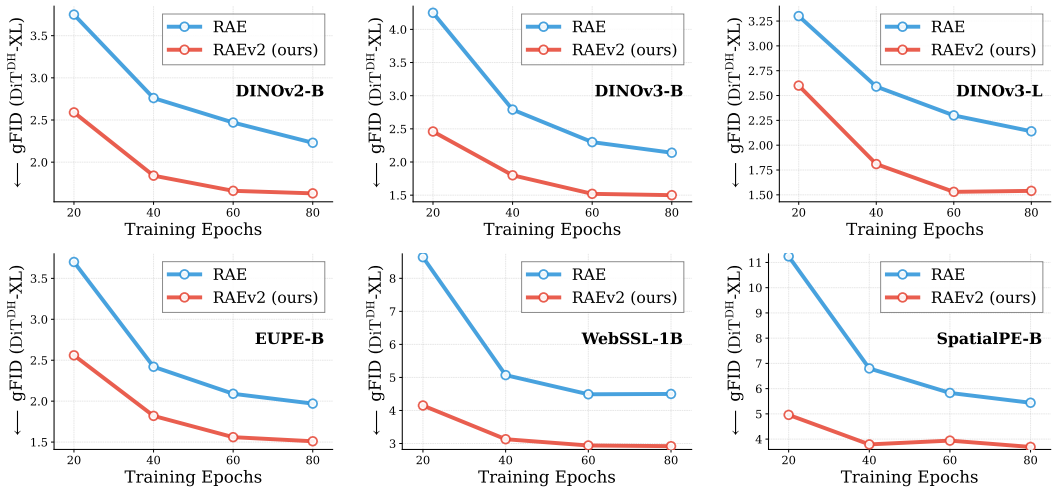


Figure 8: **Convergence comparison with original RAE.** Across DINOv2-B [48], DINOv3-B/L [59], EUPE-B [89], WebSSL-1B [16], and SpatialPE-B [6], the improved training recipe (RAEv2) consistently leads to faster convergence. All results: DiT^{DH}-XL, $K = 1$ for RAEv2, 1024 batch-size.

Impact of generalized RAE on understanding performance. A key advantage of RAE is that it provides a unified tokenization for both understanding and generation. We study the impact of the generalized formulation on the encoder's understanding performance with different K in Tab. 5 (K=1 is the original RAE). The generalized formulation improves reconstruction and guided generation (Fig. 7) while preserving linear probing performance on ImageNet. Full sweep over $K \in \{1, \ldots, 10, 23\}$ is in Tab. 18.

K	LP top-1 (%) ↑
1 (last layer; RAE)	85.39
4	85.15
7	85.10
23 (full MLS)	85.24

Table 5: Linear probing on ImageNet across K (DINOv3-L); 30 epochs of LP training, further training may improve scores. Full sweep in Tab. 18.

Finding 4: Generalized formulation of RAE helps improve reconstruction and generation performance with guidance (Fig. 7) while preserving global semantics of the representation space (Tab. 5). This enables its use for a unified tokenization for both understanding and generation.

3.2 Impact on Convergence Speed

Convergence speed. Results are shown in Fig. 8. We observe that across various vision encoders, RAEv2 consistent improves convergence speed over original RAE.

Variation in Model scale. We further validate that the gains from RAEv2 transfer across model scales. Tab. 6 compares RAE and RAEv2 on DiT^{DH}-B, -L, and -XL backbones at 20 epochs: RAEv2 consistently improves generation performance across different scales.

Scale	#Params	gFID (RAE) ↓	gFID (RAEv2) ↓
B	165M	5.48	**3.37**
L	470M	3.80	**2.76**
XL	839M	3.75	**2.61**

Table 6: Variation in model scale.

Method	Epochs	Training Efficiency		Representation Fréchet Distance (FD_r) [76] ↓						FD_r^6↓
		$EP_{FID@2}$ ↓	gFID↓	Incep.	ConvNeXt	DINOv2	MAE	SigLIP	CLIP	
SiT-XL/2 [46]	800	>800	2.12	1.26	2.02	7.89	5.62	16.14	17.69	8.44
DDT-XL [69]	800	–	1.26	0.75	1.02	4.26	4.11	10.16	13.86	5.70
SiT-XL/2-REPA [79]	800	>800	1.42	0.85	1.22	4.27	3.85	9.87	12.65	5.45
LightningDiT [78]	800	>800	1.42	0.85	1.09	3.76	3.02	8.47	10.21	4.57
REG [71]	800	560	1.54	0.92	1.14	3.45	3.02	8.42	10.86	4.64
REPA-E [41]	800	480	1.12	0.70	1.28	2.44	**2.52**	5.04	6.28	3.04
RAE-XL [85]	800	177	1.13	0.69	1.79	2.11	3.30	3.79	7.87	3.26
RAEv2 (K=7, **ours**)	**80**	**35**	**1.06**	**0.64**	**0.77**	**1.15**	2.67	**2.54**	**5.21**	**2.17**

Table 7: **Training efficiency and evaluation under alternative metrics. Left:** Training efficiency comparisons reporting $EP_{FID@2}$ (epochs to reach unguided gFID ≤ 2) and the final guided gFID. Unlike incremental improvements in gFID, $EP_{FID@2}$ provides a much better signal for training convergence across methods. **Right:** Representation Fréchet Distance (FD_r) [76] computed in six feature spaces (Inception, ConvNeXt, DINOv2, MAE, SigLIP, CLIP), and FD_r^6. Compared to prior works trained for 800 epochs, RAEv2 attains the best $EP_{FID@2}$, gFID, and FD_r^6 in just 80 epochs, without any post-training.

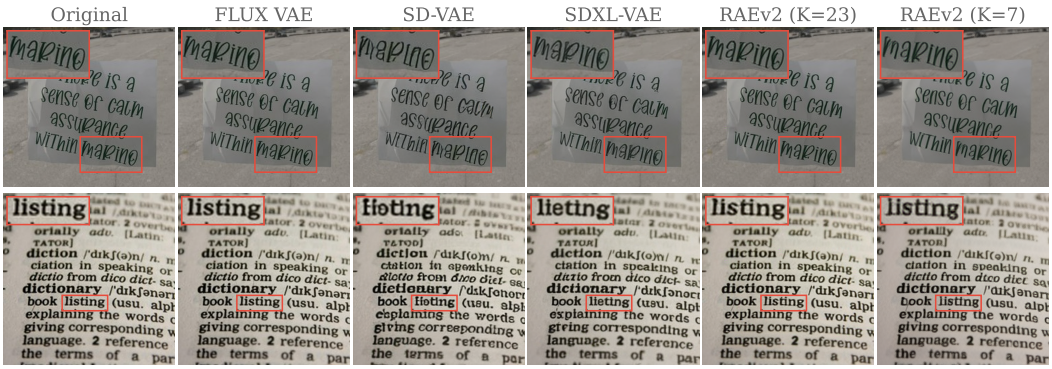


Figure 10: **Qualitative reconstruction comparisons** with additional data for decoder training [66] (pretrained vision encoder remains frozen). RAEv2 performs competitively with proprietary VAEs. Results use DINOv3-L as encoder for RAEv2. Please see Tab. 14 for further quantitative results.

Training efficiency. With improved convergence speed of RAEv2 (1.06 gFID in just 80 epochs), we believe that incremental improvements in the gFID metric might provide little signal for practical applications. Instead the training efficiency of a given method, provides much more useful signal for practical applications (e.g., T2I, world models etc §3.3). Motivated by the recent speedrun in the language domain [34], we define the $EP_{FID@k}$ metric (epochs to reach unguided gFID $\leq k$). We report results for $EP_{FID@k}$ ($k = 2$) in Tab. 7. Compared to absolute gFID which shows little variance among various methods, we observe that $EP_{FID@k}$ provides a much better signal for measuring training efficiency of a method. Notably, RAE marks a huge jump over prior works reducing $EP_{FID@2}$ from 480 to 177. RAEv2 further boosts the training efficiency achieving $EP_{FID@2}$ of just 35 epochs.

Evaluation with alternate metrics. We also evaluate generation quality with alternate evaluation metrics using recently proposed Representation Fréchet Distance (FD_r) [76], which scores sample fidelity in six different feature spaces: Inception, ConvNeXt, DINOv2, MAE, SigLIP, and CLIP. As shown in Tab. 7, despite training for only 80 epochs, RAEv2 achieves state-of-the-art performance on both FID and FD_r^6, surpassing prior baselines that are trained for 800 epochs without any post-training.

Reconstruction and generation tradeoff. Results are shown in Fig. 9. RAEv2 achieves Pareto-optimal trade-off between generation (gFID) and reconstruction (rFID) without encoder finetuning or specialized data (e.g, text) [66]. All results are reported with DINOv3-L encoder, DDT-XL and 20 epochs. Please also see Tab. 13 for further comparisons.

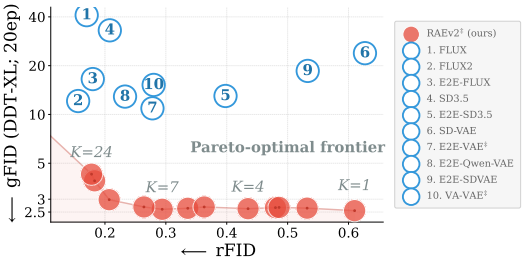


Figure 9: Reconstruction-generation trade-off

3.3 Generalization to Other Tasks

We further validate the generalization of the improved baseline (RAEv2) on text-to-image generation and navigation world model [4] tasks. Please refer §E for detailed task setup and results.

Text-to-image generation. We also validate our approach for T2I generation. For pretraining we use JourneyDB [64], long and short caption datasets from [13] and train for 150K iterations with 1024 batchsize. For finetuning, we follow [66] and use BLIP3o-60k dataset [13]. We observe that RAEv2 leads to consistent improvements over Flux-VAE and original RAE. Please see §E.1 for detailed results.

	Pretraining		Finetuning	
Method	GenEval ↑	DPG ↑	GenEval ↑	DPG ↑
Flux-VAE [38]	41.7	77.6	78.3	79.2
RAE [85]	58.4	80.1	81.5	80.6
RAEv2 (ours)	**62.4**	**81.7**	**82.7**	**82.3**

Table 8: Text-to-image generation.

Navigation world models. We further validate our approach for action-conditioned future-frame prediction [4], which stress-tests the latent space along two axes: 1) substantially longer conditioning context, and 2) autoregressive rollouts that compound error over time. On the RECON dataset [63], RAEv2-NWM achieves substantially better generation quality over RAE and previous baselines [4, 1]. Please see §E.2 for detailed setup and further results.

