SCAFFOLD THEN INTERNALIZE: REPRESENTATION INJECTION FOR DIFFUSION TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent representation alignment (REPA) methods accelerate diffusion transformer training by aligning projections of the transformer's hidden states with representations from pretrained visual encoders. In this work, we explore a reverse and complementary direction to REPA: rather than projecting diffusion representations into the encoder's space, we inject encoder representations into the diffusion transformer, allowing them to actively participate in the denoising process. To this end, we introduce *REPresentation Injection* (REPI), a training framework based on a scaffold-to-internalization strategy, in which projected encoder representations initially serve as a temporary scaffold and are then progressively internalized by the diffusion transformer. REPI not only outperforms REPA across a wide range of backbones but is also highly complementary to it, and combining the two yields substantial gains over either alone. Notably, with only 160K training steps, REPI + REPA surpasses vanilla SiT trained for 7M steps, a speedup of over $43.5\times$. Code is provided in the supplementary material and will be made publicly available.

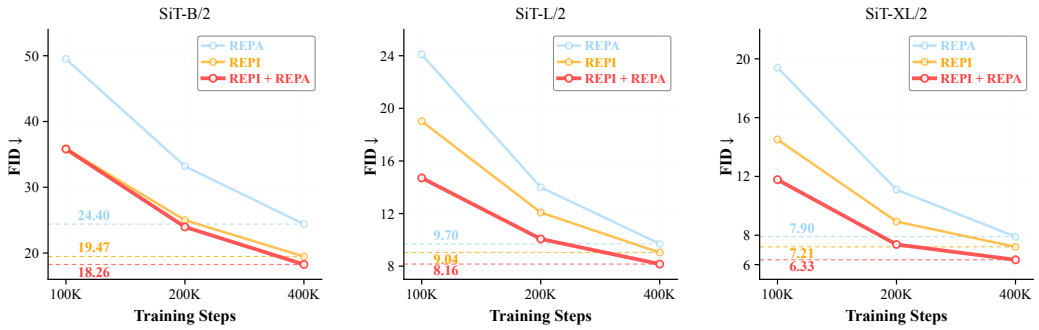


Figure 1: **REPI accelerates diffusion transformer training across model scales.** Compared with REPA, REPI consistently achieves lower FID on SiT-B, SiT-L, and SiT-XL, and combining the two yields further gains. Notably, on SiT-XL, REPI + REPA at 200K training steps already outperforms REPA trained for 400K steps.

1 INTRODUCTION

Diffusion Transformers (DiTs) (Peebles & Xie, 2023; Ma et al., 2024; Esser et al., 2024) exhibit strong scalability for visual generation, but achieving high sample quality still requires extensive training. Representation Alignment (REPA) (Yu et al., 2025) attributes this inefficiency to the difficulty DiTs face in learning semantically structured visual representations, and addresses it by distilling knowledge from a pretrained visual encoder: the diffusion transformer's hidden states are projected and aligned with the encoder's representations via an auxiliary objective (Singh et al., 2026; Wang et al., 2025c). In this paper, we explore a reverse and complementary direction: *rather than projecting diffusion representations into the encoder's space, can encoder representations instead be projected into the diffusion transformer and actively participate in denoising during training?*

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

We begin with a naive design that directly reverses REPA: the encoder output is projected through a lightweight projection layer, and the resulting projection entirely replaces the hidden state at an intermediate layer of the diffusion transformer. However, since this hidden state is the output of a transformer block (see Figure 8 for an illustration), overwriting it discards all prior computation on the noisy input. As a result, subsequent blocks can no longer condition their predictions on the noisy input, causing training to collapse and generated samples to remain pure noise.

This observation motivates moving the injection point inside the transformer block: rather than overwriting the block output, we replace an intermediate quantity within the block, such as the attention output or the key/value (KV) representations. This preserves the flow of information from the noisy input to the block output. For instance, replacing the KV representations still allows the preceding computation to reach subsequent blocks through the query (Q) representations and the residual shortcut. To investigate the potential of this injection strategy, we first consider an idealized oracle setting with access to encoder representations at inference time. Within only 30K training steps, this oracle injection method achieves a lower FID than vanilla SiT (Ma et al., 2024) trained for 7M steps, indicating that encoder representations can provide a useful signal for denoising.

However, this oracle setting relies on encoder representations extracted from clean images at inference time, which are unavailable in standard generation. To close this gap, we introduce *REPresentation Injection* (REPI), which adopts a scaffold-to-internalization training strategy that eliminates the need for encoder representations at inference time. REPI treats the injected encoder representations as a scaffold: during early training, the internal representations at one layer of the diffusion transformer are replaced by projections of the encoder representations; the scaffold is then removed, and the diffusion transformer resumes computing its own representations at that layer. To facilitate this internalization, we introduce an internalization objective that encourages the self-computed representations to match the injected encoder representations. This process allows the model to first learn to exploit the semantically structured representations provided by the scaffold, and then to learn to reproduce compatible representations on its own. After training, the visual encoder is discarded, and the resulting model is architecturally identical to the original diffusion transformer at inference.

Complementarity with REPA. REPI is highly complementary to REPA, as the two methods leverage encoder representations through different mechanisms: REPA encourages the diffusion transformer to learn the structure of encoder representations through an auxiliary alignment loss, while REPI injects and internalizes these representations into the denoising computation, adapting their structure to the denoising task. Empirically, on SiT-XL (Ma et al., 2024), REPI alone achieves an FID of 14.51 at 100K training steps, compared with 19.40 for REPA, and combining both methods further reduces FID to 11.78, substantially outperforming either method alone.

We conduct extensive experiments on class-conditional ImageNet (Deng et al., 2009) generation across multiple model scales to demonstrate the effectiveness of our method. Our approach achieves superior generation quality and training efficiency compared with several state-of-the-art baselines. Notably, on SiT-XL, REPI combined with REPA achieves an FID of 8.22 within only 160K training steps, matching the FID of 8.30 obtained by vanilla SiT after 7M steps, a training speedup of over $43.5\times$. Furthermore, our method at 200K steps already outperforms REPA trained for 400K steps.

Our main contributions are summarized as follows:

- We explore a direction that is reverse and complementary to REPA: rather than projecting diffusion representations into the encoder's space, we inject encoder representations into the diffusion transformer, allowing them to actively participate in the denoising process.
- We introduce REPI, a training framework that internalizes semantically structured representations from pretrained visual encoders into the diffusion transformer via a scaffold-to-internalization strategy.
- We show that REPI alone outperforms REPA, while the two remain highly complementary. Combined with REPA, our model trained for only 160K steps surpasses vanilla SiT trained for 7M steps, a speedup of over $43.5\times$.

