No Other Representation Component Is Needed: Diffusion Transformers Can Provide Representation Guidance by Themselves

Dengyang Jiang[1,2†] **Mengmeng Wang**[3,2*] **Liuzhuozheng Li**[2] **Lei Zhang**[1]
Haoyu Wang[1] **Wei Wei**[1] **Guang Dai**[2] **Yanning Zhang**[1] **Jingdong Wang**[4]
[1]Northwestern Polytechnical University [2]SGIT AI Lab, State Grid Corporation of China
[3]Zhejiang University of Technology [4]Baidu Inc.

Quick overview at: https://vvvvvjdy.github.io/sra
Code is available at: https://github.com/vvvvvjdy/SRA

Abstract

Recent studies have demonstrated that learning a meaningful internal representation can both accelerate generative training and enhance generation quality of the diffusion transformers. However, existing approaches necessitate to either introduce an additional and complex representation training framework or rely on a large-scale, pre-trained representation foundation model to provide representation guidance during the original generative training process. In this study, we posit that the unique discriminative process inherent to diffusion transformers enables them to offer such guidance without requiring external representation components. We therefore propose ***Self-Representation Alignment*** (**SRA**), a simple yet straightforward method that obtain representation guidance through a self-distillation manner. Specifically, SRA aligns the output latent representation of the diffusion transformer in earlier layer with higher noise to that in later layer with lower noise to progressively enhance the overall representation learning during only generative training process. Experimental results indicate that applying SRA to DiTs and SiTs yields consistent performance improvements. Moreover, SRA not only significantly outperforms approaches relying on auxiliary, complex representation training frameworks but also achieves performance comparable to methods that heavily dependent on powerful external representation priors.

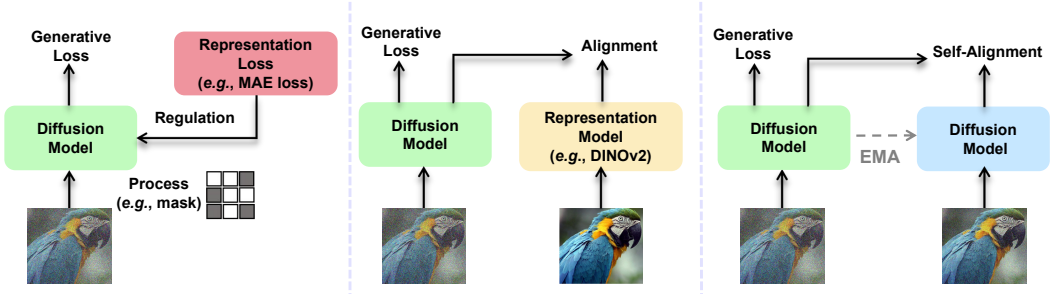
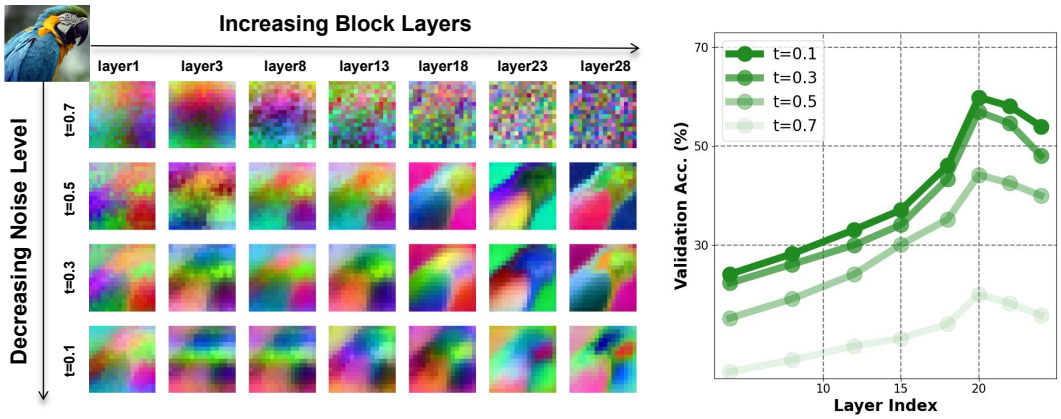


Figure 1: **Left:** Methods like MaskDiT [77] and SD-DiT [79] use an additional representation task to regulate diffsuion transformer. **Middle:** Methods like REPA [74] leverage an additional representation foundation model as guidance. **Right (our approach):** We do not use any representation component but only seek to provide representation guidance by diffusion transformer itself.

[†]Project lead.
[*]Corresponding author.