Method	FVD [67] ↓
DIAMOND [1]	762.73
NWM [4]	200.97
RAE [85]	312.01
RAEv2 (ours)	**105.61**

Table 9: Video prediction quality upto 16s on RECON [63].

4 Related Work

We discuss the most relevant work here and provide more detailed discussion in the appendix.

Pretrained encoders as latent spaces. A growing line of work replaces VAE latents with pretrained vision encoders for diffusion [85, 66, 58, 10, 20, 11, 74, 7]. We show that the original RAE recipe can be significantly improved with few simple insights leading to more then 10× faster convergence.

Representation alignment distills pretrained representations to intermediate diffusion layers [79, 60, 41, 71, 69]. We study the prevalent assumption [85, 73, 8] that RAE (using pretrained representation as encoder) eliminates the need for REPA. We find that RAE and REPA work through complementary mechanisms. Their combination is not only useful but also simplifies guidance with RAE.

Reconstruction quality of vision encoders. [66] tries to improve RAE reconstruction using specialized data (text, faces). [44, 83, 17, 32, 8, 9, 24, 5] finetune pretrained encoder itself for reconstruction. We find that frozen vision encoders themselves contain low-level details for reconstruction; achieving pareto-optimal reconstruction generation performance without any additional training.

5 Conclusion

We study an improved baseline which simplifies and improves RAE. We find that frozen vision encoders themselves contain low-level details for reconstruction. Simply aggregating the last K layers leads to pareto-optimal reconstruction-generation performance. We next perform large-scale empirical analysis showing that RAE and REPA exhibit complementary working mechanisms. Their combination is not only useful but also simplifies guidance with RAE. Furthermore it enables stronger representations (e.g., DINOv3-L) which excel in both spatial and global performance to also give better generation performance. Overall, RAEv2 achieves 10× faster convergence over RAE, improves reconstruction, and achieves state-of-art gFID and FDr^6 in just 80 epochs without any post-training. We hope our work provides useful insights for practical adaption of representation autoencoders.

References

[1] Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, 2024.

[2] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15619–15629, 2023.

[3] Fan Bao, Shen Nie, Kaiwen Xue, Chongxuan Li, Shi Pu, Yaole Wang, Gang Yue, Yue Cao, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22669–22679, 2023.

[4] Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. *arXiv preprint arXiv:2412.03572*, 2024. URL https://arxiv.org/abs/2412.03572.

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

[6] Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, Junke Wang, Marco Monteiro, Hu Xu, Shiyu Dong, Nikhila Ravi, Daniel Li, Piotr Dollár, and Christoph Feichtenhofer. Perception encoder: The best visual embeddings are not at the output of the network. *arXiv:2504.13181*, 2025.

[7] Ramon Calvo-González and François Fleuret. Laminating representation autoencoders for efficient diffusion. *arXiv preprint arXiv:2602.04873*, 2026.

[8] Hun Chang, Byunghee Cha, and Jong Chul Ye. Dino-sae: Dino spherical autoencoder for high-fidelity image reconstruction and generation. *arXiv preprint arXiv:2601.22904*, 2026.

[9] Bowei Chen, Sai Bi, Hao Tan, He Zhang, Tianyuan Zhang, Zhengqi Li, Yuanjun Xiong, Jianming Zhang, and Kai Zhang. AlignTok: Aligning visual foundation encoders to tokenizers for diffusion models. *arXiv preprint arXiv:2509.25162*, 2025.

[10] Bowei Chen, Sai Bi, Hao Tan, He Zhang, Tianyuan Zhang, Zhengqi Li, Yuanjun Xiong, Jianming Zhang, and Kai Zhang. Aligntok: Aligning visual foundation encoders to tokenizers for diffusion models. *arXiv preprint arXiv:2509.25162*, 2025.

[11] 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 *International Conference on Machine Learning*, 2025.

[12] Huayu Chen, Kai Jiang, Kaiwen Zheng, Jianfei Chen, Hang Su, and Jun Zhu. Visual generation without guidance. *arXiv preprint arXiv:2501.15420*, 2025.

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

[14] Shoufa Chen, Mengmeng Xu, Jiawei Ren, Yuren Cong, Sen He, Yanping Xie, Nicu Sebe, and Wai Lam. Gentron: Diffusion transformers for image and video generation. *arXiv preprint arXiv:2312.04557*, 2023.

[15] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9640–9649, 2021.

[16] David Fan, Shengbang Tong, Jiachen Zhu, Koustuv Sinha, Zhuang Liu, Xinlei Chen, Michael Rabbat, Nicolas Ballas, Yann LeCun, Amir Bar, et al. Scaling language-free visual representation learning. *arXiv preprint arXiv:2504.01017*, 2025.

[17] Weichen Fan, Haiwen Diao, Quan Wang, Dahua Lin, and Ziwei Liu. The prism hypothesis: Harmonizing semantic and pixel representations via unified autoencoding. *arXiv preprint arXiv:2512.19693*, 2025.

[18] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. *arXiv preprint arXiv:2410.12557*, 2024.

[19] Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. DreamSim: Learning new dimensions of human visual similarity using synthetic data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2306.09344.

[20] 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.

[21] Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025.

[22] Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J. Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models. *arXiv preprint arXiv:2512.02012*, 2025.

[23] Dhruba Ghosh, Hanna Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, 2023.

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

[25] Jiatao Gu, Yuyang Wang, Yizhe Zhang, Qihang Zhang, Dinghuai Zhang, Navdeep Jaitly, Josh Susskind, and Shuangfei Zhai. Dart: Denoising autoregressive transformer for scalable text-to-image generation. *arXiv preprint arXiv:2410.08159*, 2024.

[26] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[27] Jonathan Heek, Emiel Hoogeboom, Thomas Mensink, and Tim Salimans. Unified latents (ul): How to train your latents. *arXiv preprint arXiv:2602.17270*, 2026.

[28] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

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

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

[31] Drew A Hudson, Daniel Zoran, Mateusz Malinowski, Andrew K Lampinen, Andrew Jaegle, James L McClelland, Loic Matthey, Felix Hill, and Alexander Lerchner. Soda: Bottleneck diffusion models for representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19461–19471, 2024.

[32] Mingkai Jia, Mingxiao Li, Zhijian Shu, Anlin Zheng, Liaoyuan Fan, Jiaxin Guo, Tianxing Shi, Dongyue Lu, Zeming Li, Xiaoyang Guo, Xiaojuan Qi, Xiao-Xiao Long, Qian Zhang, Ping Tan, and Wei Yin. Dino-tok: Adapting dino for visual tokenizers. *arXiv preprint arXiv:2511.20565*, 2025.

[33] Dengyang Jiang, Mengmeng Wang, Liuzhuozheng Li, Lei Zhang, Haoyu Wang, Wei Wei, Guang Dai, Yanning Zhang, and Jingdong Wang. No other representation component is needed: Diffusion transformers can provide representation guidance by themselves. *arXiv preprint arXiv:2505.02831*, 2025.

[34] Keller Jordan, Jeremy Bernstein, Brendan Rappazzo, @fernbear.bsky.social, Boza Vlado, You Jiacheng, Franz Cesista, Braden Koszarsky, and @Grad62304977. modded-nanogpt: Speedrunning the nanogpt baseline, 2024. URL https://github.com/KellerJordan/modded-nanogpt.

[35] 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, 2024.

[36] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. *arXiv preprint arXiv:2310.02279*, 2024.

[37] Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *arXiv preprint arXiv:2404.07724*, 2024.

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

[39] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Artificial intelligence and statistics*, pages 562–570. Pmlr, 2015.

[40] Kyungmin Lee, Sihyun Yu, and Jinwoo Shin. Decoupled meanflow: Turning flow models into flow maps for accelerated sampling. *arXiv preprint arXiv:2510.24474*, 2025.

[41] 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. *arXiv preprint arXiv:2504.10483*, 2025.

[42] Baiqi Li, Zhiqiu Lin, Deepak Pathak, Jiayao Li, Yixin Fei, Kewen Wu, Tiffany Ling, Xide Xia, Pengchuan Zhang, Graham Neubig, et al. Genai-bench: Evaluating and improving compositional text-to-visual generation. *arXiv preprint arXiv:2406.13743*, 2024.

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

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

[45] Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. *arXiv preprint arXiv:2410.11081*, 2024.

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

[47] Yiyang Ma, Xingchao Liu, Xiaokang Chen, Wen Liu, Chengyue Wu, Zhiyu Wu, Zizheng Pan, Zhenda Xie, Haowei Zhang, Xingkai Yu, et al. Janusflow: Harmonizing autoregression and rectified flow for unified multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7739–7751, 2025.

[48] 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 Journal*, pages 1–31, 2024.

[49] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[50] Pablo Pernias, Dominic Rampas, Mats Leon Richter, Christopher Pal, and Marc Aubreville. Würstchen: An efficient architecture for large-scale text-to-image diffusion models. In *The Twelfth International Conference on Learning Representations*, 2023.

[51] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[52] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[53] Axel Sauer, Katja Schwarz, and Andreas Geiger. Projected gans converge faster. In *Advances in Neural Information Processing Systems*, volume 34, pages 17480–17492, 2021.

[54] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022.

[55] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023.

[56] Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. GNM: A general navigation model to drive any robot. In *International Conference on Robotics and Automation (ICRA)*, 2023. URL https://arxiv.org/abs/2210.03370.

[57] Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. ViNT: A foundation model for visual navigation. In *Conference on Robot Learning (CoRL)*, 2023. URL https://arxiv.org/abs/2306.14846.

[58] 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.

[59] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025. URL https://arxiv.org/abs/2508.10104.

[60] Jaskirat Singh, Xingjian Leng, Zongze Wu, Liang Zheng, Richard Zhang, Eli Shechtman, and Saining Xie. What matters for representation alignment: Global information or spatial structure? *arXiv preprint arXiv:2512.10794*, 2025.

[61] Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. *arXiv preprint arXiv:2310.14189*, 2023.

[62] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

[63] Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. NoMaD: Goal masked diffusion policies for navigation and exploration. *arXiv preprint arXiv:2310.07896*, 2023. URL https://arxiv.org/abs/2310.07896.

