
U-REPA: Aligning Diffusion U-Nets to ViTs

Yuchuan Tian¹, Hanting Chen², Mengyu Zheng³, Yuchen Liang⁴, Chao Xu¹, Yunhe Wang²

¹ State Key Lab of General AI, School of Intelligence Science and Technology, Peking University.

² Huawei Noah’s Ark Lab. ³ The University of Sydney. ⁴ School of Mathematical Sciences, Peking University.

tianyc@stu.pku.edu.cn, {chenhanting, yunhe.wang}@huawei.com

xuchao@cis.pku.edu.cn

Abstract

Representation Alignment (REPA) that aligns Diffusion Transformer (DiT) hidden-states with ViT visual encoders has proven highly effective in DiT training, demonstrating superior convergence properties, but it has not been validated on the canonical diffusion U-Net architecture that shows faster convergence compared to DiTs. However, adapting REPA to U-Net architectures presents unique challenges: (1) different block functionalities necessitate revised alignment strategies; (2) spatial-dimension inconsistencies emerge from U-Net’s spatial downsampling operations; (3) space gaps between U-Net and ViT hinder the effectiveness of tokenwise alignment. To encounter these challenges, we propose **U-REPA**, a representation alignment paradigm that bridges U-Net hidden states and ViT features as follows: Firstly, we propose via observation that due to skip connection, the middle stage of U-Net is the best alignment option. Secondly, we propose upsampling of U-Net features after passing them through MLPs. Thirdly, we observe difficulty when performing tokenwise similarity alignment, and further introduces a manifold loss that regularizes the relative similarity between samples. Experiments indicate that the resulting U-REPA could achieve excellent generation quality and greatly accelerates the convergence speed. With CFG guidance interval, U-REPA could reach $FID < 1.5$ in 200 epochs or 1M iterations on ImageNet 256×256 , and needs only **half** the total epochs to perform better than REPA under *sd-vae-ft-ema*. Codes: <https://github.com/YuchuanTian/U-REPA>

1 Introduction

Representation Alignment (REPA) [45], a methodology that aligns features from Diffusion Transformers (DiT) [30] to modern visual encoders, has been demonstrated to significantly accelerate DiT training. This approach holds particular significance given the growing prominence of DiTs, which have gained mainstream adoption in diffusion models and are extensively applied across image generation [4; 11; 23] and video generation domains [48; 22; 19]. However, emerging empirical evidence suggests that U-Net [33] architectures might present a more advantageous alternative to DiTs in certain scenarios [18; 7; 39; 38]: U-Net-based models exhibit substantially faster convergence while achieving generation quality comparable to their transformer-based counterparts. This dichotomy motivates our core research inquiry - can modern Vision Transformer (ViT [10])-based visual encoders be effectively adapted to guide diffusion U-Net training through alignment mechanisms similar to REPA, thereby potentially elevating the convergence speed ceiling of diffusion models?

However, establishing effective alignment between U-Net architectures and ViT-based encoders presents challenges. Unlike Diffusion Transformers (DiTs) that share structural similarities with Vision Transformers, U-Net architectures exhibit fundamentally different operational characteristics. Specifically, both DiT and ViT adopt isotropic architectures composed of uniformly stacked transformer blocks, which inherently facilitates straightforward parameter alignment between the two

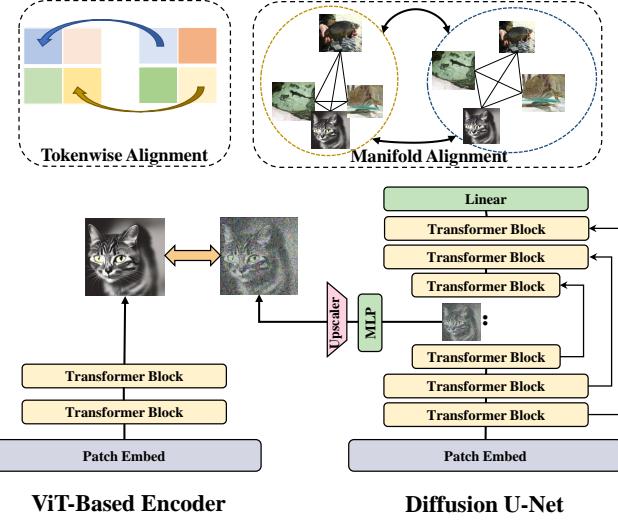


Figure 1: **The proposed U-REPA framework.** We investigated and found that semantic-rich intermediate layers are the best for representation alignment, dimension and space gaps hinders alignment efficacy. To counter these challenges, we scale-up features and propose manifold alignment.

frameworks. In contrast, U-Net’s skip connections create strong interdependencies between shallow and deep network layers by linking them together, resulting in different feature propagation dynamics. This architectural disparity renders conventional representation alignment strategies developed for DiT architectures inapplicable to U-Net frameworks. Furthermore, the progressive downsampling operations in U-Net generate feature maps with spatial dimension mismatches compared to the fixed-scale feature representations in ViT encoders, introducing additional complexity in establishing cross-architectural correspondence. In addition, features from high-stage U-Net and ViT have large space gaps, forming a barrier for cosine similarities as metrics. Forcibly using tokenwise similarity as loss is not necessarily the best option. This induces us to rethink about the optimization objective.

In order to conquer these challenges, we propose **U-REPA**, a framework that aligns U-Net hidden states to features from ViT encoders. Firstly, Our analysis reveals that skip connections fundamentally alter the functional specialization of transformer blocks in U-Net architectures. By establishing direct dependencies between early-stage and late-stage layers, these cross-connections induce a hierarchical redistribution of semantic information, with intermediate blocks exhibiting the highest semantic density. This pattern was empirically verified through controlled ablation studies on DiT augmented with skip connections, where progressive layer-wise evaluations demonstrated peak semantic richness at median network depths.

The intermediate higher-stage layers, which contain semantically dense representations, require precise alignment with the ViT-based visual encoder. However, these critical layers undergo spatial downsampling in the U-Net architecture, necessitating explicit spatial dimension reconciliation between U-Net features and ViT features during representation alignment. Through empirical exploration of various resolution-matching strategies, we identified an optimal solution: performing linear transformation via MLP on U-Net features prior to upsampling operations, which achieves superior alignment performance compared to alternative approaches.