(a) PCA Visualization Result

(b) Linear Probing Result

Figure 2: We empirically investigate the representations in diffusion transformers across different blocks and timesteps with the original SiT-XL/2 checkpoint trained for 7M iterations. **Left:** Using PCA [1] to visualize the latent features in SiT, we observe that the features lead a process from coarse to fine when increasing block layers and decreasing noise level. **Right:** The similar trend can also be seen in the linear probing results on ImageNet. Investigation of DiT is provided in Appendix A, which leads to the similar results as SiT.

1 Introduction

Diffusion transformers [57, 52, 10] and vision transformers [18, 49, 68] have held the dominant positions in visual generation and representation because of their scalability during pre-training [5, 20, 56, 65] and generalization capacity for downstream tasks [38, 47, 35, 45].

Recently, many works [77, 79, 74, 39] have explored leveraging representation components of vision transformers for diffusion transformer's training, and have shown that learning a high-quality internal representation can not only speed-up the generative training progress, but also improve the generation quality. These works either utilize the training paradigm in representation learning (*e.g.*, MAE's [26], IBOT's [78]) shown in Figure 1a or leverage a large-scale pre-trained representation foundation model (*e.g.*, DINOv2 [56], CLIP [60]) shown in Figure 1b to give representation guidance for the diffusion transformer during the original generative training. However, the former method requires a complex training framework design, and the latter method relies on a powerful prior which is trained on massive amount of data with thousands of GPUs. Thus, we raise a significant but underexplored question: *Can we obtain such representation guidance only during the generative training without any representation component involved?*

Our observations: Different from the representation model that takes a clean image as input then output semantically-rich feature, diffusion model often takes a noise latent as input and obtain cleaner one step-by-step. In other words, the generative mechanism by which the diffusion model operates can be generally considered as a coarse to fine process. Hence, we hypothesise that the representations in diffusion models also follow this trend. To testify this, we perform an empirical analysis with recent diffusion transformers [52, 57]. As shown in Figure 2a, we first find out that the latent features in the diffusion transformer are progressively refined, moving from coarse to fine, as block layers increase and noise level decreased. Next, akin to the results in prior studies [74, 70], we observe that the diffusion transformer already learns meaningful discriminative representations as shown in Figure 2b. Meanwhile, although the accuracy drop off after reach a peak at about layer 20 because the model needs to shift away to focus on generating images with high-frequency details, the quality of the representations basically transfer from bad to good by increasing block layers and decreasing noise level. These results indicate that the diffusion transformer gets a roughly from coarse-to-fine discriminative process when only generative training is performed.

This distinctive trend motivates us to leverage the better representations in the diffusion transformer to guide the weaker ones in the original generative training, thereby enhancing the representation learning of the model without involving any additional representation component. However, this approach is not straightforward during training. As observed in previous studies [11, 25], directly using the output of the same model as a supervision signal may lead to shortcut learning [23], which can compromise performance or even cause training collapse.

Our approach: To overcome abovementioned hurdles, we present *Self-Representation Alignment* (SRA), a simple but effective technique built on recent diffusion transformer architectures [52, 57]. As shown in Figure 1c, SRA dose not need any representation component, in essence, it aligns the output latent representations in earlier layer conditioned with higher noise to that in later layer conditioned with lower noise to achieve self-representation guidance. Meanwhile, in order to make the training process more stable, we obtain the target features from another model which shares the same architecture with the trainable model but updates weight by weighted moving average (EMA)[1]. Furthermore, the student's output latent feature is first passed through the projection layers to conduct a slight nonlinear transformation for better representation extraction, and then aligned with the target feature output by the teacher. In a nutshell, our SRA can offer a flexible way to integrate representation guidance without representation component needs and architect modification.

Based on our analysis, we conduct comprehensive experiments to evaluate the effectiveness of SRA. After a series of component-wise analyses within the design space, we show that SRA brings significant performance improvements to both DiTs [57] and SiTs [52]. Moreover, our ablation study highlights the crucial role of internal representations in the success of SRA, which supports our central hypothesis: *diffusion transformers can provide representation guidance by themselves*.

In summary, our main contributions are as follows:

- We analyze the latent representations in the diffusion transformers and assume that diffusion transformers have the potential to provide representation guidance by themselves.
- We introduce SRA, a simple yet effective method that aligns the output latent representations of the diffusion transformers in earlier layer with higher noise to that in later layer with lower noise to achieve self-representation guidance.
- With our SRA, both DiTs and SiTs achieve sustained training speed acceleration and nontrivial generation performance improvement.

2 Method

2.1 Preliminary: Training Object of DiT and SiT

As our method is built upon the diffusion-based model (DiT) and flow-based model (SiT), to facilitate a more seamless introduction of SRA in the subsequent part, we now present a brief overview of the training object of these two types of models. We omit the class-condition here for simplicity because it can be easily added to all analogs without causing any changes to the mathematical equations. We leave more detailed mathematical descriptions of these types of models in Appendix B.