[64] Keqiang Sun, Junting Pan, Yuying Ge, Hao Li, Haodong Duan, Xiaoshi Wu, Renrui Zhang, Aojun Zhou, Zipeng Qin, Yi Wang, Jifeng Dai, Yu Qiao, Limin Wang, and Hongsheng Li. Journeydb: A benchmark for generative image understanding. In *Advances in Neural Information Processing Systems*, 2023.

[65] Shitao Tang, Fuyang Chen, Aviral Joshi Bose, Kun Zhang, Shengjie Bi, Zhenyu Wang, Jiacheng Feng, Haozhe Cai, Alexander Liu, Antonio Torralba, et al. Ardit: Autoregressive diffusion transformers. *arXiv preprint arXiv:2406.00843*, 2024.

[66] 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.

[67] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

[68] Runqian Wang and Kaiming He. Diffuse and disperse: Image generation with representation regularization. *arXiv preprint arXiv:2506.09027*, 2025.

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

[70] Wikipedia contributors. Dimensionality reduction – Wikipedia, the free encyclopedia, 2026. URL https://en.wikipedia.org/wiki/Dimensionality_reduction. [Online; accessed April 2026].

[71] Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, Jian Yang, Ming-Ming Cheng, and Xiang Li. Representation entanglement for generation: Training diffusion transformers is much easier than you think. *arXiv preprint arXiv:2507.01467*, 2025.

[72] Haoyu Wu, Diankun Wu, Tianyu He, Junliang Guo, Yang Ye, Yueqi Duan, and Jiang Bian. Geometry forcing: Marrying video diffusion and 3d representation for consistent world modeling. *arXiv preprint arXiv:2507.07982*, 2025.

[73] Saining Xie. RIP REPA. X (formerly Twitter), 2025. URL https://x.com/sainingxie/status/1977936727839736189.

[74] Wanghan Xu, Xiaoyu Yue, Zidong Wang, Yao Teng, Wenlong Zhang, Xihui Liu, Luping Zhou, Wanli Ouyang, and Lei Bai. Exploring representation-aligned latent space for better generation. *arXiv preprint arXiv:2502.00359*, 2025.

[75] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

[76] Jiawei Yang, Zhengyang Geng, Xuan Ju, Yonglong Tian, and Yue Wang. Representation fréchet loss for visual generation. *arXiv preprint arXiv:2604.28190*, 2026.

[77] Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. *arXiv preprint arXiv:2501.01423*, 2025.

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

[79] 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.

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

[81] Huijie Zhang, Aliaksandr Siarohin, Willi Menapace, Michael Vasilkovsky, Sergey Tulyakov, Qing Qu, and Ivan Skorokhodov. Alphaflow: Understanding and improving meanflow models. *arXiv preprint arXiv:2510.20771*, 2025.

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

[83] Shilong Zhang, He Zhang, Zhifei Zhang, Chongjian Ge, Shuchen Xue, Shaoteng Liu, Mengwei Ren, Soo Ye Kim, Yuqian Zhou, Qing Liu, Daniil Pakhomov, Kai Zhang, Zhe Lin, and Ping Luo. Both semantics and reconstruction matter: Making representation encoders ready for text-to-image generation and editing. *arXiv preprint arXiv:2512.17909*, 2025.

[84] Xiangdong Zhang, Jiaqi Liao, Shaofeng Zhang, Fanqing Meng, Xiangpeng Wan, Junchi Yan, and Yu Cheng. Videorepa: Learning physics for video generation through relational alignment with foundation models. *arXiv preprint arXiv:2505.23656*, 2025.

[85] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. arXiv preprint arXiv:2510.11690, 2025. URL https://arxiv.org/abs/2510.11690.

[86] Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *arXiv preprint arXiv:2306.09305*, 2023.

[87] Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. *arXiv preprint arXiv:2503.07565*, 2025.

[88] Xingyu Zhou, Qifan Li, Xiaobin Hu, Hai Chen, and Shuhang Gu. Guiding a diffusion transformer with the internal dynamics of itself. *arXiv preprint arXiv:2512.24176*, 2025.

[89] Chenchen Zhu, Saksham Suri, Cijo Jose, Maxime Oquab, Marc Szafraniec, Wei Wen, Yunyang Xiong, Patrick Labatut, Piotr Bojanowski, Raghuraman Krishnamoorthi, and Vikas Chandra. Efficient universal perception encoder. *arXiv preprint arXiv:2603.22387*, 2026.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We support the claims made in the abstract and introduction with extensive experiments in §3 and the appendix.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.

- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in §G.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: N/A.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide the full implementation details in §A.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We will open-source the code and trained checkpoints upon publication.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.

- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the full experimental setting and training details in §A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Following standard practice in prior work, we report gFID and Inception Score on ImageNet. Following recent work [76], we additionally report FD^6_r, which aggregates across six different representation feature spaces. We further validate the approach across diverse tasks, including text-to-image generation and navigation world models.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

Justification: We report compute setup and wall-clock training time in §A.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work focuses on foundational generative-model research, proposing an improved baseline for using pretrained vision encoders for generative models.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper uses existing public datasets and pretrained vision encoders for training diffusion models.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We credit all existing assets (datasets, pretrained vision encoders, baselines) via citations to the original works.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: We provide supplementary material with all details. Code and checkpoints will be released upon publication.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: N/A.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: N/A.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: We describe LLM usage (figure generation, related-work search, paper writing) in §H.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

Configuration	ImageNet 256×256	Text-to-Image	World Models
Architecture			
Backbone	DiT^{DH}-XL	DiT^{DH}-XL	DiT^{DH}-XL
Encoder blocks / Hidden dim / Heads	28 / 1152 / 16	28 / 1152 / 16	28 / 1152 / 16
Decoder blocks / Hidden dim / Heads	2 / 2048 / 16	2 / 2048 / 16	2 / 2048 / 16
MLP ratio	4.0	4.0	4.0
Patch size (latent)	1	1	1
Input channels	768	768	768
Conditioning	In-context	In-context	In-context
Conditioning tokens	4 + 8	4 + 256	1029
Positional embedding	APE + RoPE	APE + RoPE	APE + RoPE
Normalization	RMSNorm	RMSNorm	RMSNorm
FFN activation	SwiGLU	SwiGLU	SwiGLU
RAE Encoder			
Vision encoder	DINOv3-L	SiGLIP2-B	DINOv3-L
Encoder input resolution	256	224	256
Encoder patch size	16	16	16
Latent shape	$1024 \times 16 \times 16$	$768 \times 16 \times 16$	$1024 \times 16 \times 16$
Encoder normalization	Layer norm	Layer norm	Layer norm
REPA			
Target encoder	Same as RAE encoder	Same as RAE encoder	Same as RAE encoder
Alignment layer depth	8	8	8
Projection type	Linear	Linear	Linear
REPA coefficient (λ)	0.5	0.5	0.5

Table 10: **Architecture and model configurations.** Model architecture, RAE encoder, and REPA settings for class-conditional ImageNet 256×256, text-to-image, and navigation world models. All settings share the same backbone and differ primarily in the conditioning. Continued in Tab. 11.

A Implementation Details

We provide detailed implementation configurations for reproducibility. Tab. 10 summarizes all hyperparameters for both class-conditional ImageNet and text-to-image experiments.

Architecture. We use the DDT [69] backbone (DiT^{DH}-XL), which consists of a 28-block transformer encoder with hidden dimension 1152 and 16 attention heads, followed by a 2-block DDT decoder with hidden dimension 2048. The total model has 839M parameters. All layers use RMSNorm, SwiGLU activation in the feed-forward network (MLP ratio 4.0), and rotary positional embeddings (RoPE) combined with absolute positional embeddings (APE). The latent patch size is 1, producing a sequence of $16 \times 16 = 256$ tokens from the encoder output.

RAE encoder. For ImageNet and navigation world model experiments, we use DINOv3-L [59] as the default encoder. The encoder processes 256×256 images with patch size 16, producing $16 \times 16 = 256$ patch tokens of dimension 1024, giving a latent representation of shape $1024 \times 16 \times 16$. For text-to-image experiments, we use SiGLIP2-B [80] following [66], with the same 16×16 patch grid and a feature dimension of 768. We discard [CLS] and register tokens and apply layer normalization to the patch outputs. The RAE decoder is pretrained separately for 16 epochs following [85] and kept frozen during diffusion training.

Vision encoders. We evaluate 21 encoder variants across 8 families following [60]: DINOv2 [48], DINOv3 [59], WebSSL [16], Perception Encoders [6], MoCov3 [15], CLIP [51], I-JEPA [2], and MAE [26]. Each encoder is wrapped in a unified interface that extracts patch tokens, discards any [CLS] or register tokens, and applies layer normalization. The full encoder-sweep results, comparing RAE and RAEv2 on every variant, are reported in Tab. 12.

REPA configuration. We apply representation alignment at encoder block depth 8 with a single linear projection layer mapping from the transformer hidden dimension (1152) to the target encoder dimension (768). The REPA loss coefficient is set to $\lambda = 0.5$ following [79, 60]. The target encoder is the same as the RAE encoder (self-REPA), as we show in §2.2 that this consistently improves generation across the full set of pretrained encoders we evaluate.

Configuration	ImageNet 256×256	Text-to-Image	World Models
Training			
Dataset	ImageNet-1K	JourneyDB + BLIP3o	RECON
Base learning rate	2×10^{-4}	2×10^{-4}	2×10^{-4}
Final learning rate	2×10^{-5}	2×10^{-5}	2×10^{-5}
LR schedule	Linear decay	Linear decay	Linear decay
Warmup epochs / iterations	25 epochs	50K iter	25K iter
Warmup decay end (LR reaches final)	50 epochs	150K iter	60K iter
Weight decay	0.0	0.0	0.0
Global batch size	1024	1024	256
Mixed precision	bfloat16	bfloat16	bfloat16
Gradient clipping (max norm)	1.0	1.0	1.0
EMA decay	0.9995	0.9995	0.9995
Training epochs / iterations	80	150K iter (pretrain) + 50 ep (finetune)	100K iter
CFG dropout probability	0.1	0.1	–
Flow Matching			
Base prediction type	x-prediction	x-prediction	x-prediction
REPA head prediction type	x-prediction	x-prediction	x-prediction
Time distribution	Logit-normal	Logit-normal	Logit-normal
Sampling			
Sampler	ODE (Euler)	ODE (Euler)	ODE (Euler)
Number of steps	50	50	50
Guidance interval	$[0.0, 1.0]$	$[0.0, 1.0]$	–
Text Conditioning (T2I only)			
Text encoder	–	Qwen3-0.6B	–
Max sequence length	–	256	–
Finetuning dataset	–	BLIP3o-60k	–

Table 11: **Training and sampling configurations (continued).** Training hyperparameters, flow matching, sampling, and text conditioning settings. Continuation of Tab. 10.