2 PRELIMINARIES

Diffusion with flow matching. Given a clean sample $\mathbf{x}_* \sim p_{\text{data}}$ and condition c, flow matching (Lipman et al., 2023; Albergo & Vanden-Eijnden, 2022) constructs $\mathbf{x}_t = \alpha_t \mathbf{x}_* + \sigma_t \epsilon$ on $t \in [0, 1]$, where $\epsilon \sim \mathcal{N}(0, I)$, $(\alpha_0, \sigma_0) = (1, 0)$, and $(\alpha_1, \sigma_1) = (0, 1)$. A denoiser $\mathbf{v}_\theta(\mathbf{x}_t, t, c)$ is trained by

$$\mathcal{L}_{\text{velocity}}(\theta) = \mathbb{E}_{\mathbf{x}_*, \epsilon, t}\Big[\big\| \mathbf{v}_\theta(\mathbf{x}_t, t, c) - \mathbf{v}_t \big\|^2 \Big], \qquad \mathbf{v}_t = \dot{\alpha}_t \mathbf{x}_* + \dot{\sigma}_t \epsilon. \tag{1}$$

Representation alignment. REPA (Yu et al., 2025) aligns an intermediate denoiser state $\mathbf{h}_t$ with clean-image features $\mathbf{y}_* = f(\mathbf{x}_*)$ from a pretrained visual encoder f. A projection head h_ϕ maps $\mathbf{h}_t$ into the encoder space, yielding

$$\mathcal{L}_{\text{REPA}}(\theta, \phi) = -\mathbb{E}_{\mathbf{x}_*, \epsilon, t}\Big[\frac{1}{N} \sum_{n=1}^{N} \text{sim}\big(h_\phi(\mathbf{h}_t^{[n]}),\, \mathbf{y}_*^{[n]}\big) \Big], \tag{2}$$

where n indexes patches and $\text{sim}(\cdot, \cdot)$ is a predefined similarity function.

3 CAN ENCODER REPRESENTATIONS PARTICIPATE IN DENOISING?

REPA transfers knowledge from a pretrained visual encoder to a diffusion transformer through a regularization term that aligns the transformer's projected hidden states with target representations from the encoder. In this section, we investigate a complementary question: rather than aligning with encoder representations, can they instead be mapped directly into the diffusion transformer and participate actively in denoising? We first examine this question under a diagnostic oracle setting, in which encoder representations are assumed to be available at inference time. Section 4 then introduces our method, which eliminates the need for encoder representations at inference time.

A naive design is to simply reverse the mapping direction of REPA: instead of projecting diffusion hidden states into the encoder's space, we use a lightweight projection layer to map encoder representations into the diffusion hidden-state space and substitute them for an intermediate hidden state. This design fails catastrophically, yielding an FID of 212.5 (Table 1). We attribute this failure to the fact that the hidden state is the output of a transformer block (Figure 8), so replacing it entirely severs all subsequent blocks from the noisy input. This failure suggests that external representations must be injected without overwriting the input-dependent stream. We therefore instead inject representations inside the attention module, testing five alternatives that all preserve the flow of information from the noisy input to the block output (Table 1). Details of these variants are provided in Appendix B.

Table 1: Oracle results (SiT-XL/2, 30K steps).

Injection target	FID↓
Hidden state	212.5
Attention output	5.0
Q/K/V	5.2
K/V	7.1
K only	12.4
V only	4.6

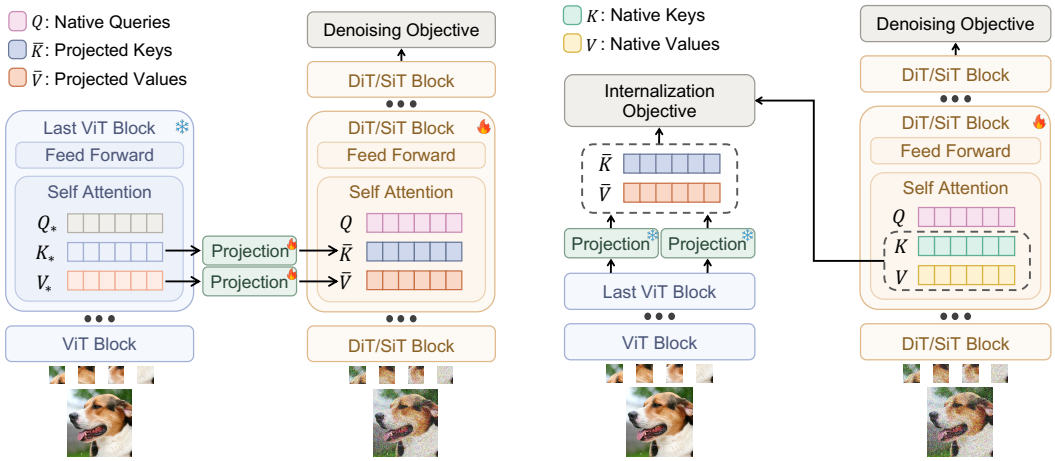
As shown in Table 1, all five variants achieve strong oracle FID after only 30K training steps, with V-only injection performing best. A plausible explanation is that V-only injection preserves the diffusion transformer's native attention routing through Q and K, while injecting semantic content from the encoder via V. Note that the optimal injection location differs for our full model, as discussed in Section 4.2.

Although this oracle setting is impractical, as it assumes access to encoder representations at inference time, these results indicate that projected encoder representations can substantially benefit the denoising process. We next describe how our method eliminates this dependence on encoder representations.

4 REPI: REPRESENTATION INJECTION AND INTERNALIZATION

4.1 OVERVIEW

The oracle setting in Section 3 requires representations extracted from clean images during sampling, which are unavailable in standard generation. To eliminate this dependency, we propose *REPresentation Injection* (REPI), which treats projected encoder representations as a scaffold and internalizes their benefit into the diffusion transformer. As illustrated in Figure 2, REPI projects a



(a) Scaffold (b) Internalization

Figure 2: **Overview of REPI.** (a) *Scaffold*: projected keys and values from a frozen visual encoder temporarily replace the diffusion transformer's native keys and values, while its native queries are preserved. (b) *Internalization*: the model resumes using its own keys and values, guided by an internalization objective that aligns them with the projected encoder representations. At inference time, the encoder and projection layers are discarded.

representation extracted from a clean image and substitutes it for the native representation within the diffusion transformer (Section 4.2). After a small number of training steps, we remove the scaffold and restore the native computation, while an internalization objective encourages the native representation to reproduce the representation previously supplied by the scaffold (Section 4.3). At inference time, the visual encoder and projection layers are discarded, and sampling relies solely on the original diffusion backbone.

4.2 Representation Injection

REPI initially substitutes projections of clean-image encoder representations for the native representations at a selected diffusion layer, serving as a temporary scaffold that is later removed as training progresses. Specifically, we inject K/V representations, as we empirically find this choice achieves the best results (Table 5). This finding is inconsistent with the oracle results in Section 3, where V-only injection performs best. One possible explanation is that the oracle setting only needs to consider how effectively an external representation can be consumed, whereas REPI additionally requires that the injected representation be subsequently internalized and removed.

Formally, let f denote a pretrained visual encoder and $\mathbf{x}_*$ a clean image. During the forward pass of $f(\mathbf{x}_*)$, we extract the key and value representations (K_*, V_*) computed within its final self-attention layer. At a selected self-attention layer of the diffusion transformer, we introduce two trainable projection layers, h_{ϕ_K} and h_{ϕ_V}, which map K_* and V_* into the corresponding key and value spaces of that layer:

$$\bar{K} = h_{\phi_K}(K_*), \qquad \bar{V} = h_{\phi_V}(V_*). \tag{3}$$

We then replace the native key and value representations with $\bar{K}$ and $\bar{V}$, respectively. The resulting attention output is

$$Z_{\text{inj}} = \text{softmax}\left(\frac{Q\bar{K}^\top}{\sqrt{d}}\right)\bar{V}, \tag{4}$$

where Q denotes the native query representation and d is the query/key dimension. For clarity, we omit layer and attention-head indices. REPI changes only the source of the keys and values, leaving the other operations of the attention block unchanged. Notably, in addition to softmax attention, this substitution generalizes to the gated linear attention used in DiG (Zhu et al., 2024), as demonstrated in Figure 5.