Diffusion-based models learn to transform Gaussian noise into data samples through a step-by-step denoising process. Given a pre-defined forward process that gradually adds noise, these models learn the reverse process to recover the original data.

For data point $\mathbf{x}_0$ from distribution $p(\mathbf{x})$, the forward noising process follows: $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_0, \beta_t^2\mathbf{I})$. The model learns to reverse this process using a neural network $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$ that predicts the noise added at each step. The network is trained using a simple mean squared error objective that measures how well it can predict the noise:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}\mathbf{x}_*, \boldsymbol{\epsilon}, t\Big[||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)||_2^2\Big]. \tag{1}$$

Different from diffusion-based model, flow-based model aims to learn a velocity field $\mathbf{v}_\theta(\mathbf{x}_t, t)$ that governs a probability flow ordinary differential equation (PF ODE). This PF ODE allows the model to sample data by starting from Gaussian noise and flowing towards the data distribution. This process is defined as:

$$\mathbf{x}_t = \alpha_t\mathbf{x}_* + \sigma_t\epsilon, \quad \alpha_0 = \sigma_T = 1, \alpha_T = \sigma_0 = 0, \tag{2}$$

where $\mathbf{x}_* \sim p(\mathbf{x})$ is the data, $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is Gaussian noise, and α_t and σ_t are decreasing and increasing functions of $t \in [0, T]$, respectively. The PF ODE is given by:

$$\dot{\mathbf{x}}_t = \mathbf{v}(\mathbf{x}_t, t), \tag{3}$$

[1] In the following part, we use term 'student' to represent the trainable model and 'teacher' to represent the EMA model in SRA for simplicity.

where the distribution of this ODE at time t matches the marginal distribution $p_t(\mathbf{x})$.

The velocity field can be expressed as:

$$\mathbf{v}(\mathbf{x}, t) = \mathbb{E}[\dot{\mathbf{x}}_t | \mathbf{x}_t = \mathbf{x}] = \dot{\alpha}_t \mathbb{E}[\mathbf{x}_* | \mathbf{x}_t = \mathbf{x}] + \dot{\sigma}_t \mathbb{E}[\epsilon | \mathbf{x}_t = \mathbf{x}]. \tag{4}$$

To learn the velocity field, the model $\mathbf{v}_\theta(\mathbf{x}_t, t)$ is trained to minimize the following loss function:

$$\mathcal{L}_{\text{velocity}} = \mathbb{E}_{\mathbf{x}_*, \boldsymbol{\epsilon}, t}\left[||\mathbf{v}_\theta(\mathbf{x}_t, t) - \dot{\alpha}_t \mathbf{x}_* - \dot{\sigma}_t \epsilon||^2\right]. \tag{5}$$

For the sake of simplicity, in the following part, we use generative loss ($\mathcal{L}_{\text{gen}}$) to uniformly represent the two generative training objects of DiT and SiT.

2.2 Self-Representation Alignment

Previous studies have demonstrated that learning good internal representation can both speed up diffusion transformer's training and improve the quality of its generated samples. In SRA, our insight is aligning students latent feature in earlier layer conditioned on higher noise with that in later layer conditioned on lower noise of teacher to conduct self-representation guidance without requiring any dedicated representation components. As depicted in Figure 3, the goal of this simple training framework is let the diffusion transformer not only predict noise-invariant, but also align with better visual representations from itself. This thereby provides a simple and resource-friendly way to enhance the representation learning in the diffusion transformer during generative training without the need to design complex representation regulation or introduce additional representation foundation models.

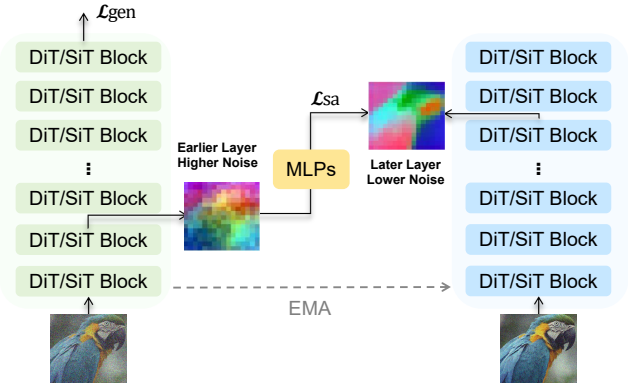


Figure 3: **Overall framework of SRA.** Our SRA explicitly aligns the diffusion transformer's representations in earlier layer with higher noise with the representations in later layer with lower noise to conduct self-representation alignment.