Encoder	Encoder properties			gFID (DiT^{DH}-XL @ 20ep) ↓	
	LP ↑	LDS ↑	Avg(LP', LDS') ↑	RAE	RAEv2 (k=1)
MoCov3-B [15]	76.4	0.15	0.46	13.84	8.35
CLIP-L [51]	84.5	0.14	0.49	7.85	4.38
PE-L [6]	85.5	0.14	0.50	7.06	4.09
PE-B [6]	80.7	0.20	0.50	6.22	3.88
WebSSL-1B [16]	84.1	0.18	0.51	8.60	4.16
LangPE-L [6]	83.0	0.20	0.52	5.04	–
SpatialPE-B [6]	70.8	0.33	0.52	11.24	5.04
JEPA-H [2]	77.5	0.33	0.55	12.46	4.48
SpatialPE-L [6]	78.4	0.34	0.56	8.77	3.97
DINOv3-B [59]	84.5	0.38	0.61	4.25	2.76
DINOv2-B [48]	83.9	0.41	0.62	3.75	2.81
DINOv3-L [59]	**87.0**	**0.42**	**0.65**	**3.30**	**2.61**

Table 12: **Full ablation on choice of pretrained vision encoder.** Extended version of Tab. 2. gFID at 20 epochs (DiT^{DH}-XL), sorted by the composite score Avg(LP', LDS'). LP denotes ImageNet linear-probing accuracy (a measure of global semantic quality), and LDS denotes the local-distance similarity score from iREPA [60] (a measure of spatial structure); LP' and LDS' are the min-max normalized values used to form the composite score. RAEv2 consistently improves over RAE across all encoder families. Stronger encoders (e.g., DINOv3-L) which excel at both global and spatial performance achieve the best generation quality. All results are reported without guidance and at batch size 1024.

Conditioning. We replace adaLN-Zero [49] with in-context conditioning. The timestep is embedded via Gaussian Fourier features into 4 tokens, and the class label (or text) is embedded into 8 tokens (or up to 256 tokens for T2I). These are concatenated with the image token sequence and processed

jointly through self-attention. The DDT decoder strips the conditioning tokens before producing the final output corresponding to the 256 image latent tokens.

Training. We use a base learning rate of 2×10^{-4}, linearly decayed to 2×10^{-5} by epoch 50, with 25 epochs of warmup, and no weight decay. Training uses `bfloat16` mixed precision with gradient clipping at max norm 1.0. We apply EMA with decay 0.9995 and report all results using the EMA model. The global batch size is 1024. For classifier-free guidance training, we drop the class label with probability 0.1.

Flow matching. We use continuous-time flow matching with velocity prediction and logit-normal time sampling following [46]. For self-guidance (§2.3), we convert the model output to x-prediction at inference time and apply guidance via the REPA head prediction as defined in Eq. 3.

Sampling and evaluation. We use the ODE solver with Euler discretization and 50 sampling steps for all experiments. We follow the online evaluation protocol from [43] and report gFID [28] and Inception Score (IS) [52] on 50K generated images. Following recent work, we additionally report FD_r [76] computed across six representation feature spaces (Inception, ConvNeXt, DINOv2, MAE, SigLIP, CLIP), and the geometric mean FD_r^6. As a measure of training efficiency, we report $\text{EP}_{\text{FID@}k}$, the number of training epochs to reach unguided gFID $\leq k$ (we report k=2). We generate 50 images per class (balanced sampling) following [85].

Text-to-image. We adapt the DiT^{DH}-XL backbone for T2I generation. We follow the same incontext architecture from ImageNet experiments (§3), replacing the 8 incontext class-conditional embedding tokens with 256 text-embedding tokens for input captions encoded by Qwen3-0.6B [75]. We pretrain on JourneyDB [64] together with the long-caption and short-caption subsets of BLIP3o [13] for 150K iterations at batch size 1024, and then finetune on BLIP3o-60k for 50 epochs at the same batch size. We evaluate on GenEval [23], DPG-Bench [30], and GenAI-Bench [42].

Navigation world models. We use the same DiT^{DH}-XL backbone as in the ImageNet and T2I settings, only altering the conditioning tokens to handle navigation inputs. The model conditions on $N = 4$ past frames at 256×256 resolution; each frame is encoded by the RAE encoder into a 16×16 patch grid, giving $N \times 256 = 1024$ context tokens. We additionally append 4 action tokens (encoding the egocentric action $(\Delta x, \Delta y, \Delta\psi)$) and a single Fourier-embedded time token for the rollout offset, for a total of 1029 conditioning tokens (compared to 8 for class-conditional ImageNet and 256 for T2I). Following [56, 57, 63], we use the RECON [4, 63] dataset with the same flow-matching, learning-rate schedule, and EMA recipe as our ImageNet experiments. We train for 100K iterations at batch size 256. For evaluation, following [4], we evaluate predicted frames against ground truth at horizons of $\{1, 2, 4, 8, 16\}$ seconds. Given an FPS of f, we obtain the prediction at a target horizon of T seconds via $T \cdot f$ autoregressive rollout steps: at each step the model predicts the next frame conditioned on the current sliding window of N context frames and the next ground-truth action, and the predicted RGB is re-encoded and fed back as context. We report FID [28], LPIPS [82], PSNR, and DreamSim [19] at each horizon, computed over rollout episodes sampled from the held-out RECON [63] validation split.

Per-layer feature visualization. Fig. 11 visualizes features and spatial self-similarity across the layers of a pretrained DINOv2-B encoder. Different depths capture complementary information: later layers emphasize global semantics, while earlier-to-middle layers retain finer spatial structure. This motivates the generalized RAE formulation in §2.1, which aggregates features across the last K layers rather than relying solely on the final layer.

Compute. We report results using a 4×8 H100 setup, which trains RAEv2 to gFID 1.06 in roughly 12 hours, compared to over a week for the original RAE under the same hardware.

B Extended Related Work

We provide a more comprehensive discussion of related work extending §4.

Representation alignment for generation. REPA [79] aligns intermediate DiT features with pretrained encoders (*e.g.* DINOv2) via a projection head, accelerating convergence. iREPA [60] showed that spatial structure in the alignment target matters more than global information. REPA-E [41] extends this to end-to-end VAE tuning. Orthogonally, RAE [85] replaces the VAE latent space

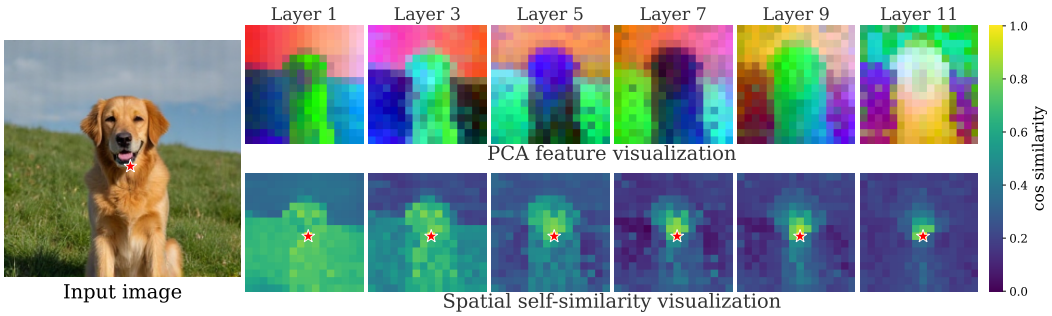


Figure 11: **Different layers of a pretrained encoder provide complementary features.** Aggregating across layers yields richer representations than using the final layer alone.

entirely with pretrained encoder features. LDiT [77] studies the tension between reconstruction and generation in the latent space. A common assumption is that RAE subsumes REPA since both use the same encoder. We show this is incorrect; RAE and REPA provide complementary benefits.

Fast-Forward Generative Models.

A major research thrust aims to reduce the number of sampling steps required by diffusion models. Consistency Models [62] learn to map any point on the probability flow ODE trajectory directly to its origin, enabling one-step generation. Improved Consistency Training (iCT) [61] and Simplified Consistency Training (sCT) [45] refined the training procedure for better stability and scalability. Consistency Trajectory Models (CTM) [36] generalized this by learning the full trajectory, allowing flexible trade-offs between speed and quality. Shortcut Models [18] take a different approach by conditioning the denoising network on the desired step size, enabling a single model to handle variable-step generation.

In the flow matching setting, MeanFlow [21] proposed training a one-step generator by matching the mean velocity field, while iMF [22] identified and addressed key challenges in this framework including training instability and mode collapse. AlphaFlow [81] and Decoupled MeanFlow [40] further refined the mean flow framework. Inductive Moment Matching (IMM) [87] offers an alternative by matching higher-order moments for few-step generation. Our work is complementary to these fast-forward methods; we focus on improving the latent space and training recipe, which can be combined with any of these distillation or one-step techniques without modification.

Guidance Mechanisms.

Classifier-free guidance (CFG) [29] has become the standard technique for improving sample quality at the cost of diversity, by interpolating between conditional and unconditional predictions. CFG Interval [37] showed that applying guidance only during specific noise levels improves both sample and distribution quality. Autoguidance [35] replaced the unconditional model with a weaker conditional model, demonstrating that guidance fundamentally works by contrasting a stronger model against a weaker one rather than requiring unconditional training.

Self-Representation Alignment (SRA) [33] showed that diffusion transformers can provide representation guidance by themselves, using intermediate features to steer generation without external models. The dispersive loss [68] regularizes representations during diffusion training itself to improve generation quality. Internal dynamics guidance [88] further explores how a model's own internal representations can substitute for external guidance signals. Recent work on guidance-free generation [12] aims to eliminate the need for guidance entirely by incorporating its benefits into training. Our improved self-guidance approach relates to SRA and autoguidance: we show that the REPA prediction head, when combined with x-prediction, can serve as an internal guidance signal, avoiding both the unconditional forward pass of CFG and the separate weaker model of autoguidance.

Representation Learning and Generation.

The intersection of representation learning and image generation has a rich history. Würstchen [50] demonstrated that operating in a highly compressed semantic latent space (rather than pixel-level

VAE latents) enables efficient large-scale text-to-image synthesis. This insight is closely related to the RAE paradigm of using pretrained encoder features as the diffusion latent space.