This design allows information from the noisy input to continue propagating through the native query Q and the residual shortcut, preserving the input-dependent information accumulated by pre-

ceding blocks. Meanwhile, the projections $\bar{K}$ and $\bar{V}$ provide a semantically structured key-value memory derived from the clean-image representation, in which the keys determine which external information is retrieved, while the values determine its content.

In practice, we implement h_{ϕ_K} and h_{ϕ_V} as linear layers, and select the same diffusion transformer layer and encoder layer as REPA. During training, the diffusion transformer and the projection layers are jointly optimized using the diffusion objective (Eq. 1).

4.3 Internalization and Scaffold Removal

The injected encoder representations provide semantically organized content for denoising, but cannot be retained during standard sampling, as their construction requires a clean image. We therefore treat these injected representations not as permanent conditioning, but as a temporary representation scaffold, which we later remove while internalizing its benefit into the diffusion transformer's native representations.

A simple strategy is to abruptly remove the scaffold and restore the diffusion transformer's native representations after a small number of training steps. Surprisingly, even this naive switch improves FID from 39.4 for vanilla SiT to 22.45 at 100K steps. We hypothesize that the scaffold provides a favorable initialization for the other transformer layers: while the scaffold is active, these layers learn to exploit semantically structured encoder representations for denoising. After the scaffold is removed, this learned capability may facilitate subsequent training as the model adapts to its own native representations.

To internalize the encoder representations more thoroughly, we introduce an internalization loss that explicitly encourages the native K/V representations to reproduce their encoder-derived counterparts. Specifically, the native representations (K, V) are trained to match the projections $(\bar{K}, \bar{V})$:

$$\mathcal{L}_{\text{int}} = \frac{1}{|K|} \left\| K - \text{sg}[\bar{K}] \right\|_F^2 + \frac{1}{|V|} \left\| V - \text{sg}[\bar{V}] \right\|_F^2, \tag{5}$$

where $\text{sg}[\cdot]$ denotes the stop-gradient operator and $|\cdot|$ denotes the number of elements in the corresponding representation.

This internalization loss also facilitates a smooth transition, as it encourages the native K/V representations to remain consistent with their pre-transition states. We additionally experimented with a linearly scheduled transition but observed no further gain, since the internalization loss alone already renders the switch sufficiently smooth.

In practice, this internalization term is added to the diffusion loss, yielding:

$$\mathcal{L} = \mathcal{L}_{\text{velocity}} + \lambda \mathcal{L}_{\text{int}}, \tag{6}$$

where λ controls the tradeoff between denoising and internalization.

4.4 Combining REPI with REPA

We empirically find that REPI and REPA are highly complementary (Figure 1), as the two methods leverage encoder representations through fundamentally different mechanisms. REPA aligns projections of intermediate diffusion representations with corresponding encoder representations, providing diffusion-to-encoder alignment supervision. In contrast, REPI injects and internalizes projections of encoder representations into the denoising computation, forming an encoder-to-diffusion injection mechanism.

When combining the two methods, REPA supervision is applied throughout the entire training process, from scaffold to internalization. We place the REPI injection layer before the REPA alignment layer, an ordering that offers two benefits. First, it allows the two mechanisms to operate cooperatively rather than interfere with each other: injecting after the alignment layer would instead alter the subsequent computation, weakening the influence of the REPA supervision applied earlier. Second, once the scaffold is removed, REPA continues to provide semantic supervision at the alignment layer, complementing the internalization objective. Table 8 reports results for different injection-layer choices, showing that reversing the order still improves performance, but the gains are substantially smaller.

Table 2: **FID comparison on SiTs.** All results are reported on ImageNet 256×256 without CFG.

Method	Iter.	FID↓
SiT-B/2	400K	33.00
+ REPA	400K	24.40
+ REPI (ours)	**400K**	**19.47**
SiT-L/2	400K	18.81
+ REPA	400K	9.70
+ REPI (ours)	**400K**	**9.04**
SiT-XL/2	7M	8.30
+ REPA	100K	19.40
+ REPA	400K	7.90
+ iREPA	100K	16.96
+ iREPA	400K	7.52
+ sREPA	100K	15.40
+ sREPA	400K	7.17
+ Stable Velocity	100K	17.12
+ Stable Velocity	400K	7.58
+ REPI (ours)	100K	14.51
+ REPI (ours)	400K	7.21
+ REPA + REPI (ours)	**100K**	**11.78**
+ REPA + REPI (ours)	160K	8.22
+ REPA + REPI (ours)	**400K**	**6.33**

Table 3: **Quantitative comparison on ImageNet 256 × 256 with CFG**. Asterisks (*) denote CFG scheduling with the guidance interval (Kynkäänniemi et al., 2024).

Model	Epochs	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
Latent Diffusion Transformer						
MaskDiT	1600	2.28	5.67	276.6	0.80	0.61
Faster-DiT	400	2.03	4.63	264.0	0.81	0.60
DiT-XL/2	1400	2.27	4.60	278.2	0.83	0.57
SiT-XL/2	1400	2.06	4.50	270.3	0.82	0.59
Representation Alignment Methods (SiT-XL/2)						
+ REPA	80	2.39	4.64	246.8	0.82	0.57
+ REPA	200	1.96	4.49	264.0	0.82	0.60
+ sREPA	80	2.25	4.65	257.5	0.83	0.58
+ sREPA	200	1.91	4.50	271.8	**0.83**	0.60
+ REPI (ours)	80	2.22	4.58	236.4	0.81	0.59
+ REPI (ours)	200	1.90	4.49	255.1	0.81	**0.61**
+ REPA + REPI (ours)	80	2.08	4.55	245.7	0.81	0.59
+ REPA + REPI (ours)	200	**1.85**	**4.46**	**278.4**	0.82	0.60
+ REPA*	80	1.98	4.60	263.0	0.80	**0.61**
+ iREPA*	80	1.93	4.59	268.8	0.80	0.60
+ Stable Velocity*	80	1.80	**4.52**	272.4	**0.81**	0.60
+ REPI (ours)*	80	1.79	4.61	269.2	0.81	0.60
+ REPA + REPI (ours)*	80	**1.73**	4.62	**278.4**	**0.81**	0.60

5 Experiments

5.1 Experimental Setup

Implementation details. We strictly follow the training protocols of REPA (Yu et al., 2025). To ensure a fair comparison, we fix the training batch size to 256 and adopt the same learning rate and EMA configurations as REPA. For SiT (Ma et al., 2024) models, we employ the SDE Euler–Maruyama sampler with 250 function evaluations (NFE). When combined with REPA, we always use its default alignment depth and loss weight. Additional implementation details and hyperparameter settings are provided in Appendix A.

Models and datasets. We evaluate our method on DiT (Peebles & Xie, 2023), SiT (Ma et al., 2024), and DiG (Zhu et al., 2024) across multiple model scales. Following REPA, our main experiments are conducted on class-conditional ImageNet (Deng et al., 2009) at 256×256 resolution. We further evaluate on ImageNet at 512×512 to assess higher-resolution generation, and on MS-COCO (Lin et al., 2014) with MM-DiT (Esser et al., 2024) for text-to-image generation. Unless otherwise specified, we adopt DINOv2-B/14 (Oquab et al., 2024) as the pretrained visual encoder, and all models operate in the latent space of a pretrained VAE (Kingma & Welling, 2013).