Formally, let f be the trainable student model and f_* be the teacher model. Considering input noise latent, timestep, and condition to be $\mathbf{x}_t$, t, and c. Then, we can obtain the student encoder latent output $\mathbf{y} = f^m(\mathbf{x}_t, t, c) \in \mathbb{R}^{B \times N \times D}$, where $B, N, D > 0$ are the batchsize, number of patches and the embedding dimension for f, and m denotes the output from the m^{th} layer in f. Similarly, the output of the teacher can be expressed as $\mathbf{y}_* = f_*^n(\mathbf{x}_{t-k}, t-k, c) \in \mathbb{R}^{B \times N \times D}$. In our SRA, we set $m \leq n$, $k \geq 0$, and $0 \leq (t-k) < t_{max}$[2]. Hence, we can conduct self-representation alignment using the teacher's output $\mathbf{y}_*$ and the student's output variant $h_\phi(\mathbf{y}) \in \mathbb{R}^{B \times N \times D}$, where $h_\phi(\mathbf{y})$ is a projection of the student encoder output $\mathbf{y}$ that through a lightweight trainable MLPs head h_ϕ. Notably, this projection head can be discarded optionally after training, which enables SRA to provide guidance without altering any architecture in diffusion transformers.

In particular, SRA attains self-alignment by minimizing the patch-wise distance between the teacher's output ($\mathbf{y}_*$) and the student's output variant ($h_\phi(\mathbf{y})$):

$$\mathcal{L}_{\text{sa}}(\theta_s, \phi) = \mathbb{E}_{\mathbf{x}_t, t, c}\left[\frac{1}{N}\sum_{i=1}^{N} \text{dist}(\mathbf{y}_*^{[i]}, h_\phi(\mathbf{y}^{[i]}))\right], \tag{6}$$

where $[i]$ is a patch index, $\text{dist}(\cdot, \cdot)$ is a pre-defined distance calculation function, and θ_s, ϕ is the parameters of student diffusion transformer and the projection head.

Finally, we add this objective in the original diffusion-based objective described in Section 2.1 and Appendix B for joint learning:

$$\mathcal{L} = \mathcal{L}_{\text{gen}} + \lambda \mathcal{L}_{\text{sa}}, \tag{7}$$

[2]For SiTs, $t_{max} = 1$ and for DiTs, $t_{max} = 1000$. In practice, we truncate $(t-k)$ to 0 if it is less than 0.

where $\lambda > 0$ is a hyperparameter that controls the trade-off between generation object and self-representation alignment object.

2.3 EMA Teacher Network

In SRA, we do not need a off-the-shelf teacher to give the prior guidance. Meanwhile, using the output of the same model as the target to conduct supervision would cause training instability [11, 25]. Thus, we follow some works in self-supervised learning (SSL) [78, 8, 27] that build the teacher from past iterations of the student network using an exponential moving average (EMA) on the student weights. In specific, the update role of EMA is $\theta_t = \alpha\theta_t + (1-\alpha)\theta_s$, where $\alpha \in [0, 1)$ is the momentum coefficient. We study different values of α in Section 3.2 and show that $\alpha = 0.9999$ unchanged works surprisingly well in our framework. Moreover, we do not use other operations like clustering constraints [6, 7], batch normalizations [25, 61], and centering [8, 78] in SSL, because we find that the training progress is already stable enough without applying these tricks.

3 Experiment

In this section, we seek to validate the effect of SRA through extensive experiments. In a specific, we mainly focus on answering the following questions:

- How each design choice and component in SRA influence the performance? (Table 1)
- Dose SRA work on different baselines across different model size? (Figure 4)
- Can SRA show comparable or superior performance against other methods that leverage either representation training paradigm or representation foundation model? (Table 2)
- Does SRA genuinely enhance the representation capacity of the baseline model, and is the generation capability indeed strongly correlated with the representation guidance? (Figure 6)

3.1 Experimental Setup

Implementation details. Unless otherwise specified, the training details are strictly followed the setup in DiT [57] and SiT [52], including AdamW [51] with a constant learning rate of 1e-4, no weight decay, batchsize of 256, using the Stable Diffusion VAE [62] to extract the latent, and etc. We use ImageNet [16], where each image is preprocessed to the resolution of 256×256 (denoted as ImageNet 256×256), and follow ADM [17] for other data preprocessing protocols. For model configurations, we use the B/2, L/2, and XL/2 architectures introduced in the DiT and SiT papers, which process inputs with a patch size of 2. Additional experimental details, hyperparameter settings, linear probing details, and computing resources, are provided in Appendix C.