Several recent works explore how to best construct latent spaces that serve both reconstruction and semantic tasks. FAE [20] proposes single-layer adaptation of pretrained features for latent diffusion, showing that minimal fine-tuning of a frozen encoder can yield effective generation latents. MAETok [11] uses a masked autoencoder tokenizer to bridge self-supervised features and discrete token-based generation. FlatDINO [7] compresses DINOv2 patch features into flatter distributions better suited for diffusion training. ReaLS [74] injects semantic priors from pretrained models into the VAE latent space, while SVG [58] directly uses frozen DINO features as the generation target.

Unified Latents [27] jointly trains the encoder, diffusion prior, and decoder with MSE regularization, showing that end-to-end optimization of the full latent pipeline can improve over separately trained components. PS-VAE [83] addresses the tension between semantic richness and pixel-level reconstruction by training representation encoders that excel at both, making them ready for text-to-image generation and editing. These works share a common theme with our approach: the latent space is not merely a compression bottleneck but an active design choice that shapes generation quality, training efficiency, and downstream flexibility.

Several works have extended representation alignment beyond static image generation. VideoREPA [84] applies relational alignment with foundation models to video generation, while Geometry Forcing [72] marries video diffusion with 3D representations. JanusFlow [47] unifies multimodal understanding and generation through shared representations and rectified flow.

From the discriminative side, using pretrained features for generation has shown strong results. ProjectedGAN [53] and StyleGAN-XL [54] leveraged pretrained feature networks as discriminators, achieving faster GAN convergence on diverse datasets. Adversarial diffusion distillation [55] combined diffusion training with an adversarial loss using pretrained discriminator features. SODA [31] studied bottleneck diffusion models and showed that the denoising objective itself learns useful representations, connecting generation and self-supervised learning. These works collectively show that pretrained representations and generation are deeply intertwined, a theme central to our investigation of RAE and REPA.

Transformer Architectures for Diffusion.

DiT [49] introduced vision transformers with adaptive layer normalization (adaLN-Zero) for class-conditional diffusion, demonstrating strong scaling properties. SiT [46] extended this to the flow matching framework with scalable interpolant transformers. U-ViT [3] showed that a simpler ViT backbone with long skip connections also works well for diffusion. GenTron [14] and MaskDiT [86] explored architectural variants for improved efficiency and generation quality. ARDiT [65] and DART [25] bridged autoregressive and diffusion approaches within the transformer framework.

A key architectural choice is how to inject conditioning information. DiT's adaLN-Zero modulates layer statistics via scale and shift parameters, which has become the dominant approach. An alternative is in-context conditioning, where condition tokens are concatenated with the input sequence and processed jointly through self-attention. In our work, we adopt in-context conditioning for the class/text embeddings, which we find works well with the RAE latent space and simplifies the architecture compared to adaLN-based designs.

C Reconstruction vs Generation Comparison

Tab. 13 compares RAEv2 against recent representation-based autoencoders (§4) that target improved reconstruction. All prior works rely on auxiliary losses, encoder fine-tuning, or architectural modifications to the pretrained encoder; in contrast, our generalized RAE formulation (MLS) is strictly training-free, yet simultaneously achieves the best generation quality (at K=7) and the best reconstruction quality (at K=23).

Impact of additional decoder training data for reconstruction. Tab. 14 reports reconstruction performance for the RAEv2 decoder trained on ImageNet and with additional training data from [66]. Training with more data consistently improves reconstruction. Note: All results are reported with training for only 16 epochs. Training with more data (similar to proprietary VAEs) and for longer can help further improve reconstruction performance.

Encoder	Training-free Encoder	Recon-Gen Tradeoff Control	Epochs	Stage 1		Stage 2	
				rFID↓	PSNR↑	gFID↓	IS↑
DINO-Tok [32]	✗	✗	80	0.32	**28.54**	5.94	152.6
DINO-SAE [8]	✗	✗	80	0.37	26.20	3.07	209.7
VFM-VAE [5]	✗	✗	80	0.52	–	3.41	160.4
AlignTok [9]	✗	✗	80	0.26	25.83	3.71	148.9
RPiAE [24]	✗	✗	80	0.50	21.30	2.25	208.7
RAE [85]	✓	✗	80	0.602	18.93	2.23	214.8
RAEv2 (K=7, **ours**)	✓	✓	80	0.29	22.57	**1.65**	**228.0**
RAEv2 (K=23, **ours**)	✓	✓	80	**0.18**	27.03	3.02	206.0

Table 13: **Reconstruction vs Generation Comparison.** RAEv2 in its generalized form improves both reconstruction and generation performance over recent representation-based autoencoders [44, 83, 17, 32, 8, 5, 9, 24] *without* fine-tuning the pretrained encoder. Furthermore, by simply varying the value of K (the number of last-layer features aggregated), the generalized formulation provides an easy way to control the reconstruction–generation trade-off; on this benchmark RAEv2 achieves the best generation quality at K=7 and the best reconstruction quality at K=23.

Decoder	PSNR ↑	SSIM ↑	LPIPS ↓	rFID ↓
DINOv3-L (K=7)	22.58	0.6257	0.1531	0.299
+ more data	24.18	0.6946	0.1209	0.276
DINOv3-L (K=23)	27.04	0.8062	0.0874	0.185
+ more data	29.13	0.8625	0.0654	0.158

Table 14: **Impact of additional data on RAEv2 decoder training.** Results with and without training on additional data [66] for decoder training. Training with more data consistently improves reconstruction. Note: All results are reported with training for only 16 epochs with frozen pretrained vision encoder. Training with more data (similar to proprietary VAEs) and for longer can help further improve reconstruction performance.

D Additional Ablation Results

We provide extended versions of the ablation tables condensed in the main paper.

Component-wise ablation. Tab. 15 isolates the contribution of each component of RAEv2. While x-prediction alone gives a modest improvement, it enables REPA-head guidance (§2.3), which yields the largest gain without an extra forward pass (CFG) or a separately trained weaker model (AutoGuidance). Self-REPA consistently improves generation, reinforcing the complementary roles of RAE and REPA (§2.2). The generalized RAE formulation (K=7) substantially improves reconstruction, and improves generation once combined with REPA-head guidance.

Configuration	rFID ↓	gFID ↓
RAE	0.61	2.20
+ x-prediction	0.61	2.08
+ self-REPA	0.61	1.50
+ generalized RAE (K=7)	0.29	1.65
+ REPA-head guidance (RAEv2)	**0.29**	**1.06**

Table 15: **Component-wise ablation.** Starting from RAE, we add each component of RAEv2 in turn. All results use the DINOv3-L encoder, DDT-XL, 80 epochs, and batch size 1024 on ImageNet 256×256.

Generalized RAE formulation. Tab. 16 extends Tab. 3 (main paper) with all swept $K \in \{2, 4, 6, 8\}$ values and all five reconstruction/generation metrics (PSNR, SSIM, rFID, gFID, IS). MLS consistently dominates MLR on Stage-2 generation (gFID) across every K, while the two methods are essentially tied on the Stage-1 reconstruction metrics.

Encoder Layers (last K)	Method	Stage-1 metrics			Stage-2 metrics	
		PSNR ↑	SSIM ↑	rFID ↓	gFID ↓	IS ↑
2	MLR	19.72	0.509	0.570	3.085	–
	MLS	19.44	0.502	0.532	**2.586**	**243.6**
4	MLR	20.86	0.558	0.425	2.954	–
	MLS	20.50	0.545	0.435	**2.622**	**230.9**
6	MLR	21.97	0.607	0.342	3.118	–
	MLS	21.92	0.605	0.336	**2.637**	**223.3**
8	MLR	23.36	0.669	0.268	3.580	–
	MLS	23.30	0.663	0.264	**2.688**	**220.8**

Table 16: **Full ablation on formulation for generalized RAE.** Extended version of Tab. 3. We compare two parameter-free ways of combining the last K encoder layers (§2.1): **MLS** (multi-layer sum) is a simple addition $\boldsymbol{x} = \sum_\ell \boldsymbol{z}_\ell$; **MLR** (multi-layer random projection) concatenates the layers and projects back with a fixed random matrix. Encoder is DINOv3-L (24 layers); Stage-1 reports decoder reconstruction (PSNR, SSIM, rFID); Stage-2 reports DiT^{DH}-XL training (gFID, IS at 20 epochs). MLS dominates MLR on Stage-2 gFID at every K, while the two are essentially tied on Stage-1 reconstruction.

Guidance mechanism. Tab. 17 extends Tab. 4 (main paper) with the additional Inception Score (IS) column for both K=7 and K=23. REPA Guidance achieves the best gFID and IS in both configurations while requiring no separate model and no extra forward pass.

Guidance	RAEv2 (K=7)		RAEv2 (K=23)	
	gFID ↓	IS ↑	gFID ↓	IS ↑
w/o Guidance	1.65	228.0	3.01	206.0
CFG [29]	1.49	242.1	2.83	220.1
Autoguidance (AG) [35]	1.14	255.3	1.37	252.0
REPA Guidance (Ours)	**1.06**	**255.3**	**1.25**	**256.8**

Table 17: **Full ablation on guidance mechanism in RAEv2.** Extended version of Tab. 4, additionally reporting Inception Score (IS). We compare four guidance options for RAEv2 across two encoder-layer aggregation choices (K=7 and K=23). REPA Guidance (§2.3) achieves the best gFID and IS while requiring no additional model (unlike AG) and no extra forward pass (unlike CFG). DiT^{DH}-XL backbone with DINOv3-L encoder.

Impact of generalized RAE on understanding (linear probing). A key advantage of RAE is that it provides a unified tokenization for both understanding and generation. While the generalized RAE formulation greatly improves both reconstruction and generation performance (§2.1), it is important to understand its impact on the encoder's understanding performance (linear probing). We compare the original DINOv3-L final-layer encoder (K=1) against the generalized multi-layer-sum (MLS) variants used in RAEv2 (K=7 and K=23) in Tab. 18. Despite significantly improving reconstruction and generation performance, the generalized RAE formulation does not meaningfully degrade the encoder's understanding performance, as measured by linear probing accuracy on ImageNet.