Further analysis revealed a fundamental incompatibility of measuring cosine similarity between the feature spaces of U-Net and ViT encoders. Enforcing strict token-wise similarity constraints proves excessively rigid due to inherent architectural discrepancies. To address this, we introduce a manifold loss that implements soft alignment through relational regularization. This loss operates on the relative geometric relationships between samples rather than imposing direct feature correspondence, thereby accommodating cross-architectural variations. Comprehensive experiments demonstrate that our proposed U-REPA framework effectively bridges the U-Net-ViT alignment gap while preserving the distinct advantages of both architectures.

Our contributions are as follows:

1. We identify U-Net’s representation alignment to be a challenge due to different block functionalities, spatial-dimension inconsistencies, and larger feature space gaps.
2. We evaluate the contribution of downsampling and skips in U-Net and demonstrate U-Net’s potential advantage over DiTs.
3. We propose U-REPA, a framework that evaluates layers, investigates the best scale-up policy, and introduces manifold-space loss in aid of alignment.
4. We conduct experiments and verify the effectiveness of our U-REPA framework in terms of fast convergence. Specifically, U-REPA reaches $FID < 1.5$ in just 200 epochs on ImageNet 256×256 ; and it reaches 1.41 FID, with only half the epochs of REPA under the same training setting.

2 Related Work

The development of diffusion architectures. The conventional diffusion works [17; 35; 36; 9] leverages a U-Net [33] architecture, whose basic block is a concatenation of convolution layers and self-attention. More recent architectural innovations in diffusion models have witnessed a paradigm shift from conventional U-Net frameworks toward transformer-based architectures. The emergence of Diffusion Transformers [1; 30] demonstrates their competitive performance despite abandoning the inductive biases inherent in U-Net designs. U-ViT [1] represents an intermediate architecture that preserves U-Net’s hierarchical structure but replaces convolutional blocks with transformer layers, notably omitting the traditional downsampling operations. Subsequent developments have further streamlined the architecture: DiT [30] adopts a pure transformer backbone with isotropic scaling, while SiT [26] integrates the transformer architecture into the RectifiedFlow framework. Some other works either improve the micro-designs [6; 25], or focuses on architectural efficiency [3; 42; 40].

In contrary to these DiT works, some works still sticks to U-Net architectures and offer valuable rethinking on this conventional architectural preference: in pixel-space image generation, works including SimpleDiffusion [18] and HourglassDiT [7] still sticks to U-Net; in with its variants like U-DiT [39], Playground v3 [24], and DiC [38] extending its success to latent-space diffusion through simple Conv3 \times 3 designs. While these implementations empirically validate U-Net’s accelerated convergence and stable training dynamics compared to transformer-based alternatives, current research predominantly focuses on proposing architectural modifications rather than uncovering the reasons of U-Net’s superior diffusion performance.

Techniques for better DiT performance. Building upon the success of self-supervised learning [16], MDT [13] and MaskDiT [47] pioneer masked image modeling in diffusion frameworks by adaptively masking a good proportion of input patches during training. Other than the masking strategy, a bunch of diffusion works refer to higher-level semantic guidance from off-the-shelf pretrained models that significantly improves generation quality. REPA [45] establishes feature alignment between ViT-based encoder embeddings and diffusion latent spaces through contrastive learning. LightningDiT [44] innovates through an improved VAE distilled from MAE [16] and DINOv2 [28]. Ma et al. [27] introduces CLIP [31] and DINO [2] to verify inference-time scaling of diffusion models. These methods demonstrate that a higher-level semantic-rich feature-map from pretrained vision encoders is helpful to diffusion-based generation.

3 Method

3.1 Preliminaries: REPA for DiT

Representation Alignment (REPA) [45] distills Diffusion Transformers with semantic features from off-the-shelf ViT-based vision encoders (e.g. DINOv2 [28], CLIP [31], MAE [16], et cetera). Given a ViT-based vision encoder f and clean image \mathbf{x}_* , let $\mathbf{y}_* = f(\mathbf{x}_*) \in \mathbb{R}^{N \times D}$ denote its patch embeddings, where N and D represent the number of patches and embedding dimension, respectively. REPA establishes feature alignment between \mathbf{y}_* and the projected diffusion encoder outputs $h_\phi(\mathbf{h}_t)$, where $\mathbf{h}_t = f_\theta(\mathbf{z}_t)$ is the latent representation from the diffusion transformer at timestep t , and h_ϕ is a trainable multilayer perceptron (MLP).

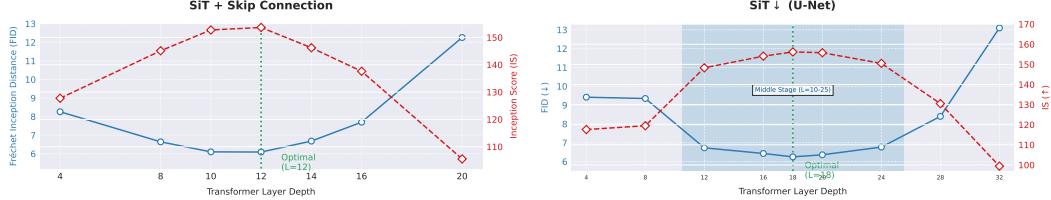


Figure 2: **Investigating alignment with respect to encoder depths on diffusion models with skip connections.** **Left:** SiT with skip connections. Due to the change of block functionalities due to newly established skip dependencies, the most optimal encoder depth is shifted towards the middle of the model. **Right:** SiT \downarrow , the U-Net-based SiT model. Shadowed region represents higher U-Net stage. The plot infers that stage transitions (downsampling & upsampling in U-Net) bring large block functionality gaps. Alignment within higher U-Net stage is thus necessary for alignment performance.

The alignment is enforced by maximizing token-wise feature similarities, *i.e.* the similarity of a token from DiT hidden-state with its corresponding counterpart in the ViT encoder feature:

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

where $\text{sim}(\cdot, \cdot)$ denotes a similarity metric (e.g., cosine similarity). Typically, \mathbf{z}_t is adopted as the output from early layers (the original work adopts layer index 8) in DiT for better alignment. This alignment term is combined with the basic flow-based diffusion objective (*i.e.* SiT [26]) through a tunable coefficient $\lambda > 0$, and the final loss for diffusion model training is formulated as follows:

$$\mathcal{L} := \mathcal{L}_{\text{velocity}} + \lambda \mathcal{L}_{\text{REPA}}. \quad (2)$$

3.2 Evaluating the Potential of U-Net

In diffusion models, U-Net and isotropic architectures (e.g., DiT) exhibit distinct design philosophies. While DiT achieves state-of-the-art results through scalability and integration with advanced techniques, U-Net-based methods emphasize faster convergence [39]. To dissect U-Net’s efficacy, we isolate its two core components: skip connections and downsampling.