Evaluation. We report Fréchet inception distance (FID [28]), sFID [54], inception score (IS [64]), precision (Pre.) and recall (Rec.) [41] . To ensure a fair comparison with previous methods, we also use the ADMs TensorFlow evaluation suite [17]with 50K samples and the same reference statistics. More details of each metric are provided in Appendix D.

Sampler. For DiT [57], we use DDPM sampler and set the number of function evaluations (NFE) as 250 by default. For SiT [52], we use the SDE Euler-Maruyama sampler (for SDE with $w_t = \sigma_t$) and set the NEF as 250 by default. The settings also align with those used in DiT and SiT papers.

Baselines. We use several recent diffusion-based generation methods as baselines. Specifically, we consider the following three types of approaches: (a) *Pixel diffusion*: ADM [17], VDM++ [37], Simple diffusion [32], CDM [30], (b) *Latent diffusion with U-Net*: LDM [63], and (c) *Latent diffusion with transformers*: DiT [57], SiT [52], SD-DiT [79], MaskDiT [77], TREAD [39], REPA [74], and MAETok [9]. Here in diffusion transformers family, we choose to compare with DiT/SiT and their modifications with representation components involved but do not compare with works aiming at designing advanced architecture like lightningDiT [73] and DDT [67]. Detailed descriptions of each baseline method are provided in Appendix E.

3.2 Component-Wise Analysis

Below, we provide a detailed analysis of the impact of each component. To save time, we use SiT-B/2 and train with SRA for 400K iterations for evaluation. Results are shown in Table 1.

Table 1: **Component-wise analysis** on ImageNet 256×256 without classifier-free guidance (CFG). ↓ and ↑ indicate whether lower or higher values are better, respectively. $m \rightarrow n$ denotes aligning features from m^{th} layer of the student with that from n^{th} layer of the teacher. $0 \sim k_{max}$ denotes time Interval k is a random value in $[0, k_{max})$. PH. denotes whether to use projection head.

Block Layers	Time Interval	Objective	EMA teacher	PH.	FID↓	IS↑
SiT-B/2 Baseline [52]					33.02	43.71
$6 \rightarrow 10$	$0 \sim 0.2$	smooth-ℓ_1	Gen's mom.	✓	34.85	40.58
$4 \rightarrow 8$	$0 \sim 0.2$	smooth-ℓ_1	Gen's mom.	✓	30.00	47.78
$4 \rightarrow 10$	$0 \sim 0.2$	smooth-ℓ_1	Gen's mom.	✓	30.65	46.69
$4 \rightarrow 12$	$0 \sim 0.2$	smooth-ℓ_1	Gen's mom.	✓	33.19	43.30
$2 \rightarrow 6$	$0 \sim 0.2$	smooth-ℓ_1	Gen's mom.	✓	32.14	46.36
$2 \rightarrow 8$	$0 \sim 0.2$	smooth-ℓ_1	Gen's mom.	✓	29.31	50.13
$3 \rightarrow 8$	$0 \sim 0.2$	smooth-ℓ_1	Gen's mom.	✓	**29.10**	**50.20**
$3 \rightarrow 3$	$0 \sim 0.2$	smooth-ℓ_1	Gen's mom.	✓	37.08	41.54
$3 \rightarrow 8$	0	smooth-ℓ_1	Gen's mom.	✓	31.07	47.32
$3 \rightarrow 8$	0.1	smooth-ℓ_1	Gen's mom.	✓	29.55	49.01
$3 \rightarrow 8$	0.2	smooth-ℓ_1	Gen's mom.	✓	30.70	47.72
$3 \rightarrow 8$	$0 \sim 0.1$	smooth-ℓ_1	Gen's mom.	✓	29.38	49.32
$3 \rightarrow 8$	$0 \sim 0.2$	smooth-ℓ_1	Gen's mom.	✓	**29.10**	**50.20**
$3 \rightarrow 8$	$0 \sim 0.3$	smooth-ℓ_1	Gen's mom.	✓	29.15	50.01
$3 \rightarrow 8$	$0 \sim 0.2$	smooth-ℓ_1	Gen's mom.	✓	29.10	**50.20**
$3 \rightarrow 8$	$0 \sim 0.2$	ℓ_1	Gen's mom.	✓	**29.06**	50.08
$3 \rightarrow 8$	$0 \sim 0.2$	ℓ_2	Gen's mom.	✓	29.54	49.25
$3 \rightarrow 8$	$0 \sim 0.2$	smooth-ℓ_1	Student copy	✓	35.71	42.18
$3 \rightarrow 8$	$0 \sim 0.2$	smooth-ℓ_1	SSL's mom.	✓	33.17	44.96
$3 \rightarrow 8$	$0 \sim 0.2$	smooth-ℓ_1	Gen's mom.	✓	**29.10**	**50.20**
$3 \rightarrow 8$	$0 \sim 0.2$	smooth-ℓ_1	Gen's mom.	×	34.23	41.07
$3 \rightarrow 8$	$0 \sim 0.2$	smooth-ℓ_1	Gen's mom.	✓	**29.10**	**50.20**