Encoder	Feature dim	LP top-1 (%) ↑
DINOv3-L (K=1, last layer)	1024	85.39
DINOv3-L MLS (K=2)	1024	85.29
DINOv3-L MLS (K=3)	1024	85.28
DINOv3-L MLS (K=4)	1024	85.15
DINOv3-L MLS (K=5)	1024	85.13
DINOv3-L MLS (K=6)	1024	85.14
DINOv3-L MLS (K=7)	1024	85.10
DINOv3-L MLS (K=8)	1024	85.10
DINOv3-L MLS (K=9)	1024	85.10
DINOv3-L MLS (K=10)	1024	85.12
DINOv3-L MLS (K=23, full)	1024	85.24

Table 18: **Impact of generalized RAE on understanding (linear probing).** Linear probing top-1 accuracy on ImageNet across all $K \in \{1, \ldots, 10, 23\}$ for the generalized multi-layer-sum (MLS) variant on DINOv3-L. K=1 corresponds to the original RAE (final-layer feature). The generalized formulation (§2.1) improves reconstruction without meaningfully impacting global semantic performance, enabling unified tokenization for both understanding and generation. All values are computed at 30 epochs of LP training with learning rate 1×10^{-2}; continued training may further improve linear probing scores.

E Generalization to Other Tasks

E.1 Text-to-Image Generation

Training setup. We pretrain a text-to-image model from scratch on JourneyDB [64] together with the long-caption and short-caption subsets of BLIP3o [13], for 150K iterations at a global batch size of 1024. Following [66], we use SiGLIP2-B [80] as the RAE encoder and adapt the DiT^{DH}-XL backbone for text-conditioning. Text captions are encoded by Qwen3-0.6B [75] with a maximum sequence length of 256 tokens. Optimization mirrors the ImageNet recipe (lr 2×10^{-4} linearly decayed to 2×10^{-5}, `bfloat16`, EMA decay 0.9995). We then finetune on the BLIP3o-60k subset for 50 epochs at the same batch size.

Evaluation. Following [66], we report results on GenEval [23], DPG-Bench [30], and GenAI-Bench [42], covering compositional, dense-prompt, and human-preference axes. Samples are generated with the ODE (Euler) sampler at 50 steps using the EMA model.

Pretraining. Results are shown in Tab. 19. We observe that as compared to widely used Flux-VAE [38], the use of representation autoencoders leads to significant improvements for text-to-image generation. Furthermore, using the improved training recipe leads to even further gains across all evaluation metrics. For instance, while Flux-VAE and RAE lead to a GenEval score of 41.7 and 58.4 respectively, the use of improved baseline RAEv2 leads to better performance with GenEval score of 62.4.

Finetuning. Following [66], we also perform finetuning of our pretrained model using the 60k finetuning dataset from BLIP3o [13]. We use a batch size of 1024 and 50 epochs for finetuning. Similar to findings of [66], we find that this helps significantly increase the performance to 82.7 on GenEval with RAEv2. Furthermore, while finetuning reduces the gap between various methods, RAEv2 still shows improved performance over original RAE and Flux-VAE.

E.2 Navigation World Models

We follow the navigation world modeling setup of NWM [4]. In this setting, the model is conditioned on the last $N = 4$ egocentric video frames together with an action sequence, and is trained to predict the next frame in the trajectory. At inference time, the model rolls out future frames *autoregressively*: at each step, the predicted frame is fed back into the context window so that long-horizon predictions can be produced from a short history.

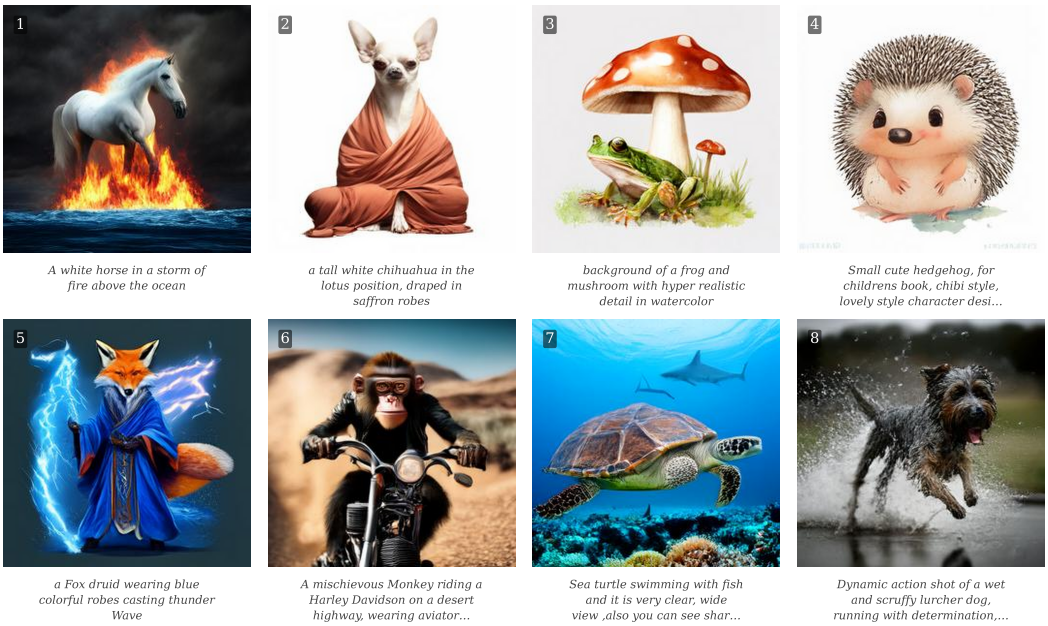


Figure 12: **Text-to-image qualitative samples at 256×256.** Qualitative samples from RAEv2 (0.9B) trained for 100K iterations with batch size 1024 (equivalent to ∼80 epochs on ImageNet at the same batch size), evaluated on MJHQ test set prompts using self-guidance via the REPA head. Additional samples and full prompts are provided in Fig. 16–Fig. 21.

Method	Model	Params	GenEval↑	GenAI-Bench↑	DPG-Bench↑
Pretraining					
Flux-VAE [38]	DiT^{DH}-XL	0.9B	41.7	57.3	77.6
RAE [85]	DiT^{DH}-XL	0.9B	58.4	63.2	80.1
RAEv2	DiT^{DH}-XL	0.9B	**62.4**	**63.8**	**81.7**
Finetuning					
Flux-VAE [38]	DiT^{DH}-XL	0.9B	78.3	63.9	79.2
RAE [85]	DiT^{DH}-XL	0.9B	81.5	67.2	80.6
RAEv2	DiT^{DH}-XL	0.9B	**82.7**	**68.0**	**82.3**

Table 19: **Text-to-image generation.** Results comparing proposed RAEv2 with original RAE [85] and Flux-VAE [38]. Results for pretraining are reported at 150K steps with batch-size of 1024 and JourneyDB, long-caption and short-caption subsets from BLIP3o pretraining subset [13]. For finetuning similar to [66], we use the 60k subset from [13], and 1024 batchsize. Across all settings, we observe that RAEv2 leads to faster training over original RAE and Flux-VAE.

Training setup. We use the same DiT^{DH}-XL backbone as in the previous sections, only altering the conditioning tokens to handle navigation inputs. The model conditions on $N = 4$ past frames at 256×256 resolution; each frame is encoded by the RAE encoder into a 16×16 patch grid, giving $N \times 256 = 1024$ context tokens. We additionally append 4 action tokens (encoding the egocentric action $(\Delta x, \Delta y, \Delta\psi)$) and a single Fourier-embedded time token for the rollout offset, for a total of 1029 conditioning tokens (compared to 8 for class-conditional ImageNet and 256 for T2I). Following [56, 57, 63], we use the RECON [4, 63] dataset with the same flow-matching, learning-rate schedule, and EMA recipe as our ImageNet experiments. We train for 100K iterations at a batch size of 256, on the same 4×8 H100 setup used for the ImageNet experiments.

Evaluation. Following [4], we evaluate predicted frames against ground truth at horizons of $\{1, 2, 4, 8, 16\}$ seconds. Given an FPS of f, we obtain the prediction at a target horizon of T

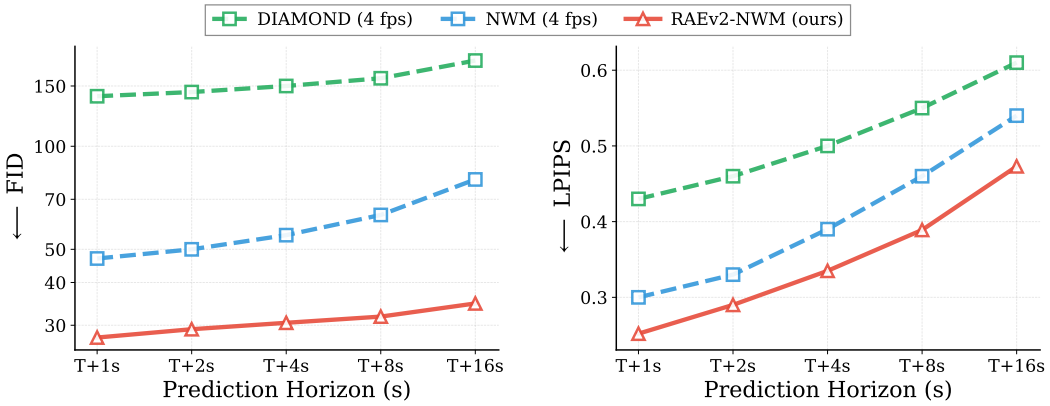


Figure 13: **Future state prediction across rollout horizons.** Comparing generation accuracy and quality of NWM [4] and DIAMOND [1] at 1 and 4 FPS as function of time, up to 16 seconds of generated video on the RECON dataset.

seconds via $T \cdot f$ autoregressive rollout steps: at each step the model predicts the next frame conditioned on the current sliding window of N context frames and the next ground-truth action, and the predicted RGB is re-encoded and fed back as context. We report FID [28], LPIPS [82], PSNR, and DreamSim [19] at each horizon, computed over rollout episodes sampled from the RECON validation split.

Future state prediction and synthesis. Across rollout horizons RAEv2-NWM produces noticeably more accurate and temporally stable predictions than DIAMOND, NWM, and the RAE baseline. Quantitatively, RAEv2-NWM achieves an FVD of 105.61 on the RECON validation set, compared to 762.73 for DIAMOND, 200.97 for NWM, and 312.01 for RAE (Tab. 20); the same ordering holds at every horizon from 1 to 16 seconds on both FID and LPIPS (Fig. 13). Qualitatively, the rollouts also exhibit much less flickering between consecutive frames (Fig. 14).