1. **Skip Connections:** Provide shortcuts between encoder and decoder layers, theoretically aiding gradient flow and feature reuse.
2. **Downsampling:** Reduces spatial resolution (typically by a scale factor of 2 at each stage) to enable hierarchical, multi-scale feature learning. Critically, downsampling is always paired with skip connections to mitigate information loss.

Toy experiments on U-Net components. On top of DiT, we perform toy experiments that reveal the contribution of components mentioned above.

ImageNet 256 \times 256, DiT 400K, cfg=1			
Model	FLOPs (G)	FID \downarrow	IS \uparrow
DiT-XL/2	118.6	19.47	-
DiT-XL/2*	118.6	20.05	66.74
+ Skip Connections	114.1	19.86	67.29
+ Downsampling	108.8	13.78	88.93
DiT \downarrow -XL/2 (+Tricks)	108.8	11.02	100.35

Table 1: **Evaluating the contribution of U-Net components in terms of fast convergence.** Experiments are conducted using hyperparameters from [30] for 400K iterations. Model depth is changed when a modification is made such that the overall FLOPs is kept almost the same with DiT.

This suggests that U-Net’s fast-convergence advantages primarily stem from multi-scale hierarchical modeling via downsampling, not skip connections. Downsampling compresses features into compact, semantically rich representations, accelerating learning while maintaining information flow through skip-augmented decoder layers. However, skip connection is not useless as it compensates for the information loss due to downsampling.

Building on this insight, we propose $\text{DiT}\downarrow$ (and $\text{SiT}\downarrow$ for the flow-based version, following the naming convention of [26]) by adding tricks of RoPE [37] and SwiGLU following previous work [6; 39]

3.3 Aligning U-Net to ViT Encoders

Since U-Net has good potentials to achieve excellent generation, we are motivated to investigate whether REPA could also work on U-Net. We are first focused on the block functionality pattern and investigates the most optimal position for alignment; then we are interested in feature size alignment problems; lastly, we are dedicated to merging space gaps between features from U-Net and ViT, respectively.

Position for alignment. Regarding the concern that block functionalities differ between U-Net and DiT, the comparison of prior studies [12; 5] demonstrates divergent hierarchical specialization: U-Net architectures typically employ mid-network layers for high-level semantic synthesis while reserving shallow layers for low-level image refinement, whereas DiTs exhibit a totally different pattern - early layers primarily govern semantic-rich outline formation with deeper layers handling detailed image refinement.

These previous findings find empirical support in REPA’s experimental findings, where representation alignment proves most effective when applied to initial transformer blocks. This phenomenon stems from DiT’s early layers encoding semantic-rich representations that align well with the semantically dense outputs of ViT-based visual encoders, enabling meaningful guidance. Unlike the straightforward DiT architecture, the inherent skip connections in U-Net architectures induce fundamentally distinct block functionality compared to Diffusion Transformers. While all blocks in ViT or DiT maintain homogeneous computational roles, following a continuous flow of transition from input to output, U-Net’s cross-layer shortcuts establish direct dependencies between shallow and deep layers, fundamentally altering feature-map evolution patterns. As shown in Fig. 2 (R), DiTs with skips indicates median layer is the best for representation alignment. The same pattern goes for U-Net (Fig. 2 (L)) despite the downsampling stage.

Feature size alignment. The implementation of alignment between Diffusion U-Net’s median stage and ViT encounters a critical spatial resolution dilemma stemming from architectural disparities. While our analysis identifies mid-network U-Net features as optimal semantic carriers, their spatial dimensions drastically differ from ViT’s full-resolution token sequence. This dimensional mismatch obstructs REPA’s token-wise similarity computation, which requires strict cardinality matching between compared features.

In order to align two feature-maps (*i.e.* from U-Net and ViT encoder, respectively), from the macro level we advocate for upscaling the smaller-sized U-Net features rather than downscaling the larger visual encoder features. This design principle stems from the critical observation that compressing ViT’s high-resolution features to match U-Net’s reduced dimensions inevitably discards fine-grained visual information, thereby degrading alignment effectiveness. Preserving ViT’s native resolution while expanding U-Net’s bottleneck features proves essential for maintaining semantic fidelity.

At the implementation level, we empirically evaluated various upscaling strategies for U-Net features:

1. **Upscale first and then MLP:** feature upsampling is performed before passing the feature into the MLP.
2. **Upscale within MLP:** the MLP also acts as a feature upsampler that receives a low-resolution input and outputs a high-resolution one via linear mapping and pixel un-shuffling.
3. **MLP first and then upscale:** the feature from higher-stage U-Net is first passed through MLP and then upsampled.

Among the three options, we found “MLP first and then upscale” is the best both in terms of performance and efficiency (minimum FLOPs cost), which will be discussed in the Ablation Study in Sec. 4.

Manifold space alignment. Though we select the most suitable U-Net feature for alignment and keep dimensions between U-Net and ViT features aligned, challenges remain in feature space compatibility. First, compared to the structural congruence between DiT and ViT encoders, the architectural discrepancy of U-Net (with its skip connections and hierarchical downsampling) creates a more pronounced feature distribution gap between U-Net hidden states and visual encoder outputs.

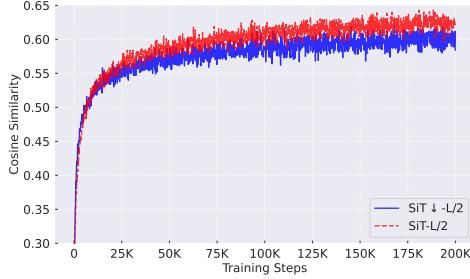


Figure 3: **The convergence of average tokenwise similarities.** While SiT-L/2 could achieve better tokenwise similarities, SiT \downarrow converges at a lower similarity value, indicating difficulties of feature alignment.

Second, the dimensional transformation required for alignment inevitably modifies U-Net’s native feature space characteristics.

Some recent works on Diffusion U-Net [41; 34] reveals that higher-stage U-Net features are low-frequency subspaces that discards higher frequency components, including noises. Gaps are inevitable when evaluating the cosine similarities of detail-rich, high-frequency-rich vectors and flat, low-frequency-dominated vectors. In this sense, strict token-level alignment constraints like the original REPA loss prove suboptimal under these conditions, as they assume implicit feature space homogeneity between aligned modalities.