Block layers for alignment. We begin by analyzing the effect of using different blocks of student and teacher for alignment. We observe that using the teacher's last but not least few layers (*e.g.*, 8) to regulate the student's first layers (*e.g.*, 3) leads to optimal performance. We assume that the first few layers need more guidance so they can catch semantically meaningful representation for subsequent generation. Meanwhile, there is a strong correlation between the quality of the representations of the teacher's layers and the performance of the corresponding aligned student (we give the quantitative results and analysis in latter Section 3.4). Based on these results, we set alignment layers as $3 \rightarrow 8$, $6 \rightarrow 16$, and $8 \rightarrow 20$ for B, L, and XL models respectively as default[3]

Time interval for alignment. We then study the time interval (meaning the same as k in Section 2.2) used for alignment. Here we study fixed and dynamic interval. Notably, we find that using teacher's features input with lower noise than student's leads to performance enhancement and interval value with 0.1 or the mean with 0.1 is optimal. We hypothesize that this is because lower noise levels can offer better representation guidance; but an excessively large time interval can hinder the model's learning process, causing it to only focus on optimizing alignment loss at the expense of neglecting the generative aspects. As dynamic interval shows slightly better performance, we apply time interval as $0 \sim 0.2$ in our feature experiments[4].

Objective for alignment. We next compare three simple regression training objectives for alignment, including smooth-ℓ_1, ℓ_1, and ℓ_2. Empirically, we find that all three objectives can lead to good performance and are stable during training. We adopt smooth-ℓ_1 by default in future experiments.

Teacher network for alignment. In other generative learning studies, the EMA model is often employed solely for evaluation. However, as we need it to provide guidance during training, we study different updating methods here. Here, we investigate several different strategies to build the teacher. First, we find that using teacher copied from student would impair the performance, which

[3]For DiTs, we set alignment layers as $3 \rightarrow 7$, $6 \rightarrow 14$, and $8 \rightarrow 16$ by default because DiT's discriminative behavior across layers is sightly different from SiT's (see Figure 2 and Figure 7).

[4]For DiTs, we set time interval as $\lfloor 0 \sim 200 \rfloor$ by default since DiTs adopt the linear variance schedule with t where $t \in \{0, 1, 2, ..., 999\} \cap \mathbb{Z}$.

testifies our argument in Section 1. Next, we consider using strategy used in self-supervised learning works [8, 78, 25] that updates momentum coefficient from 0.996 to 1 during training. However, in our framework, this does not work well. Finally, we use momentum coefficient of 0.9999 unchanged which is commonly used in other generative learning works [17, 57, 52] and find it is also suitable for our framework. Thus, we set momentum coefficient as 0.9999 in future experiments.

Effect of projection head for alignment. We finally examine the effect of the projection head for alignment. Surprisingly, We observe that using this simple head to post-possess the student's output is much better than directly using it to alignment. We hypothesize this slight operation enables the model to learn more effective hidden representations for subsequent projection head to conduct transformation for final alignment, rather than explicitly aligning the entire latent feature that could potentially disrupt the original generation field that each layer and timestep is responsible for [76, 75, 22]. Hence, we keep using the projection head in future experiments.

3.3 System-Level Comparison

Based on the analysis, we perform a system-level comparison between recent state-of-the-art diffusion model approaches and diffusion transformers with SRA.

First, we compare the FID values between vanilla DiT or SiT and the same models trained with SRA. As shown in Figure. 4, diffusion transformers trained with SRA demonstrate substantial improvements in performance at each training step across different types , as well as various sizes. Moreover, similar to the observation in some SSL works [56, 21], we notice that the effect of SRA in larger size model is more significant, which probably because larger model tends to provide richer guidance. It is also worth noting that the benefits of SRA do not saturate even when the models have already achieved a low FID score. We assume that this is likely due to the teacher's constantly improving capacity throughout training, which allows it to provide better and better representation guidance for the student when the training goes on.