	DIAMOND [1]	NWM [4]	RAE [85]	RAEv2 (ours)
#Params	1B	1B	622M	622M
FVD [67] ↓	762.73	200.97	312.01	**105.61**

Table 20: **Video prediction quality on RECON [63].** FVD computed over autoregressive rollouts up to 16s. Reference values for DIAMOND and NWM are from [4].

Importance of generalized representation autoencoders. A large fraction of these gains comes from using the generalized RAE formulation (§2.1), which aggregates the encoder's last K layers rather than relying on the final layer alone. The earlier layers retain low-level texture and geometry that are critical for temporally consistent navigation rollouts. As a result, the generalized formulation converges substantially faster during training (Fig. 15; reaching the RAE baseline's final error within roughly 10K iterations), and produces better future-state prediction and video quality across rollout horizons, translating into the substantially lower FVD reported in Tab. 9.

Impact on convergence speed. Fig. 15 shows training curves on RECON under the online single-shot protocol with random offset $\in [1, 8]$ frames at 4 FPS, i.e. predictions 0.25–2 seconds into the future. RAEv2-NWM converges within $\sim$30K iterations to noticeably lower FID and LPIPS than the RAE baseline (FID 7.5 vs. 18.0, LPIPS 0.24 vs. 0.29), and matches the RAE baseline's final FID within the first 10K iterations. This mirrors the speedup we observe on ImageNet (§3.2) and indicates that the improved recipe transfers to navigation world models without modification.

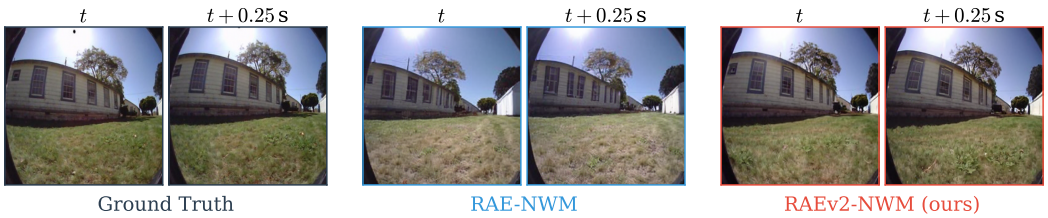


Figure 14: **Qualitative rollouts with and without the generalized representation autoencoder.** Consecutive frames at t and $t{+}0.25$s for ground truth, RAE, and RAEv2-NWM (ours, with the generalized RAE of §2.1). RAE flickers between consecutive frames, while RAEv2-NWM retains low-level detail and preserves scene structure across time, which translates into substantially better FVD (Tab. 9).

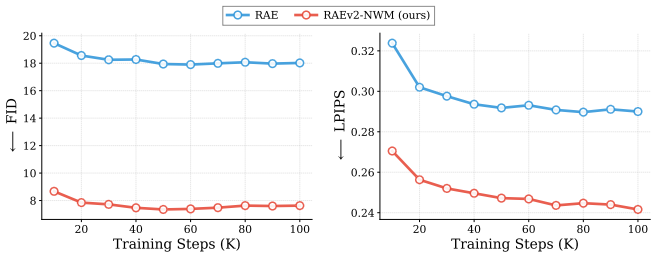


Figure 15: **Convergence speed on RECON validation set.** FID and LPIPS over training steps for RAE and RAEv2-NWM (ours), evaluated under the single-shot prediction protocol (no autoregressive rollout) with random target offset $\in [1, 8]$ frames at 4 FPS, i.e. predictions 0.25–2 seconds into the future. RAEv2-NWM leads to substantially faster convergence speed compare to RAE.

F Qualitative Results

Text-to-image generation. We additionally show text-to-image samples from RAEv2 (0.9B) in Fig. 16–Fig. 18. The model (0.9B) is trained for 100K iterations with batch size 1024 and evaluated on MJHQ test set prompts, generating at 256×256 resolution using self-guidance with the REPA head. Despite the relatively short training schedule and small model size, the samples demonstrate strong prompt adherence across a range of subjects including animals, landscapes, and stylized scenes. The corresponding text prompts are listed in Fig. 19–Fig. 21.

G Discussion and Limitations

We next provide a discussion of some of the limitations of the current work, which might motivate further research in this area. Similar to iREPA [60], we identify the best representation for RAE through empirical search over a discrete set of pretrained encoders. In future work, we would like to directly optimize the representation itself for better generation, with end-to-end learning [41].

H Note on LLM Usage

All figures in the paper are directly generated from our experiment logs and checkpoints using Claude Code (Anthropic, 2025). Additionally, we use LLM help for searching and formulating relevant work in §B. We also use Cursor in some parts to help with paper writing.

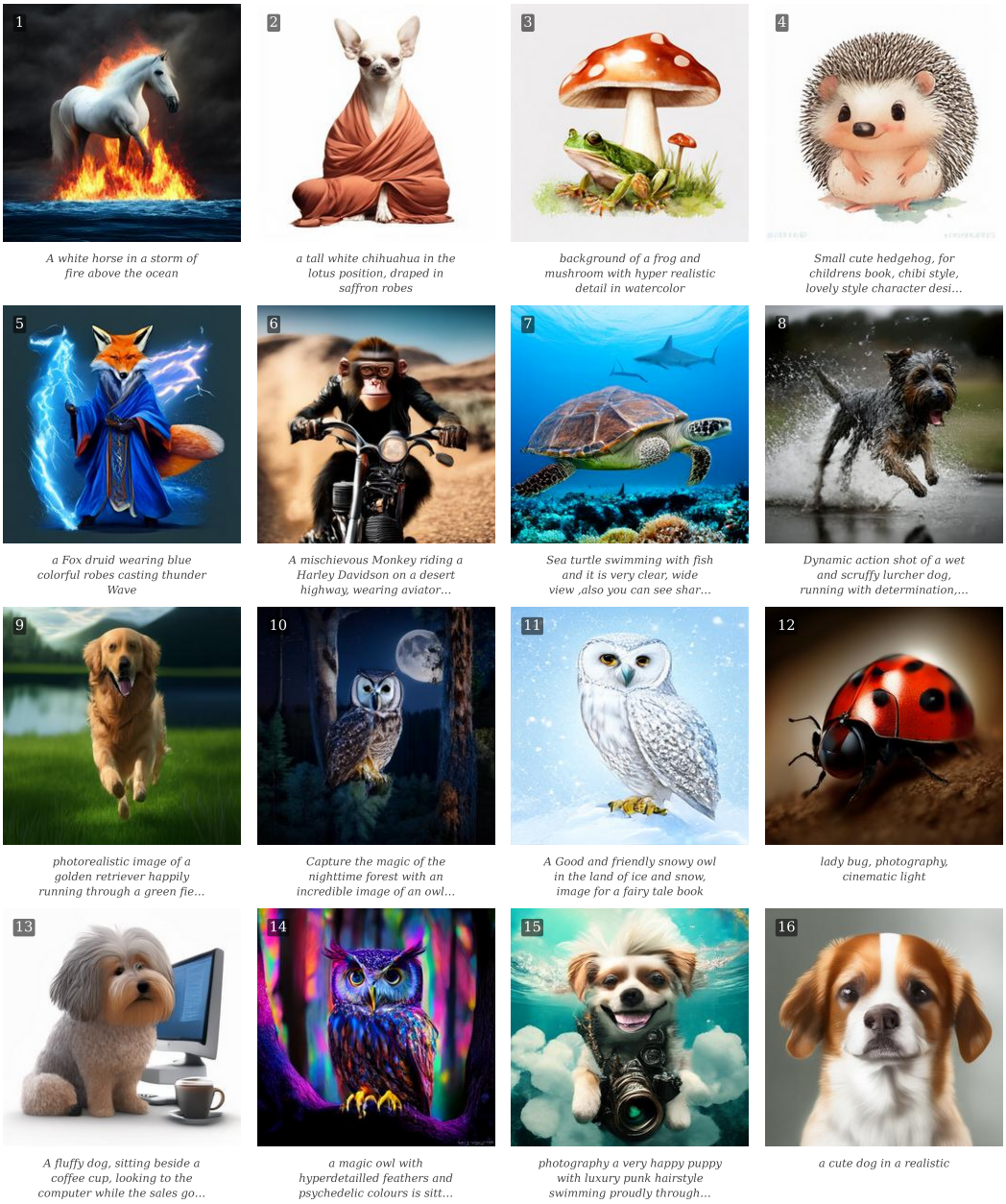


Figure 16: **Text-to-image qualitative examples at 256×256 resolution (1/3).** RAEv2 (0.9B) trained for 100K iterations with batch size 1024, evaluated on MJHQ test set prompts. Generated with self-guidance using REPA head. Corresponding prompts are listed in Fig. 19.