As is shown in Fig. 3, we conducted continuous measurements of token-wise cosine similarity against ViT features during training. The two models that we compare are SiT-L/2 of isotropic, standard transformer architecture and SiT \downarrow -L/2 of U-Net architecture. Our experiments revealed a characteristic learning trajectory: while U-Net achieves slightly faster similarity improvement in early training phases - thanks to skip connection that helps convergence - its progress stagnates beyond this point, ultimately plateauing at 0.60 - notably inferior to DiT’s sustained growth reaching around 0.63 similarity. The similarity gap between SiT and SiT \downarrow This phenomenon suggests that naively aligning U-Net with ViT encoders through angular similarity metrics alone encounters inherent limitations due to architectural incompatibilities.

Rather than strict token-wise regularization, we resort to looser objectives that does not require rigid angular alignment. Inspired by manifold knowledge distillation [14], we hold that aligning similarities between samples from the same feature space could be a promising solution. Hence, we define Manifold Loss \mathcal{L}_{ML} as

$$\mathcal{L}_{\text{ML}}(\theta, \phi) := -\mathbb{E}_{\mathbf{x}_*, \epsilon, t, i, j} [\text{d}(\mathbf{y}_*, h_\phi(\mathbf{h}_t))], \quad (3)$$

where

$$\text{d} := \|\text{sim}(\mathbf{y}_*^{[i]}, \mathbf{y}_*^{[j]}) - \text{sim}(h_\phi(\mathbf{h}_t^{[i]}), h_\phi(\mathbf{h}_t^{[j]}))\|_F^2. \quad (4)$$

In the formula, cosine similarity is adopted as the similarity metric, and F represents Frobenius Norm of matrices. By introducing affine hyperparameter w , the overall optimization target is then formulated as

$$\mathcal{L} := \mathcal{L}_{\text{velocity}} + \lambda (\mathcal{L}_{\text{REPA}} + w\mathcal{L}_{\text{ML}}). \quad (5)$$

3.4 Other Improvements

We also propose and evaluate some other improvements. Due to page limits, the proposed methods and corresponding ablations are enclosed in the Appendix.

4 Experiments

4.1 Experiment Setup

Experiment settings. Our implementation completely adheres to the training protocol established in REPA [45]. Following the architectural configuration of latent diffusion models [32], we employ the identical VAE variant (*sd-vae-ft-ema*) and adopt the AdamW optimizer. To ensure fair comparison, we maintain identical hyperparameter settings across all experiments: a global batch size of 256, fixed learning rate of $1e - 4$, and disabled weight decay (set to 0). (β_1, β_2) is set as $(0.9, 0.999)$. All experiments are conducted on the ImageNet 2012 benchmark [8] under a controlled environment with a fixed random seed (global seed=0). 8 NVIDIA A100 GPUs are used for main experiments.

For main experiments (Tab. 5), we apply guidance interval [21] $[0, 0.7]$ and SDE sampling according to the convention of REPA [45] for fair comparison. We select smaller *cfg* of 1.65, because we found it is better for our architecture, different from SiT. For all ablation experiments, we train models for 100K iterations, which is sufficient to show the trend of model performance; sampling is conducted with the default setting of the official REPA codebase, *i.e.* $cfg = 1.8$ in ODE and guidance interval $[0, 0.7]$.

Model settings. By aligning channel dimensions and FLOPs with standard Diffusion Transformers (DiTs or SiTs), our U-REPA-compatible variants maintain architectural parity while introducing critical adaptations for U-Net principles. The base model ($\text{SiT}\downarrow\text{-B}$) employs a stage arrangement of $[5,5,5]$, achieving 199.7G FLOPs. Scaling to larger models, we have the L variant (686.6M) and XL variant (954.4M params) that increases channel width (1024 vs. 1152 in base) through increased stage-wise block allocation ($[9,14,9]$ vs. $[10,16,10]$). Notably, when FLOPs are aligned, $\text{SiT}\downarrow$ models usually have more parameters than SiTs due to increased depth.

Model	Params (M)	FLOPs (G)	Patch Size	Channel	# Heads	Blocks in Stages
SiT\downarrow-B	199.7	24.1	2	768	12	$[5,5,5]$
SiT\downarrow-L	686.6	79.3	2	1024	16	$[9,14,9]$
SiT\downarrow-XL	954.4	109.3	2	1152	16	$[10,16,10]$

Table 2: **Configurations of $\text{SiT}\downarrow$ architecture at different model sizes.** The proposed $\text{SiT}\downarrow$ in U-Net architectures are aligned to DiTs in terms of FLOPs and channel dimension.

4.2 The Advantage of U-REPA

Comparing $\text{SiT}\downarrow$ with SiT at different scales. We evaluate our U-REPA alignment method on ImageNet 256 under a generation setting with $cfg = 1$ (REPA framework without classifier-free guidance). As shown in Table 3, our approach consistently improves generation quality while significantly reducing computational costs across model scales. For the base-size SiT-B/2 variant, integrating U-REPA achieves a 39.3% improvement in FID (from 24.4 to 15.3) with comparable FLOPs (24.1G vs. 23.0G) and identical training iterations (400K), demonstrating that feature alignment enhances parameter efficiency without additional training overhead. The acceleration effect becomes more pronounced in larger models: for SiT-L/2, U-REPA reduces required iterations by 42.9% (700K \rightarrow 400K) while simultaneously lowering FLOPs (79.3G vs. 80.8G) and achieving a 30.9% FID improvement (8.4 \rightarrow 5.8). Most notably, the XL-scale variant with U-REPA (*cf.* Fig. 5 for FIDs vs. Training iters) attains state-of-the-art FID (5.4) using **90%** fewer iterations (400K vs. 4M) and fewer FLOPs (108.8G vs. 118.6G) compared to the baseline, proving our method’s fast convergence.

We also demonstrate the advantage of the proposed U-REPA framework when measuring by parameters (rather than computation FLOPs), as shown in Fig. 4. Though U-Net brings extra parameters when FLOPs are aligned with DiTs, the advantage of $\text{SiT}\downarrow$ +U-REPA is obvious as depicted in the Parameter versus FID plot.