Evaluation. We compare our method against four representation alignment methods: REPA (Yu et al., 2025), iREPA (Singh et al., 2026), sREPA (Xu et al., 2026), and Stable Velocity (Yang et al., 2026). We report FID (Heusel et al., 2017), sFID (Nash et al., 2021), IS (Salimans et al., 2016), and Precision/Recall (Kynkäänniemi et al., 2019), all computed on 50K generated samples following the ADM protocol (Dhariwal & Nichol, 2021). Unless otherwise noted, all results are obtained using the EMA model without classifier-free guidance (CFG) (Ho & Salimans, 2022). For our method, reported training steps account for the entire training process, from scaffold to internalization.

5.2 Main Results

Quantitative results. Table 2, together with Figures 1 and 5, presents the quantitative comparison without classifier-free guidance. Across a diverse set of backbones, REPI consistently outperforms REPA, and combining the two methods yields further gains. Notably, on SiT-XL, REPA + REPI achieves an FID of 8.22 after only 160K training steps, matching the FID of 8.30 obtained by vanilla SiT after 7M steps, a training speedup of over $43.5\times$. Moreover, REPA + REPI at 200K steps already surpasses REPA at 400K steps. We further evaluate under classifier-free guidance, both with and without the guidance interval (Kynkäänniemi et al., 2024). As shown in Table 3, REPI alone achieves comparable performance to the baselines, while REPA + REPI achieves the best FID and IS among all methods in both settings.

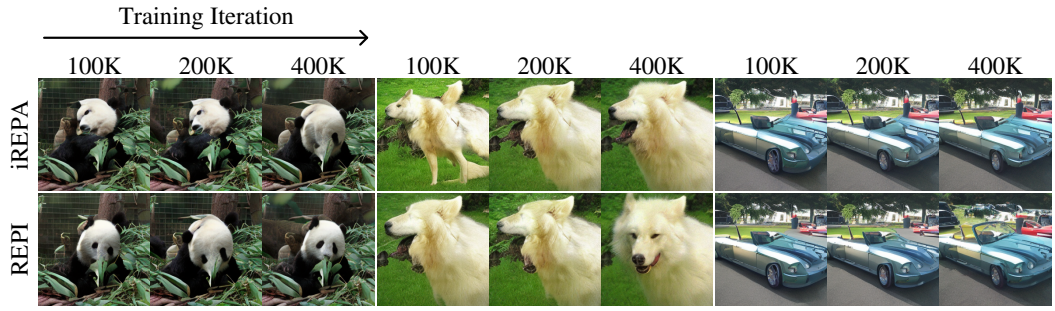


Figure 3: **Qualitative comparison across training iterations.** Generated samples from SiT-XL/2 trained with iREPA (top) and REPI (bottom) over the first 400K iterations. Both models use the same initial noise and sampler, without classifier-free guidance.

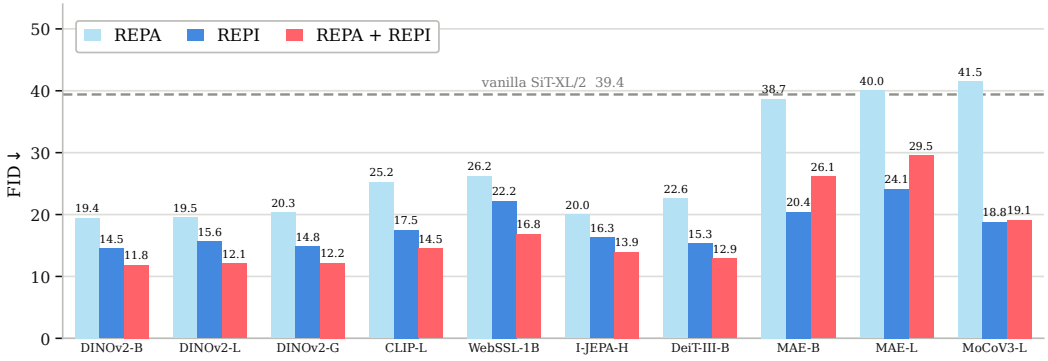


Figure 4: **Generality of REPI across visual encoders.** FID results of REPA, REPI, and REPA + REPI at 100K training steps on SiT-XL/2. REPI consistently outperforms REPA across all ten encoders. Notably, on encoders such as MAE-L and MoCoV3-L, where REPA underperforms vanilla SiT, REPI still achieves substantial gains.

Qualitative results. Figure 3 compares samples from our method and iREPA, generated from the same initial noise without classifier-free guidance. Our method produces images with more coherent semantic structure and finer details. Additional qualitative results are provided in Appendix E.

Text-to-image generation. Following REPA, we train MM-DiT on MS-COCO for 150K steps and use ODE sampling with NFE $= 50$, adopting the same evaluation protocol as REPA. As shown in Table 4, REPI consistently outperforms REPA both with and without CFG, reducing FID from 10.40 to 9.47 (w/o CFG) and 4.73 to 4.61 (w/ CFG). Combining REPI with REPA further reduces FID to 9.08 and 4.56, respectively.

Table 4: FID comparison on text-to-image generation.

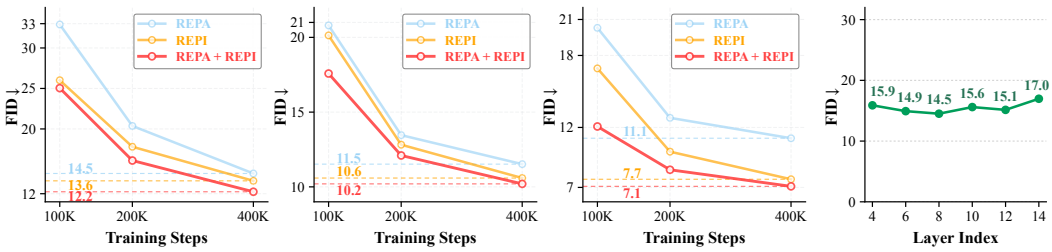
Method	w/o CFG	w/ CFG
REPA	10.40	4.73
REPI	9.47	4.61
REPA + REPI	**9.08**	**4.56**

5.3 Generality of REPI

In this section, we demonstrate that REPI improves diffusion transformer training across a broad range of settings, including visual encoders, diffusion backbones, and injection depths.

Visual encoders. We first study the effect of different visual encoder types and sizes. As shown in Figure 4, REPI consistently improves upon the vanilla model and outperforms REPA across all encoders. Notably, on several encoders (e.g., MAE and MoCoV3), REPA yields FID scores even worse than vanilla SiT, consistent with observations in iREPA (Singh et al., 2026). In contrast, REPI still achieves substantial gains on these encoders. For the remaining encoders, combining REPI with REPA further reduces FID substantially.

Diffusion backbones. We next evaluate REPI across a diverse set of diffusion transformer backbones. Figure 1 reports results across model scales (SiT-B, SiT-L, and SiT-XL), while Figure 5



(a) DiT-L/2 (b) DiG-L/2 (c) SiT-XL/2 (512) (d) Injection depth

Figure 5: **Robustness of REPI across diffusion backbones and injection layers.** (a–c) FID comparison on DiT-L/2, DiG-L/2, and SiT-XL/2 (512×512). REPI consistently outperforms REPA, and combining the two yields further gains. (d) REPI is robust to the choice of injection depth, with layer 8 achieving the best performance on SiT-XL/2 at 100K steps.

examines generality across architectures (DiT-L and DiG-L, spanning standard softmax attention and gated linear attention) and at a higher resolution (SiT-XL at 512×512). REPI generalizes consistently across model scales, architectures, and resolutions, outperforming REPA in every setting, and combining the two methods yields further substantial gains.