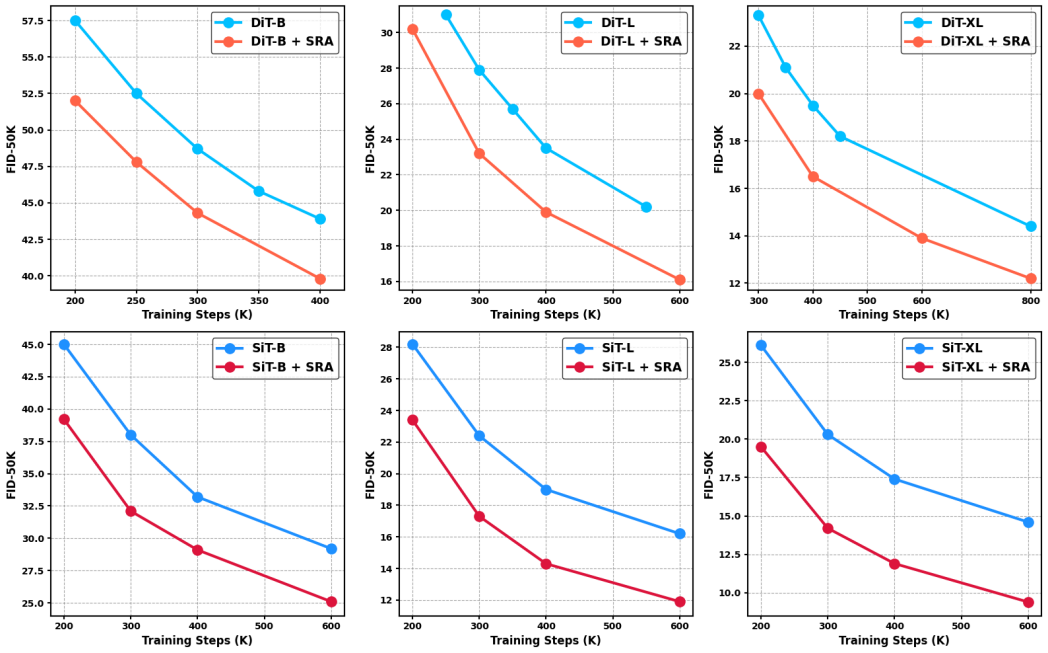


Figure 4: **FID comparisons with vanilla DiTs and SiTs across different model size** on ImageNet 256×256 without classifier-free guidance (CFG).

Finally, we provide a quantitative comparison between SiT-XL with SRA and other recent dif- fusion model methods using classifier-free guidance [31]. Our method already outperforms the original SiT-XL model with 1000 fewer epochs and it is further improved with longer training. At 800 epochs, SiT-XL with SRA achieves FID of 1.58 and IS of 311.4. It is worth noting that this result is far superior than methods (*e.g.*, MaskDiT [77]) that rely on auxiliary and complex representation training frameworks, and is comparable with methods (*e.g.*, REPA [74]) that depend heavily

Table 2: **System-level comparison** on ImageNet 256×256 with Classifier-free Guidance (CFG). The **best** and second-best results on each metric are highlighted in bold and underlined.

Model	Epochs	Tokenizer	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
Pixel diffusion							
ADM-U	400	-	3.94	6.14	186.7	0.82	0.52
VDM++	560	-	2.40	-	225.3	-	-
Simple diffusion	800	-	2.77	-	211.8	-	-
CDM	2160	-	4.88	-	158.7	-	-
Latent diffusion, U-Net							
LDM-4	200	LDM-VAE	3.60	-	247.7	**0.87**	0.48
Latent diffusion, Transformer							
DiT-XL/2	1400	SD-VAE	2.27	4.60	278.2	0.83	0.57
SiT-XL/2	1400	SD-VAE	2.06	**4.50**	270.3	0.82	0.59
SD-DiT	480	SD-VAE	3.23	-	-	-	-
MaskDiT	1600	SD-VAE	2.28	5.67	276.6	0.80	0.61
DiT + TREAD	740	SD-VAE	1.69	4.73	292.7	0.81	0.63
SiT + REPA	800	SD-VAE	**1.42**	4.70	305.7	0.80	**0.65**
SiT + MAETok	800	MAE-Tok	1.67	-	311.2	-	-
SiT + SRA (ours)	**400**	SD-VAE	1.85	**4.50**	297.2	0.82	0.61
SiT + SRA (ours)	**800**	SD-VAE	1.58	4.65	**311.4**	0.80	0.63

Figure 5: **Selected samples** on ImageNet 256×256 from the SiT-XL + SRA. We use classifier-free guidance with $w = 4.0$. More **uncurated samples** are provided in Appendix J.

on powerful external representation priors. Moreover, due to the progressively higher-quality guidance teacher provide throughout training, we find that our method is much less likely to encounter saturation compared with REPA (we provide detailed results and analysis in Appendix F).

3.4 Ablation Study

In this part, we aim at testing whether representation truly matters in SRA. We give the answer by conducting following experiments.