Convergence performance. We also compare our method with previous State-of-the-Arts, as shown in Tab. 5. Our proposed $\text{SiT}\downarrow$ +U-REPA achieves a competitive FID of 1.48 with only 200 training epochs, significantly outperforming existing methods in training efficiency. Notably, while state-of-the-art masked diffusion transformers like MDTv2-XL/2 require 1,080 epochs to reach 1.58 FID, our method attains better performance (1.48) with 80% fewer iterations. Even compared to the SOTA

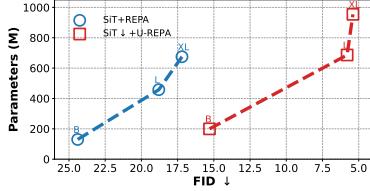


Figure 4: **Comparing SiT↓+U-REPA against SiT+REPA in terms of parameter scalability.** While the U-Net architecture makes the diffusion model parameter-rich compared with same-FLOPs Diffusion Transformers, SiT↓ models still outcompetes SiTs by large margins in terms of parameters.

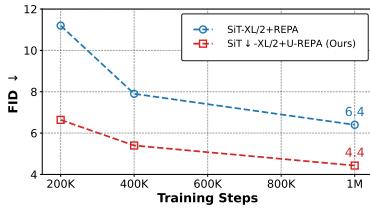


Figure 5: **Comparing SiT↓+U-REPA against SiT+REPA in terms of convergence speed.** SiT↓-XL/2 converges much faster than SiT-XL/2 with the help of U-REPA.

ImageNet 256×256, w/o cfg			
Model	FLOPs (G)	Iter.	FID↓
SiT-B/2+REPA	23.0	400K	24.4
SiT↓-B/2+U-REPA	24.1	400K	15.3
SiT-L/2+REPA	80.8	700K	8.4
SiT↓-L/2+U-REPA	79.3	400K	5.8
SiT-XL/2+REPA	118.6	4M	5.9
SiT↓-XL/2+U-REPA	108.8	400K	5.4

Table 3: **Comparing U-REPA against REPA across various model sizes without classifier-free guidance.** U-Nets equipped with U-REPA show excellent capabilities. Notably, U-REPA achieves 10× faster convergence compared with REPA in terms of performance w/o CFG.

ImageNet 256×256, w/ cfg				
Model	Dep.	Feat. Dim.	FID↓	IS↑
SiT↓-XL/2	4	16	9.42	117.6
SiT↓-XL/2 (REPA)	8	16	9.35	119.5
SiT↓-XL/2	12	8	6.73	148.3
SiT↓-XL/2	16	8	6.43	154.1
SiT↓-XL/2	18	8	6.25	156.2
SiT↓-XL/2	20	8	6.36	155.8
SiT↓-XL/2	24	8	6.77	150.5
SiT↓-XL/2	28	16	8.40	130.5
SiT↓-XL/2	32	16	13.10	99.4

Table 4: **Ablations on encoder depths for alignment in SiT↓.** Feat. Dim. stands for the spatial height& width at the certain layer. Compared with the default REPA setting, aligning at the centering layer (higher stage in U-Net) performs much better.

SiT-XL/2 + REPA baseline (800 epochs for 1.42 FID), our approach uses only 1/2 of the training epochs (400) while achieving better generation quality (1.41 FID). The results demonstrate that the proposed U-REPA establishes a new efficiency frontier for diffusion models.

4.3 Ablation Studies

Encoder depths. The ablation study on encoder layer depths for feature alignment (Tab. 4) coincides with the pattern of DiT with skip connections, as we analyzed in Sec. 4: despite progressive down-sampling operations that reduce spatial resolution, the centermost layers exhibit optimal alignment efficacy. For the SiT↓-XL/2 model, aligning features at layer 18 (midway through the 36-layer architecture) achieves peak performance with 6.25 FID and 156.2 IS, outperforming both shallower and deeper alignment points. This phenomenon persists even as the spatial dimension (Feat. Dim.) halves from 16×16 to 8×8 in the intermediate stage, indicating that semantic richness—not spatial resolution—dominates alignment quality. Performance degradation occurs when alignment takes place at shallower or deeper stages, even though the feature size is kept the same with DINO in these stages.

Alignment dimension choices. The comparative results in Table 6 reveal that upsampling U-Net’s higher-stage features (\uparrow_2) to match DINOv2’s native resolution achieves superior performance (5.72 FID, 161.6 IS), outperforming the alignment alternative in generation quality. This demonstrates that preserving ViT encoder’s original feature granularity during alignment is beneficial for alignment.

Feature-map upscale choices. Among the three upscaling options mentioned in Sec. 4, we figure out that upscaling U-Net hidden states after getting passed through MLP is the best option, achieving 5.72 FID and 161.6 IS. This option is also the most optimal one in terms of computation cost analysis.

ImageNet 256×256, w/ cfg		
Model	Epochs	FID↓
<i>Pixel diffusion</i>		
ADM-U [9]	400	3.94
VDM++ [20]	560	2.40
Simple diffusion [18]	800	2.77
<i>Latent Diffusion Transformer</i>		
U-ViT-H/2 [1]	240	2.29
DiffiT [15]	-	1.73
DiT-XL/2 [30]	1400	2.27
SiT-XL/2 [26]	1400	2.06
<i>Masked Diffusion Transformer</i>		
MaskDiT [47]	1600	2.28
MDTv2-XL/2 [13]	1080	1.58
<i>Representation Alignment</i>		
SiT-XL/2 + REPA [45]	800	1.42
SiT _↓ -XL/2 + U-REPA (Ours)	200	1.48
SiT_↓-XL/2 + U-REPA (Ours)	400	1.41

Table 5: **Comparing U-REPA against State-of-the-Art baselines with classifier-free guidance.** U-REPA could reach $FID < 1.5$ in merely 200 epochs and $FID = 1.41$ in 400 epochs; The proposed method converge 2× faster while achieving lower FID.

ImageNet 256×256, w/ cfg		
Alignment Choices	FID↓	IS↑
U-Net DINOv2 _{↓2}	5.99	158.8
U-Net_{↑2} DINOv2	5.72	161.6

Table 6: **Alignment dimension choices.** Up-sampling higher-stage U-Net features in alignment with ViT performs better due to less information loss.

ImageNet 256×256, w/ cfg		
Alignment	FID↓	IS↑
Upscale before MLP	5.84	158.5
Upscale in MLP	6.36	153.4
Upscale after MLP	5.72	161.6

Table 7: **Feature-map upscale choices.** Among the three options, Upscaling after passing through MLP performs best; and it has lower cost as the small-sized feature map is passed through MLP.

ImageNet 256×256, w/ cfg			
Model	w	FID↓	IS↑
SiT _↓ -XL/2+U-REPA	0	6.25	156.2
SiT _↓ -XL/2+U-REPA	2	5.81	160.8
SiT_↓-XL/2+U-REPA	3	5.72	161.6
SiT _↓ -XL/2+U-REPA	4	5.79	160.6

Table 8: **Adjusting weight w in Eq. 5.** Manifold loss boosts U-Net’s alignment performance. The most optimal result is taken at $w = 3$.