Injection depth. Finally, we demonstrate the robustness of REPI to the choice of injection layer. As shown in Figure 5(d), FID ranges only from 14.51 to 16.97 across injection layers 4 through 14, substantially below the 39.41 achieved by vanilla SiT-XL. Injecting at layer 8 achieves the best performance, consistent with the layer choice adopted in REPA.

5.4 Ablation Studies

Scaffold and internalization. Figure 6 ablates the contributions of the scaffold and internalization objective by progressively adding each component. The scaffold substantially improves performance over the base model, regardless of whether REPA is applied. Notably, using the scaffold alone, without REPA or the internalization objective (i.e., abrupt scaffold removal), reduces FID on SiT-XL from 17.2 to 9.49 at 400K steps. This suggests that the scaffold provides a favorable initialization for the diffusion transformer, with lasting benefits even after its removal. Applying the internalization objective further reduces FID, both with and without REPA.

Injection target. We next study where encoder representations should be injected within the attention module, comparing five variants summarized in Table 5. All variants improve upon vanilla SiT, with K/V injection achieving the best FID and IS. Interestingly, this result contrasts with the oracle study in Section 3, where V-only injection performs best. We hypothesize that jointly injecting keys and values is better suited to scaffold removal and internalization: the projected keys determine how the native queries retrieve encoder-derived information, while the projected values supply the corresponding content. Notably, injecting into the attention output or into all of Q/K/V yields relatively worse results, suggesting that preserving native queries computed from the noisy input is important for internalization.

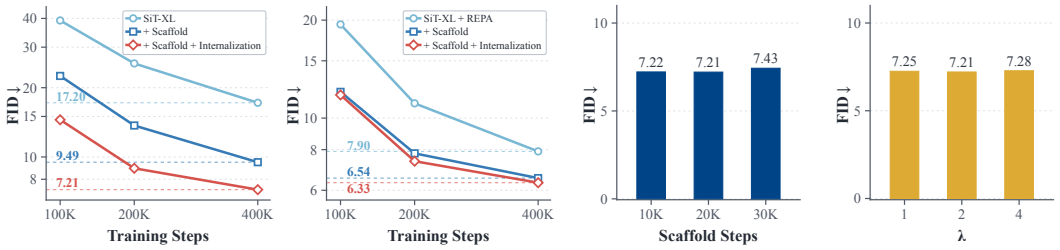
Table 5: Injection-target ablation (SiT-XL/2, 100K steps).

Target	FID↓	IS↑
Attention output	16.34	78.62
Q/K/V	17.40	71.60
K only	15.81	77.57
V only	15.53	79.31
K/V	**14.51**	**82.71**

Scaffold duration. We also investigate the effect of scaffold duration, varying it from 10K to 30K training steps. As shown in Figure 7(a), performance remains stable across durations, with FID varying by only 0.22, demonstrating that our method is robust to this choice. Notably, a short scaffold duration is sufficient to achieve strong performance in practice, as our oracle experiments show that the projection layers can be learned effectively within a small number of training steps.

Internalization loss weight. Figure 7(b) shows that REPI is also robust to the internalization loss weight. Varying λ from 1.0 to 4.0 yields similar FID scores, ranging between 7.21 and 7.28. We use $\lambda = 2$ by default, which achieves the best result among the tested values.

We provide additional ablation studies for the combination of REPI and REPA in Appendix C.



(a) Without REPA (b) With REPA

Figure 6: **Ablation of the scaffold and internalization objective.** Starting from SiT-XL/2, (a) without REPA and (b) with REPA, we progressively add the scaffold and the internalization objective. Each component consistently improves FID.

(a) Scaffold duration (b) Loss weight

Figure 7: **Robustness to scaffold duration and internalization loss weight.** REPI achieves stable FID across (a) scaffold durations and (b) internalization loss weights, evaluated on SiT-XL/2 at 400K training steps.

6 Related Work

Diffusion transformer. Diffusion Transformers (DiTs) (Peebles & Xie, 2023) adopt Vision Transformers (Dosovitskiy et al., 2021) as backbones for latent diffusion (Rombach et al., 2022), while Scalable Interpolant Transformers (SiTs) (Ma et al., 2024) connect diffusion and flow matching (Lipman et al., 2023) through stochastic interpolants. Subsequent works improve efficiency through linear attention (Xie et al., 2025; Zhu et al., 2024; Wang et al., 2025b; Katharopoulos et al., 2020; Wang et al., 2020; Choromanski et al., 2021) and masked image modeling (Zheng et al., 2024; Gao et al., 2023; He et al., 2022), and training strategies that accelerate convergence without architectural modifications (Yao et al., 2024).

Representation alignment for generation. REPA (Yu et al., 2025) accelerates diffusion transformer training by aligning intermediate features with clean-image representations from pretrained visual encoders. Subsequent work refines the alignment target and training strategy by emphasizing spatial structure (Singh et al., 2026), aligning relational geometry (Xu et al., 2026), terminating alignment early in training (Wang et al., 2025c), or adopting more flexible and adaptive alignment schemes (Pang et al., 2026; Wang et al., 2025a; Mo & Yun, 2026). Other work derives alignment targets from the diffusion process itself (Jiang et al., 2025; Peng et al., 2026; Chefer et al., 2026) or from VAE features (Wang et al., 2026; Min et al., 2026), while others optimize the generative latent space directly or jointly model image and semantic representations (Yao et al., 2025; Leng et al., 2025; Kouzelis et al., 2025; Wu et al., 2025a). The paradigm has further been extended to U-Nets (Tian et al., 2025), pixel-space transformers (Shin et al., 2026), and video generation (Zhang et al., 2025; Wu et al., 2025b; Lian et al., 2026). Unlike these methods, which treat external representations as an alignment target, we assign them a complementary role: projected encoder representations are temporarily injected into the denoising computation as a scaffold, while the model's native representations gradually internalize this guidance.

7 Conclusion

In this paper, we introduced REPI, a training framework that leverages pretrained vision encoders in a direction complementary to REPA. Specifically, we investigated whether encoder representations can be injected into a diffusion transformer to benefit the denoising task. REPI follows a scaffold-to-internalization strategy: projected encoder representations initially serve as a temporary scaffold, while an internalization objective encourages the diffusion transformer to reproduce compatible representations on its own. Extensive experiments show that REPI improves both the generation quality and training efficiency of diffusion transformers. We hope this work will motivate further exploration of how external representations can be leveraged in generative training. Promising future directions include extending REPI beyond image generation to video generation and other generative domains.

AI USE STATEMENT

In this work, we used generative AI tools for language editing and polishing to improve the clarity and readability of the manuscript, as well as for code optimization and debugging. We did not use generative AI tools to generate research ideas, formulate hypotheses, design the methodology, conduct experiments, or interpret the results. All AI-assisted text was carefully reviewed and revised by the authors, and all AI-assisted code was verified and tested for correctness. We take full responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the model architectures, training configurations, and hyperparameter settings in Section 5.1 and Appendix A. Our source code will be made publicly available to facilitate reproducibility.