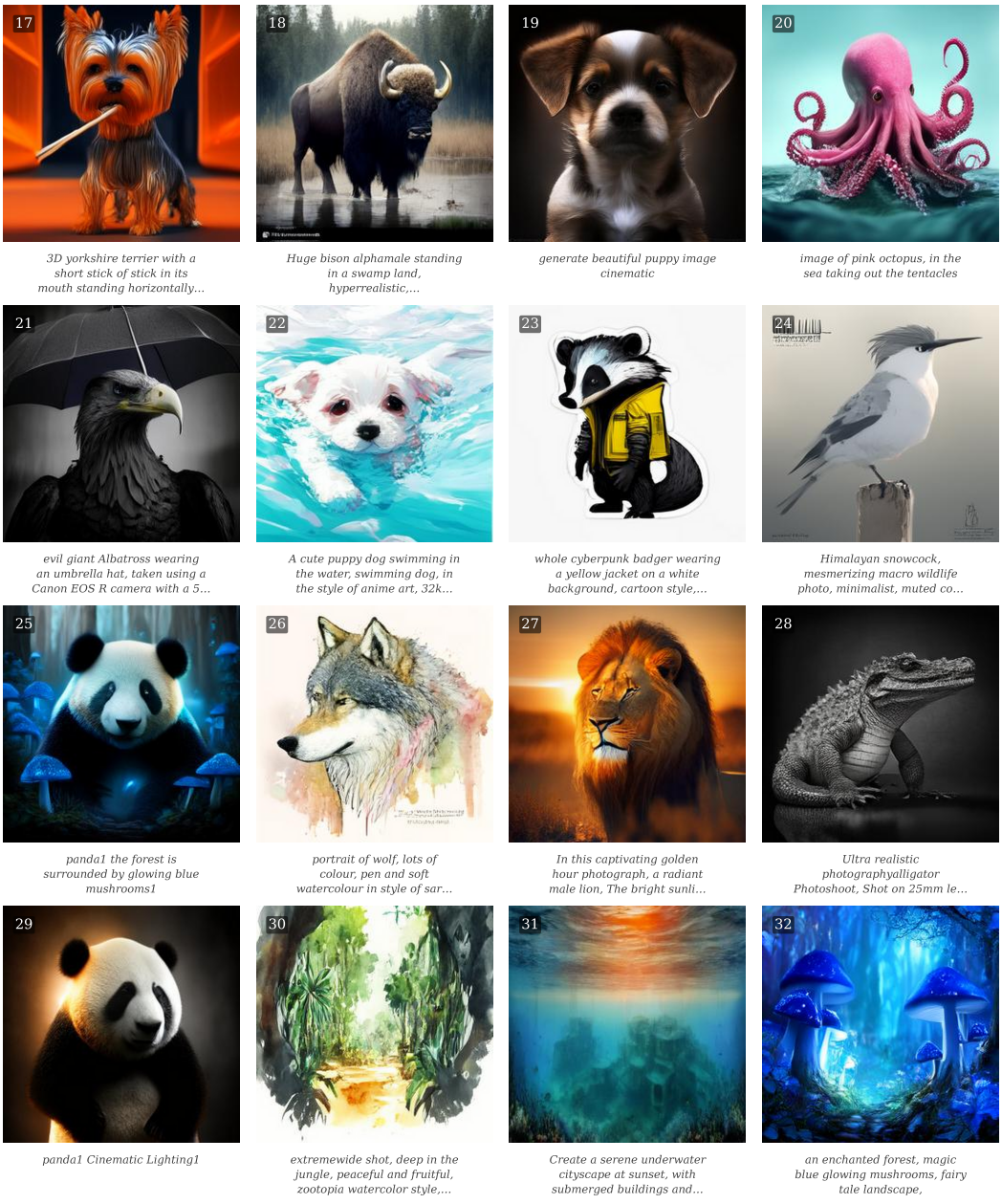


Figure 17: **Text-to-image qualitative examples at 256×256 resolution (2/3).** RAEv2 (0.9B) trained for 100K iterations with batch size 1024, evaluated on MJHQ test set prompts. Generated with self-guidance using REPA head. Corresponding prompts are listed in Fig. 19.

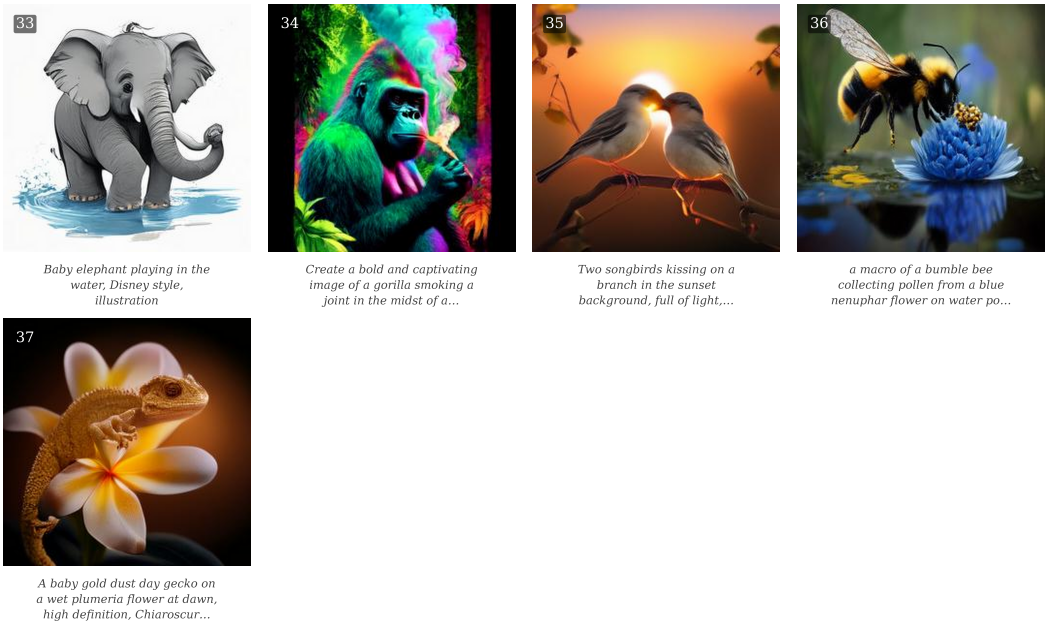


Figure 18: **Text-to-image qualitative examples at 256×256 resolution (3/3).** RAEv2 (0.9B) trained for 100K iterations with batch size 1024, evaluated on MJHQ test set prompts. Generated with self-guidance using REPA head. Corresponding prompts are listed in Fig. 19.

1. A white horse in a storm of fire above the ocean
2. a tall white chihuahua in the lotus position, draped in saffron robes
3. background of a frog and mushroom with hyper realistic detail in watercolor
4. Small cute hedgehog, for childrens book, chibi style, lovely style character design, funny cartoon, lovely animation, simple watercolor, white background, artistic watercolor, very detailed, watercolor, white background
5. a Fox druid wearing blue colorful robes casting thunder Wave
6. A mischievous Monkey riding a Harley Davidson on a desert highway, wearing aviator goggles and a leather jacket, with a trail of dust behind them. intricate 8k, soft lighting, beautifully color graded, Unreal Engine, Cinematic , Color Grading, Photography, Photoshoot, ...
7. Sea turtle swimming with fish and it is very clear, wide view ,also you can see sharks and manta rays in the distance, colored corals are visible on the bottom ,wide angle,Oil painting full body,very detailed,photograph, taken with Hasselblad X1D50c, ...
8. Dynamic action shot of a wet and scruffy lurcher dog, running with determination, splashing water droplets, blurred background to emphasize motion, outdoor setting, overcast day, Nikon D850, 70200mm lens, f2.8, 11000s shutter speed, ISO 800
9. photorealistic image of a golden retriever happily running through a green field with a lake in the background with cinematic lighting, high definition, depth of field superresolution, insanely detailed
10. Capture the magic of the nighttime forest with an incredible image of an owl perched in a tree, illuminated by the full moon. Use the Canon EOS1D X Mark III with the Canon EF 600mm f4L IS III USM lens ...
11. A Good and friendly snowy owl in the land of ice and snow, image for a fairy tale book
12. lady bug, photography, cinematic light
13. A fluffy dog, sitting beside a coffee cup, looking to the computer while the sales go down, ultra realistic, pixar style, 8k, high definition, white background

Figure 19: **Text prompts for T2I qualitative samples (1/3).** Prompts corresponding to the generated images in Fig. 16–Fig. 18.

14. a magic owl with hyperdetailled feathers and psychedelic colours is sitting on a tree in a psychedelic coloured magic forest wide angle photography. huge eyes photorealistic

15. photography a very happy puppy with luxury punk hairstyle swimming proudly through clouds underwater theme dreamy magical steampunk vintage fantasy

16. a cute dog in a realistic

17. 3D yorkshire terrier with a short stick of stick in its mouth standing horizontally, the dog looks proud and raised its chest. It has a futuristic background and the ambient color is orange tones. The dog and the environment have ...

18. Huge bison alphamale standing in a swamp land, hyperrealistic, photorealistic, high details, high quality, shot on Nikon D6, Galen Rowell, Peter Lik, Marc Adamus, David Muench,

19. generate beautiful puppy image cinematic

20. image of pink octopus, in the sea taking out the tentacles

21. evil giant Albatross wearing an umbrella hat, taken using a Canon EOS R camera with a 50mm f1.8 lens, f2.2 aperture, shutter speed 1200s, ISO 100 and natural light, Full Body, Hyper Realistic Photography, Cinematic, Cinema, Hyperdetail, UHD, Color Correction, ...

22. A cute puppy dog swimming in the water, swimming dog, in the style of anime art, 32k uhd, white, impressionism, cute and colorful, fluid, 8k

23. whole cyberpunk badger wearing a yellow jacket on a white background, cartoon style, cyberpunk

24. Himalayan snowcock, mesmerizing macro wildlife photo, minimalist, muted color palette, adventurecore, in the style of handpainted anime, filtered sunlight, conceptual fantasy, intricate details, Noah Bradley, Dan McPharlin.

25. panda1 the forest is surrounded by glowing blue mushrooms1

26. portrait of wolf, lots of colour, pen and soft watercolour in style of sarah taylor art

27. In this captivating golden hour photograph, a radiant male lion, The bright sunlight casts warm, moody hues on the scene, highlighting his long, flowing hair and creating a cinematic atmosphere filled with reflections and contrasts. Captured with a Canon EOS ...

Figure 20: **Text prompts for T2I qualitative samples (2/3).**

28. Ultra realistic photographyalligator Photoshoot, Shot on 25mm lens, Depth of Field, Tilt Blur, Shutter Speed 11000, F22, White Balance, 32k, SuperResolution, Pro Photo RGB, Half rear Lighting, Backlight, Dramatic Lighting, Incandescent, Soft Lighting, Volumetric, ConteJour, Global Illumination, Screen Space Global ...

29. panda1 Cinematic Lighting1

30. extremewide shot, deep in the jungle, peaceful and fruitful, zootopia watercolor style, vibrant color,

31. Create a serene underwater cityscape at sunset, with submerged buildings and structures that have a hint of ancient architecture, surrounded by thriving marine life. The scene should be depicted in a digital painting, inspired by the atmospheric depth and lighting ...

32. an enchanted forest, magic blue glowing mushrooms, fairy tale landscape,

33. Baby elephant playing in the water, Disney style, illustration

34. Create a bold and captivating image of a gorilla smoking a joint in the midst of a colorful, lush forest. The gorilla should be positioned prominently in the foreground, with the psychedelic smoke swirling around its head and creating a ...

35. Two songbirds kissing on a branch in the sunset background, full of light, very high resolution

36. a macro of a bumble bee collecting pollen from a blue nenuphar flower on water pond in a garden inspired by Manet. Intricate details, unreal world, fantasy. Chiaroscuro, Color Grading, Depth of Field, 32k, Super Resolution, Cinematic Lighting, Studio Lighting, ...

37. A baby gold dust day gecko on a wet plumeria flower at dawn, high definition, Chiaroscuro, Color Grading, Depth of Field, 32k, Super Resolution, Cinematic Lighting, Studio Lighting, Global Illumination, CGI, VFX, SFX, insanely detailed and intricate

Figure 21: **Text prompts for T2I qualitative samples (3/3).**