Manifold loss weight w . The ablation study on alignment weight w in Eq. 5 demonstrates a clear performance peak at $w = 3$ achieving the lowest FID (5.72) and highest IS (161.6) among tested configurations.

4.4 Higher Resolution Experiments

At the higher-resolution ImageNet 512×512 (w/ cfg) setting, U-REPA remains clearly superior to the REPA baseline (Tab. 9). Using SiT_↓-XL/2, U-REPA reduces FID from 2.44 to 2.21 and raises IS from 247.3 to 274.7. These results indicate that U-REPA’s benefits persist at 512 resolution, yielding better distributional fidelity and sample quality/diversity, and demonstrating strong scalability.

ImageNet 512×512, w/ cfg		
Alignment Choices	FID↓	IS↑
SiT-XL/2 + REPA	2.44	247.3
SiT_↓-XL/2 + U-REPA (Ours)	2.21	274.7

Table 9: **Comparing U-REPA against REPA on ImageNet 512×512.** On higher resolution, the proposed U-REPA still maintain a clear advantage.

4.5 The Energy Cost Advantage of U-REPA

We also assess the energy-cost advantage of U-REPA over REPA. We train on eight NVIDIA A100 GPUs and record each GPU’s power draw. Combining the measured power with the training duration,

ImageNet 256×256				
Model	Avg. Pow. (W)	Training Hours	Est. Energy (J)	FID↓
SiT-XL/2+REPA (4M iter)	373.2	302.7	3.25×10^9	1.42
SiT↓-XL/2+U-REPA (2M iter)	295.3	230.7	1.96×10^9	1.41

Table 10: **Energy cost comparison.** We compare the energy cost of U-REPA (at 2M iters) and REPA (at 4M iters). U-REPA could significantly reduce the cost of training a State-of-the-Art diffusion model.

we estimate the total energy consumed. The statistics for average power and estimated total energy used by all 8 GPUs are summarized in Tab. 10. Results indicate that our U-REPA method is "greener", costing far less energy.

Reducing training energy directly curbs operational CO_2 emissions. Methods that achieve comparable accuracy with lower energy, such as U-REPA vs. REPA in our study, advance both sustainability and the economic viability of large-scale AI.

5 Conclusion

In this paper, we propose U-REPA, an adapted version of REPA on Diffusion U-Net. We identify key challenges in U-Net hidden state alignment and show that U-REPA effectively bridges the gap between U-Net-based diffusion models and ViT-based encoders. By aligning intermediate features, resolving spatial mismatches via post-MLP upsampling, and enforcing manifold-aware regularization, U-REPA achieves faster convergence and an FID score of 1.41 on ImageNet-256×256 at 2M iters.

Acknowledgement. This work is supported by the National Key R&D Program of China under grant No. 2022ZD0160300 and the National Natural Science Foundation of China under grant No. 62276007. This work is funded by Peking University–BHP Carbon and Climate Wei-Ming PhD Scholars Program (Program Name: Research on Low-Carbon and Energy-Efficient Large Model Architectures; Program Number: WM202505). We sincerely thank Sibo Fang for his generous help during this project.

References

- [1] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 22669–22679. IEEE, 2023.
- [2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 9630–9640. IEEE, 2021.
- [3] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart- Σ : Weak-to-strong training of diffusion transformer for 4k text-to-image generation. *CoRR*, abs/2403.04692, 2024.
- [4] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart- α : Fast training of diffusion transformer for photorealistic text-to-image synthesis, 2023.
- [5] Pengtao Chen, Mingzhu Shen, Peng Ye, Jianjian Cao, Chongjun Tu, Christos-Savvas Bouganis, Yiren Zhao, and Tao Chen. δ -dit: A training-free acceleration method tailored for diffusion transformers, 2024.
- [6] Xiangxiang Chu, Jianlin Su, Bo Zhang, and Chunhua Shen. Visionllama: A unified llama interface for vision tasks. *CoRR*, abs/2403.00522, 2024.
- [7] Katherine Crowson, Stefan Andreas Baumann, Alex Birch, Tanishq Mathew Abraham, Daniel Z. Kaplan, and Enrico Shippole. Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers. *CoRR*, abs/2401.11605, 2024.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society, 2009.

[9] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat gans on image synthesis. In Marc’Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 8780–8794, 2021.

[10] 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. *CoRR*, abs/2010.11929, 2020.

[11] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

[12] Gongfan Fang, Xinyin Ma, and Xinchao Wang. Structural pruning for diffusion models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[13] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Mdtv2: Masked diffusion transformer is a strong image synthesizer, 2024.

[14] Zhiwei Hao, Jianyuan Guo, Ding Jia, Kai Han, Yehui Tang, Chao Zhang, Han Hu, and Yunhe Wang. Learning efficient vision transformers via fine-grained manifold distillation. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[15] Ali Hatamizadeh, Jiaming Song, Guilin Liu, Jan Kautz, and Arash Vahdat. Diffit: Diffusion vision transformers for image generation. *CoRR*, abs/2312.02139, 2023.

[16] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 15979–15988. IEEE, 2022.

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239, 2020.

[18] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 13213–13232. PMLR, 2023.

[19] Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *CoRR*, abs/2410.05954, 2024.

[20] Diederik P. Kingma and Ruiqi Gao. Understanding diffusion objectives as the ELBO with simple data augmentation. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[21] Tuomas Kynkänniemi, 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 Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[22] PKU-Yuan Lab and Tuzhan AI etc. Open-sora-plan, April 2024.