REFERENCES

Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Hila Chefer, Patrick Esser, Dominik Lorenz, Dustin Podell, Vikash Raja, Vinh Tong, Antonio Torralba, and Robin Rombach. Self-supervised flow matching for scalable multi-modal synthesis. *arXiv preprint arXiv:2603.06507*, 2026.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *International Conference on Computer Vision*, 2021.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition*, 2009.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024.

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. In *International Conference on Computer Vision*, 2025.

Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Masked diffusion transformer is a strong image synthesizer. In *International Conference on Computer Vision*, 2023.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017.

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

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.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, 2020.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Theodoros Kouzelis, Efstathios Karypidis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Boosting generative image modeling via joint image-feature synthesis. *arXiv preprint arXiv:2504.16064*, 2025.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, 2019.

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. In *Advances in Neural Information Processing Systems*, 2024.

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.

Jiesong Lian, Zixiang Zhou, Ruizhe Zhong, Yuan Zhou, Qinglin Lu, Rui Wang, Long Hu, Yixue Hao, and Baoru Huang. Sara: Semantically adaptive relational alignment for video diffusion models. *arXiv preprint arXiv:2605.07800*, 2026.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

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*, 2024.

Ruibin Min, Yexin Liu, Aimin Pan, Changsheng Lu, Jiafei Wu, Kelu Yao, Xiaogang Xu, and Harry Yang. Ahpa: Adaptive hierarchical prior alignment for diffusion transformers. *arXiv preprint arXiv:2605.03317*, 2026.

Shentong Mo and Sukmin Yun. Improving visual representation alignment generation with grpo. *arXiv preprint arXiv:2606.00583*, 2026.

Charlie Nash, Jacob Menick, Sander Dieleman, and Peter Battaglia. Generating images with sparse representations. In *International Conference on Machine Learning*, 2021.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, 2021.

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*, pp. 1–31, 2024.

Lianyu Pang, Tianlin Pan, Cheng Da, Changqian Yu, Huan Yang, Kun Gai, Song Guo, and Wenhan Luo. Maskalign: Token-subset representation alignment for efficient diffusion training. *arXiv preprint arXiv:2606.08788*, 2026.

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

Changhao Peng, Yuqi Ye, Shuangjun Du, Wenxu Gao, and Wei Gao. Dual-path condition alignment for diffusion transformers. In *International Conference on Learning Representations*, 2026.

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*, pp. 8748–8763. PMLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, 2016.

Jaeyo Shin, Jiwook Kim, and Hyunjung Shim. Representation alignment for just image transformers is not easier than you think. *arXiv preprint arXiv:2603.14366*, 2026.

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? In *International Conference on Learning Representations*, 2026.

Yuchuan Tian, Hanting Chen, Mengyu Zheng, Yuchen Liang, Chao Xu, and Yunhe Wang. U-repa: Aligning diffusion u-nets to vits. In *Advances in Neural Information Processing Systems*, 2025.

Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In *European conference on computer vision*, pp. 516–533. Springer, 2022.

Chenyu Wang, Cai Zhou, Sharut Gupta, Zongyu Lin, Stefanie Jegelka, Stephen Bates, and Tommi Jaakkola. Learning diffusion models with flexible representation guidance. *The Thirty-Ninth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025a.

Jiahao Wang, Ning Kang, Lewei Yao, Mengzhao Chen, Chengyue Wu, Songyang Zhang, Shuchen Xue, Yong Liu, Taiqiang Wu, Xihui Liu, Kaipeng Zhang, Shifeng Zhang, Wenqi Shao, Zhenguo Li, and Ping Luo. Lit: Delving into a simplified linear diffusion transformer for image generation. *arXiv preprint arXiv:2501.12976*, 2025b.

Mengmeng Wang, Dengyang Jiang, Liuzhuozheng Li, Yucheng Lin, Guojiang Shen, Xiangjie Kong, Yong Liu, Guang Dai, and Jingdong Wang. Sra 2: Variational autoencoder self-representation alignment for efficient diffusion training. *arXiv preprint arXiv:2601.17830*, 2026.

Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Ziqiao Wang, Wangbo Zhao, Yuhao Zhou, Zekai Li, Zhiyuan Liang, Mingjia Shi, Xuanlei Zhao, Pengfei Zhou, Kaipeng Zhang, Zhangyang Wang, Kai Wang, and Yang You. Repa works until it doesn't: Early-stopped, holistic alignment supercharges diffusion training. *arXiv preprint arXiv:2505.16792*, 2025c.

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

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*, 2025b.

Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, and Song Han. Sana: Efficient high-resolution text-to-image synthesis with linear diffusion transformers. In *International Conference on Learning Representations*, 2025.

Shaodong Xu, Zhendong Wang, Litong Gong, Zexian Li, Wengang Zhou, Tiezheng Ge, and Houqiang Li. Beyond point-wise matching: Structural representation alignment for accelerating diffusion transformers. *arXiv preprint arXiv:2605.16949*, 2026.

Donglin Yang, Yongxing Zhang, Xin Yu, Liang Hou, Xin Tao, Pengfei Wan, Xiaojuan Qi, and Renjie Liao. Stable velocity: A variance perspective on flow matching. In *International Conference on Machine Learning*, 2026.

Jingfeng Yao, Cheng Wang, Wenyu Liu, and Xinggang Wang. Fasterdit: Towards faster diffusion transformers training without architecture modification. In *Advances in Neural Information Processing Systems*, 2024.

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

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025.

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.

Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *Transactions on Machine Learning Research*, 2024.

Lianghui Zhu, Zilong Huang, Bencheng Liao, Jun Hao Liew, Hanshu Yan, Jiashi Feng, and Xinggang Wang. Dig: Scalable and efficient diffusion models with gated linear attention. *arXiv preprint arXiv:2405.18428*, 2024.

A Additional Implementation Details and Hyperparameters

More implementation details. We strictly follow the training protocol of REPA (Yu et al., 2025). Latent representations are pre-computed using the `stabilityai/sd-vae-ft-ema` VAE encoder. The K/V projection heads h_{ϕ_K} and h_{ϕ_V} are each implemented as a single linear layer. All parameters are optimized with AdamW (Loshchilov & Hutter, 2019), using a constant learning rate of 10^{-4}, momentum parameters $(\beta_1, \beta_2) = (0.9, 0.999)$, and no weight decay. Training is conducted in fp16 mixed precision with `torch.compile` for improved throughput, together with gradient clipping and an exponential moving average (EMA) of the generative model weights for stable optimization. For DiT-L/2 and DiG-L/2, we adopt the Improved DDPM objective of Nichol & Dhariwal (2021), predicting both noise and variance. Hyperparameters for different backbones when using REPI alone (i.e., without REPA) are summarized in Table 6.

Table 6: **Hyperparameters and model configurations for standalone REPI**, shared across all ImageNet 256×256 and 512×512 experiments.

	SiT-B/2	SiT-L/2	SiT-XL/2	DiT-L/2	DiG-L/2
Architecture					
Number of layers	12	24	28	24	24
Hidden dimension	768	1024	1152	1024	1024
Number of heads	12	16	16	16	16
REPI					
λ	2.0	2.0	2.0	2.0	4.0
Injection layer	6	8	8	8	8
Scaffold duration	20K	20K	20K	20K	20K
K/V projection layer	Linear	Linear	Linear	Linear	Linear
Optimization					
Batch size	256	256	256	256	256
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW
Learning rate	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}
(β_1, β_2)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)
Weight decay	0	0	0	0	0
Diffusion					
Objective	Linear interpolants	Linear interpolants	Linear interpolants	Improved DDPM	Improved DDPM
Prediction	Velocity	Velocity	Velocity	Noise and variance	Noise and variance
Sampler	Euler–Maruyama	Euler–Maruyama	Euler–Maruyama	Respaced DDPM	Respaced DDPM
Sampling steps	250	250	250	250	250

Combining REPI with REPA. When combining with REPA, all settings are kept consistent with those used for REPI alone (Table 6), except that the injection layer is changed to layer 4 and λ is set to 0.25. For REPA, we adopt all of its default settings, including the alignment depth and alignment loss weight. Ablations for these combination-specific settings are reported in Appendix C.

Pretrained encoders. For our main results, we adopt DINOv2-B/14 as the pretrained visual encoder, following common practice in prior representation alignment work. In Section 5.3, we further evaluate REPI's generality across a broad range of pretrained encoders, including MAE (He et al., 2022), MoCoV3 (Chen et al., 2021), CLIP (Radford et al., 2021), DINOv2 (Oquab et al., 2024), WebSSL (Fan et al., 2025), I-JEPA-H (Assran et al., 2023), and DeiT-III (Touvron et al., 2022).

Training cost. All experiments are conducted on four NVIDIA H200 GPUs. Table 7 reports the total training time for 100K steps on SiT-XL/2. As shown, REPI introduces only marginal overhead over REPA (+3.2% training time), while substantially improving generation quality, reducing FID by 25.2% and improving IS by 22.7%. Combining REPA and REPI further amplifies these gains, achieving a 39.3% reduction in FID and a 43.6% improvement in IS, at a modest additional cost of only 3.9% training time over REPA.

Table 7: **Total training time** for 100K steps on SiT-XL/2 using four NVIDIA H200 GPUs. Percentages in parentheses denote the relative change with respect to REPA.

Method	Time (h)	FID↓	IS↑
REPA	4.63	19.40	67.4
REPI	4.78 (+3.2%)	14.51 (−25.2%)	82.7 (+22.7%)
REPA + REPI	4.81 (+3.9%)	11.78 (−39.3%)	96.8 (+43.6%)

B Details of Oracle Injection Variants

As described in Section 3, the oracle experiment is a diagnostic evaluation that assumes access to clean-image encoder representations at inference time. We investigate six injection targets: (1) hidden state, (2) attention output, (3) Q/K/V, (4) K/V, (5) K only, and (6) V only. The corresponding injection interfaces within a transformer block are illustrated in Figure 8. All six variants inject at block 8 of SiT-XL/2.

Hidden state. The injection source is the encoder output, and the target is the hidden state of a transformer block (i.e., the block's output). As shown in Table 1, this setting fails catastrophically, yielding an FID of 212.5, because overwriting the block's output discards the noisy-input-dependent state accumulated by all preceding blocks.

Attention output. The injection source is the encoder output, and the target is the attention output of a transformer block. As shown in Figure 8, this setting allows the noisy-input-dependent state to propagate to subsequent layers through the residual shortcut.

Q/K/V. The injection source is the Q/K/V of the encoder's last attention layer, and the target is the corresponding Q/K/V of a transformer block. We evaluate four variants: Q/K/V, K only, V only, and K/V, each replacing the corresponding component(s). In all four variants, the noisy-input-dependent state can still propagate to subsequent layers through the residual shortcut, as well as through whichever Q/K/V components remain unreplaced.

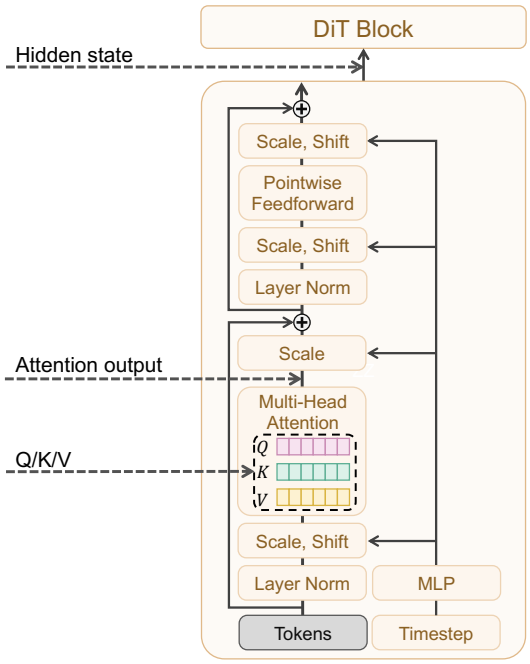


Figure 8: Illustration of injection targets within a DiT block.

C Additional Ablation Studies

We provide additional ablations of REPI when combined with REPA, examining the injection layer and internalization loss weight. Throughout these experiments, we retain REPA's default settings, including its alignment layer and alignment loss weight. We also report detailed metrics for the injection-target ablation, complementing Table 5 in the main paper.

Injection layer. Table 8 reports results obtained by varying the REPI injection layer while fixing the REPA alignment layer at layer 8. Placing the injection layer before the alignment layer yields better performance than placing it after, with injection at layer 4 achieving the best result. As discussed in Section 4.4, placing REPI first allows the subsequent REPA objective to supervise a representation that has already incorporated the injected encoder structure, encouraging this structure to propagate through subsequent layers. By contrast, injecting after alignment modifies the subsequent computation and may weaken the influence of the earlier alignment supervision.

Internalization loss weight. Table 9 reports ablations on the internalization loss weight λ when combined with REPA. Our method remains robust to the choice of λ in this setting: varying it from 0.125 to 1.0 yields similar FID scores, ranging between 11.78 and 12.68, substantially below vanilla SiT-XL's 39.4.

Injection target. We further provide detailed results for the ablation on injection targets in Table 10. All variants improve upon vanilla SiT, with K/V injection achieving the best performance.

Table 8: **Ablation results on the REPI injection layer when combined with REPA.** The REPA alignment layer is fixed at its default, layer 8. All results are reported on SiT-XL/2 at 100K training steps on ImageNet 256×256 without classifier-free guidance.

Layer	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
2	12.62	5.66	91.56	0.70	0.61
4	**11.78**	**5.58**	**96.77**	**0.70**	**0.61**
6	12.79	5.64	91.55	0.69	0.61
8	14.70	5.79	83.75	0.68	0.61
10	14.61	5.78	82.81	0.68	0.61
12	14.02	5.76	84.30	0.68	0.61

Table 9: **Ablation results on the internalization loss weight when combined with REPA.** All results are reported on SiT-XL/2 at 100K training steps on ImageNet 256×256 without classifier-free guidance.

λ	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
0.125	12.13	5.80	95.17	0.70	0.61
0.25	**11.78**	**5.58**	**96.77**	**0.70**	**0.61**
0.5	12.02	6.37	95.13	0.70	0.60
1.0	12.68	7.05	93.18	0.69	0.60

Table 10: **Ablation results on the injection target.** All results are reported on SiT-XL/2 at 100K training steps on ImageNet 256×256 without classifier-free guidance.