- [23] Black Forest Labs. Flux, August 2024.
- [24] Bingchen Liu, Ehsan Akhgari, Alexander Visheratin, Aleks Kamko, Linmiao Xu, Shivam Shirao, Chase Lambert, Joao Souza, Suhail Doshi, and Daiqing Li. Playground v3: Improving text-to-image alignment with deep-fusion large language models, 2024.
- [25] Zeyu Lu, Zidong Wang, Di Huang, Chengyue Wu, Xihui Liu, Wanli Ouyang, and Lei Bai. Fit: Flexible vision transformer for diffusion model. *CoRR*, abs/2402.12376, 2024.
- [26] 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. *CoRR*, abs/2401.08740, 2024.
- [27] Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. Inference-time scaling for diffusion models beyond scaling denoising steps, 2025.
- [28] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Trans. Mach. Learn. Res.*, 2024, 2024.
- [29] Byeongjun Park, Sangmin Woo, Hyojun Go, Jin-Young Kim, and Changick Kim. Denoising task routing for diffusion models, 2024.
- [30] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 4172–4182. IEEE, 2023.
- [31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021.
- [32] 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, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10674–10685. IEEE, 2022.
- [33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells III, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241. Springer, 2015.
- [34] Chenyang Si, Ziqi Huang, Yuming Jiang, and Ziwei Liu. Freeu: Free lunch in diffusion u-net. *CoRR*, abs/2309.11497, 2023.
- [35] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *CoRR*, abs/2010.02502, 2020.
- [36] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.
- [37] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
- [38] Yuchuan Tian, Jing Han, Chengcheng Wang, Yuchen Liang, Chao Xu, and Hanting Chen. Dic: Rethinking conv3x3 designs in diffusion models. *CoRR*, abs/2501.00603, 2025.
- [39] Yuchuan Tian, Zhijun Tu, Hanting Chen, Jie Hu, Chao Xu, and Yunhe Wang. U-dits: Downsample tokens in u-shaped diffusion transformers, 2024.
- [40] 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, 2025.

- [41] Christopher Williams, Fabian Falck, George Deligiannidis, Chris C. Holmes, Arnaud Doucet, and Saifuddin Syed. A unified framework for u-net design and analysis. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.
- [42] 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 image synthesis with linear diffusion transformer, 2024.
- [43] Yixing Xu, Chao Li, Dong Li, Xiao Sheng, Fan Jiang, Lu Tian, and Ashish Sirasao. Fdvit: Improve the hierarchical architecture of vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5950–5960, October 2023.
- [44] Jingfeng Yao and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. *CoRR*, abs/2501.01423, 2025.
- [45] 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. *CoRR*, abs/2410.06940, 2024.
- [46] Wangbo Zhao, Yizeng Han, Jiasheng Tang, Kai Wang, Yibing Song, Gao Huang, Fan Wang, and Yang You. Dynamic diffusion transformer. *arXiv preprint arXiv:2410.03456*, 2024.
- [47] Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *Trans. Mach. Learn. Res.*, 2024, 2024.
- [48] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all, March 2024.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and precede the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer **[Yes]** , **[No]** , or **[NA]** .
- **[NA]** means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "**[Yes]**" is generally preferable to "**[No]**", it is perfectly acceptable to answer "**[No]**" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "**[No]**" or "**[NA]**" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer **[Yes]** to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist",**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers.**

1. Claims

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

Answer: **[Yes]**

Justification: The abstract demonstrates our motivation, the proposed ideas and a brief summary of experiment results.

Guidelines:

- The answer NA 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 NA 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: The paper has discussed the limitations of the work in the appendix due to page limits.

Guidelines:

- The answer NA 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: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA 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: The paper fully discloses all the information needed to reproduce the main experimental results of the paper.

Guidelines:

- The answer NA 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: **[Yes]**

Justification: The paper will provide open access to the data and code during camera ready period.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.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://nips.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, etc.) necessary to understand the results?

Answer: **[Yes]**

Justification: This paper has specified all the training and test details.

Guidelines:

- The answer NA 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: **[NA]**

Justification: This is not relevant to this paper.

Guidelines:

- The answer NA 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: The paper has indicated sufficient information on the computer resources.

Guidelines:

- The answer NA 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: This research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA 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: [\[Yes\]](#)

Discussed in the appendix due to page limits.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [\[Yes\]](#)

Justification: This paper has described safeguards.

Guidelines:

- The answer NA 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: The utilization of code, data and models in this paper is in accordance with the license and the terms.

Guidelines:

- The answer NA 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: [\[NA\]](#)

Justification: This paper does not release new assets.

Guidelines:

- The answer NA 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: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA 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: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA 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 rigorosity, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: This paper is centered around LLM architecture.

Guidelines:

- The answer NA 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 (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

Appendix

A Other Improvements

In the appendix, we address the effect of proposed "Other Improvements" (Sec. 4), including "Time-aware MLP" and "Weight schedule of loss".

Time-aware MLP. Some works on in the diffusion task [29; 46] reveals that a channel dimension is particularly sensitive and useful to a certain subset of time during sampling. As the time-variant feature is to be aligned to time-invariant vision encoder features, we hold that alignment could perform better when the MLP is time-aware and extract time-invariant information out of diffusion features for alignment.

Inspired by the conditioning of canonical Diffusion U-Net [17; 36; 9] and DiT [30], we add a module to predict a pair of channel-wise shift& scale vector $(\gamma(t), \beta(t))$. The module is in parallel to MLP and follows the design of DiT's AdaLN, which is a concatenation of a SiLU and a Linear layer. The shift& scale vectors are imposed on the output MLP as follows:

$$h_\phi^t(\mathbf{h}_t^{[n]}) = \gamma(t) \odot h_\phi(\mathbf{h}_t^{[n]}) + \beta(t)$$

Weight schedule of loss. As is prompted in REPA [45], designing weight schedule is a future direction. We try various weight schedules (*i.e.* make λ in Eq. 2 a function $\lambda(t)$ with respect to time) but found that these schedules bring very limited improvements on the proposed U-REPA. Hence, we stick to the original constant weight strategy of REPA.

These improvements bring slight increases on the generation metrics. Hence, we do not include them in the main experiments for the simplicity of the method. The effects of proposed measures are shown in Tab. 11 and Tab. 12.

ImageNet 256×256, w/ cfg		
Alignment Choices	FID↓	IS↑
Ordinary MLP	5.72	161.6
Time-aware MLP	5.63	163.3

Table 11: The effect of time-aware MLPs.

ImageNet 256×256, w/ cfg		
Alignment Choices	FID↓	IS↑
Constant	5.72	161.6
$\max(1, t + 0.5)$	5.85	161.3
$\max(1, -t + 1.5)$	5.72	161.6
$\min(1, \max(-2t + 1.5, 2t - 0.5))$	5.58	164.0

Table 12: The effect of different weight schedules of loss.

B Additional Experiments

Evaluating SiT↓-XL/2. We also evaluated the proposed U-Net architecture on the Scalable Interpolant Transformers (SiT) framework without the guidance of REPA. The results are shown in Tab. 13.

Notably, though the amount of FID improvement brought by U-REPA is not as great as REPA (*i.e.* SiT / SiT+REPA vs. SiT↓+U-REPA), we hold that this comparison is invalid due to the following reasons:

1. As generation performance gets stronger, it is also becoming much harder to improve (especially for FID when it gets lower).
2. Aligning SiT↓ and ViT is much harder than aligning SiT and ViT, because the backbone of SiT and ViT encoders are very similar. Aligning SiT↓ to ViT encoder is a special case due to great architecture difference.

However, we hold that comparing REPA and U-REPA on the same model of SiT↓ is fair. The default REPA achieves FID 9.35 (as shown in Tab. 4) while our method achieves FID 5.72, both trained for 100K iterations with cfg and guidance interval adopted.

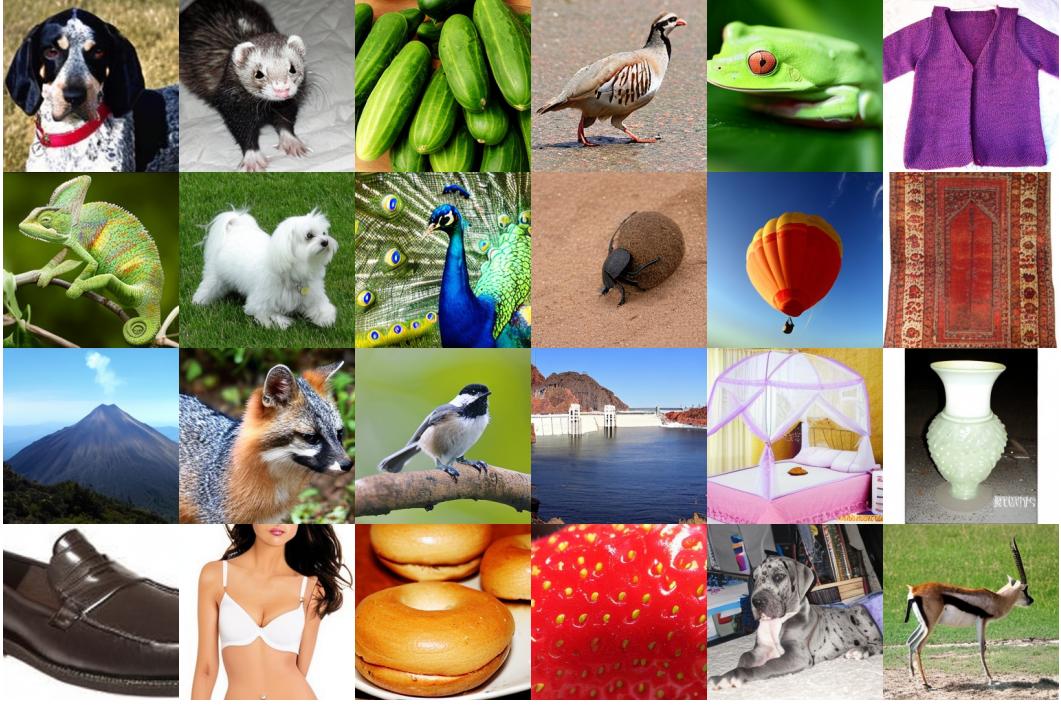


Figure 6: **Samples generated by $\text{SiT}\downarrow\text{+U-REPA}$ at 1M iterations.** The samples are generated following the setting of REPA, at $cfg = 4$. Best viewed on screen.

ImageNet 256×256, w/o cfg		
Model	Iter.	FID \downarrow
SiT-XL/2	400K	17.2
SiT\downarrow-XL/2	400K	9.2
SiT-XL/2+REPA	400K	7.9
SiT\downarrow-XL/2+U-REPA	400K	5.4

Table 13: **Evaluating the performance of $\text{SiT}\downarrow$.** $\text{SiT}\downarrow\text{-XL/2}$ performs much better than SiT-XL/2 , and U-REPA further reduces the FID of $\text{SiT}\downarrow\text{-XL/2}$ to 5.4 without classifier-free guidance.

Seed Sensitivity. In our paper, we take $seed = 0$ following the setting of REPA. We also tested other seeds ($seed = 1, 2$) in training to examine the seed sensitivity of our method, shown in Tab. 14. The experiments are run for 600K iterations with guidance interval and cfg , following REPA.

ImageNet 256×256, w/ cfg		
Model	seed	FID \downarrow
SiT\downarrow-XL/2+U-REPA	0	1.618
SiT\downarrow-XL/2+U-REPA	1	1.599
SiT\downarrow-XL/2+U-REPA	2	1.588
SiT\downarrow-XL/2+U-REPA	mean	1.602 ± 0.012

Table 14: **Examining seed sensitivity.** We selected $seed = 0, 1, 2$ and evaluate the performance with cfg . The performance fluctuation is limited to a narrow interval (approximately 0.01).

Ablations on the REPA Loss. We also conduct ablations on the REPA loss (\mathcal{L}_{REPA}) while leaving the proportion of manifold loss intact (keeping the multiplication λ_w fixed). The results are shown in Tab. 15.

ImageNet 256×256, w/ cfg				
λ	0.5	0.25	0 (No \mathcal{L}_{REPA})	
FID \downarrow	5.72	6.42	10.91	

Table 15: **Adjusting the hyperparameter for REPA loss λ in Eq. 5.** The REPA loss is vital for the generation performance; removing \mathcal{L}_{REPA} would cause a significant performance decay.



Figure 7: **Comparing the visual quality of SiT+REPA (upper row) and SiT \downarrow +U-REPA (lower row).** The samples are generated following the sampling strategy that yields the State-of-the-Art FIDs in respective methods. Best viewed on screen.

One-on-one visualization comparison. Apart from quantitative comparisons, we also provide qualitative comparisons in Fig. 7 by inserting the same random noise into trained SiT+REPA (at 4M iterations, FID 1.42) and SiT \downarrow +U-REPA (at 2M iterations, FID 1.41). The samples are not cherrypicked; we directly pick the first several samples at seed=0. Samples generated by SiT \downarrow +U-REPA has better visual quality.

C Limitations & Impact

Limitations and Future work. The U-Net architecture is a simple one with only one intermediate stage. We do not further refine the architecture as we want to show U-Net architectures as simple as SiT \downarrow could also achieve rapid convergence. Further improvements on the U-Net architecture includes efficient attention [42; 39], non-integer down& up scaling factors [43], and more use of convolutions [36; 38]. Besides, whether U-REPA could be applied to downstream diffusion tasks that rely heavily on U-Nets (e.g. Low-Level Vision) remains to be investigated.

Broader Impact. As a work centered around AIGC, it is probable that inappropriate contents may appear from the output. We should be aware of this negative societal impact.