Injection target	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
Attention output	16.34	5.89	78.62	0.67	0.60
Q/K/V	17.40	5.91	71.60	0.67	0.60
K only	15.81	5.89	77.57	0.67	0.60
V only	15.53	**5.52**	79.31	0.68	0.60
K/V	**14.51**	5.57	**82.71**	**0.68**	**0.60**

D Detailed Quantitative Results

Table 11 provides more detailed quantitative results, complementing those in Table 2 of the main paper. All results are obtained using the same checkpoints and evaluation protocol, without classifier-free guidance. We compare REPA, REPI, and REPA + REPI at different training steps. REPI consistently outperforms REPA, and combining the two methods yields further improvements.

Table 11: **Detailed quantitative results across different SiT and DiT models.** All results are obtained on ImageNet 256×256 without classifier-free guidance.

Model	#Params	Iter.	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
SiT-B/2 (Ma et al., 2024)	130M	400K	33.0	6.50	43.7	0.53	0.63
+ REPA	130M	100K	49.5	7.00	27.5	0.46	0.59
+ REPA	130M	200K	33.2	6.68	43.7	0.54	0.63
+ REPA	130M	400K	24.4	6.40	59.9	0.59	0.65
+ REPI	130M	100K	35.9	6.65	40.3	0.54	0.62
+ REPI	130M	200K	25.0	6.74	59.6	0.59	0.65
+ REPI	130M	400K	19.5	6.49	74.3	0.61	0.65
+ REPA + REPI	130M	100K	35.8	7.29	42.1	0.53	0.62
+ REPA + REPI	130M	200K	24.0	7.07	62.5	0.59	0.64
+ REPA + REPI	130M	400K	18.3	6.50	79.2	0.62	0.65
SiT-L/2 (Ma et al., 2024)	458M	400K	18.8	5.30	72.0	0.64	0.64
+ REPA	458M	100K	24.1	6.25	55.7	0.62	0.60
+ REPA	458M	200K	14.0	5.18	86.5	0.67	0.64
+ REPA	458M	400K	9.7	5.20	109.2	0.69	0.65
+ REPI	458M	100K	19.0	5.48	67.8	0.65	0.62
+ REPI	458M	200K	12.1	5.16	96.2	0.68	0.64
+ REPI	458M	400K	9.0	5.10	115.5	0.69	0.65
+ REPA + REPI	458M	100K	14.7	5.63	84.9	0.67	0.62
+ REPA + REPI	458M	200K	10.1	5.21	109.2	0.69	0.64
+ REPA + REPI	458M	400K	8.2	5.26	124.4	0.69	0.65
SiT-XL/2 (Ma et al., 2024)	675M	7M	8.3	6.30	131.7	0.68	0.67
+ REPA	675M	100K	19.4	6.06	67.4	0.64	0.61
+ REPA	675M	200K	11.1	5.05	100.4	0.69	0.64
+ REPA	675M	400K	7.9	5.06	122.6	0.70	0.65
+ REPI	675M	100K	14.5	5.57	82.7	0.68	0.60
+ REPI	675M	200K	8.9	4.96	113.3	0.70	0.63
+ REPI	675M	400K	7.2	5.03	131.4	0.70	0.66
+ REPA + REPI	675M	100K	11.8	5.58	96.8	0.70	0.61
+ REPA + REPI	675M	160K	8.2	4.90	118.9	0.71	0.63
+ REPA + REPI	675M	200K	7.4	4.86	127.3	0.71	0.64
+ REPA + REPI	675M	400K	6.3	5.02	140.7	0.71	0.66
DiT-L/2 (Peebles & Xie, 2023)	458M	400K	21.4	6.7	63.7	0.62	0.63
+ REPA	458M	100K	32.9	7.44	44.2	0.55	0.63
+ REPA	458M	200K	20.4	7.06	70.1	0.62	0.64
+ REPA	458M	400K	14.5	6.97	91.8	0.64	0.65
+ REPI	458M	100K	26.0	6.44	53.3	0.60	0.62
+ REPI	458M	200K	17.8	6.71	75.3	0.63	0.63
+ REPI	458M	400K	13.6	6.52	93.3	0.66	0.65
+ REPA + REPI	458M	100K	25.0	7.16	57.9	0.60	0.63
+ REPA + REPI	458M	200K	16.1	6.81	84.3	0.64	0.64
+ REPA + REPI	458M	400K	12.2	6.70	101.7	0.66	0.66

E ADDITIONAL QUALITATIVE RESULTS

Figures 9–23 present additional qualitative samples generated by REPA + REPI on SiT-XL/2, trained for 400K steps. Figure 24 presents additional samples at 512×512 resolution. All samples are generated with classifier-free guidance at scale $w = 4.0$.



Figure 9: Generated samples from REPA + REPI on SiT-XL/2, trained for 400K steps, using classifier-free guidance ($w = 4.0$), conditioned on class "great white shark" (2).



Figure 10: Generated samples from REPA + REPI on SiT-XL/2, trained for 400K steps, using classifier-free guidance ($w = 4.0$), conditioned on class "loggerhead sea turtle" (33).



Figure 11: Generated samples from REPA + REPI on SiT-XL/2, trained for 400K steps, using classifier-free guidance ($w = 4.0$), conditioned on class "macaw" (88).



Figure 12: Generated samples from REPA + REPI on SiT-XL/2, trained for 400K steps, using classifier-free guidance ($w = 4.0$), conditioned on class "Blenheim Spaniel" (156).



Figure 13: Generated samples from REPA + REPI on SiT-XL/2, trained for 400K steps, using classifier-free guidance ($w = 4.0$), conditioned on class "Border collie" (232).



Figure 14: Generated samples from REPA + REPI on SiT-XL/2, trained for 400K steps, using classifier-free guidance ($w = 4.0$), conditioned on class "Arctic wolf" (270).



Figure 15: Generated samples from REPA + REPI on SiT-XL/2, trained for 400K steps, using classifier-free guidance ($w = 4.0$), conditioned on class "castle" (483).



Figure 16: Generated samples from REPA + REPI on SiT-XL/2, trained for 400K steps, using classifier-free guidance ($w = 4.0$), conditioned on class "convertible" (511).



Figure 17: Generated samples from REPA + REPI on SiT-XL/2, trained for 400K steps, using classifier-free guidance ($w = 4.0$), conditioned on class "ice cream" (928).



Figure 18: Generated samples from REPA + REPI on SiT-XL/2, trained for 400K steps, using classifier-free guidance ($w = 4.0$), conditioned on class "cheeseburger" (933).



Figure 19: Generated samples from REPA + REPI on SiT-XL/2, trained for 400K steps, using classifier-free guidance ($w = 4.0$), conditioned on class "buckeye" (990).



Figure 20: Generated samples from REPA + REPI on SiT-XL/2, trained for 400K steps, using classifier-free guidance ($w = 4.0$), conditioned on class "cliff" (972).



Figure 21: Generated samples from REPA + REPI on SiT-XL/2, trained for 400K steps, using classifier-free guidance ($w = 4.0$), conditioned on class "coral reef" (973).



Figure 22: Generated samples from REPA + REPI on SiT-XL/2, trained for 400K steps, using classifier-free guidance ($w = 4.0$), conditioned on class "lakeshore" (975).



Figure 23: Generated samples from REPA + REPI on SiT-XL/2, trained for 400K steps, using classifier-free guidance ($w = 4.0$), conditioned on class "volcano" (980).



Figure 24: Generated samples (512×512) from REPA + REPI on SiT-XL/2, trained for 400K steps, using classifier-free guidance ($w = 4.0$).