Enhanced representation capacity with SRA. We first compare the representation capacity of vanilla SiT and SiT trained with SRA. As shown in Figure 6a and Figure 6b, SRA consistently improve the quality of latent representation in the diffusion transformer, as indicated by better linear probing results across different blocks and timesteps.

Tight coupling between generation quality and representation guidance in SRA. We then investigate the correlation between generation performance and the representation guidance (detailed experimental setup can be find in Appendix G) in SRA. Figure 6c reveals a strong correlation between linear probing accuracy and FID scores as the teacher network layers for alignment are varied.

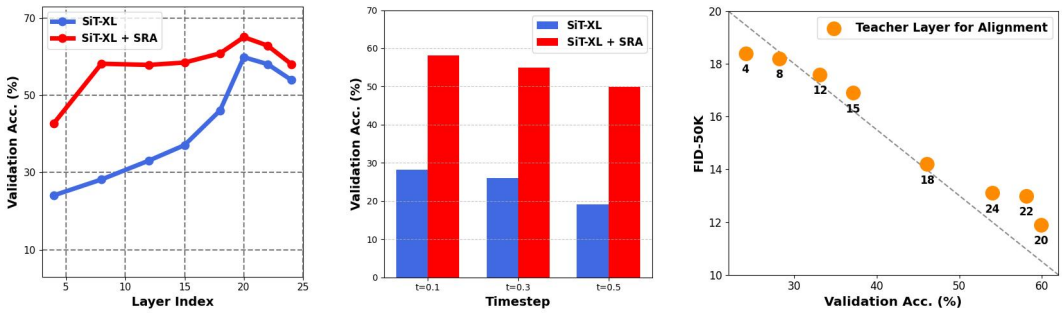


Figure 6: We empirically investigate the effect of representations in SRA. **Left and Middle:** Linear probing result of vanilla SiT-XL trained for 1400 epochs and SiT-XL + SRA trained for 800 epochs. **Right:** Linear probing vs. FID plot of SiT-XL + SRA with different teacher's output layers for alignment (similar plot of DiT + SRA is provided in Appendix H).

This finding underscores that the model's generative capabilities are indeed closely tied to the effectiveness of the self-representation guidance mechanism.

4 Related Work

We discuss with the most relevant studies here and provide a more discussion in Appendix I.

Representations guidance for diffusion transformer's pre-training. Many recent works have attempted to introducing representation guidance in diffusion transformer's training. MaskDiT [77] and SD-DiT [79] add MAE's [26] and IBOT's [78] training paradigm into the original DiT's training progress. TREAD [39] design a token routing strategy with MAE loss to speed-up diffusion model's training. REPA [74] utilize a large-scale data pre-trained representation model to regulate diffusion model's latent feature. VA-VAE [73] and MAETok [9] align the latent distribution of the tokenizer with additional representation foundation model and show this alignment is beneficial to resulting diffusion models. Different from these work, we aim at looking for representation guidance in the diffusion model itself and it's own training paradigm.

EMA model as teacher guidance. Different from traditional knowledge distillation [24] that uses a stronger pre-trained model as teacher, using EMA model [33] as teacher can be seen as self-distillation because the weight of the teacher model is obtained by weighted moving average of the parameters of the student model. This method needs a feasible pretext task to succeed. For example, MoCo [27, 13] sets the EMA teacher as momentum encoder and use contrastive learning to guidance the student model; DINO [8, 56] feeds two views of images to EMA teacher and trainable student then forces the student's output distribution to be close to teacher's. Our work also shares some similarities, where we set aligning student model's latent feature in earlier layer and higher noise with that in later layer and lower noise of the EMA teacher as our pretext task for training.

5 Conclusion and Discussion

In this study, we show that diffusion transformers can provide representation guidance by themselves to boost generation performance with our proposed SRA, which aligns the output latent representation of the diffusion transformer in earlier layer with higher noise to that in later layer with lower noise to progressively enhance the overall representation learning during only generative process.

Meanwhile, a few open questions are worth discussing. First, we have observed that the effect of SRA becomes more pronounced when scaling model size. However, owing to the limitation of computational resources, we are unable to further scale the model size and data. Investigating the scalability of SRA in more complex and resource-intensive scenarios (*e.g.*, text-to-image) will be an exciting future direction. Next, similar to other related works [74, 77, 9, 79], our method is also experiment-driven. Exploring theoretical insights into why learning a good representation is beneficial to generation will also be an exciting future direction.

Considering the simplicity and effectiveness of SRA, we believe it will facilitate more future researches that focus on unveiling the intertwined nature of visual representation and generation.

References

[1] Abdi, H., Williams, L.J.: Principal component analysis. Wiley interdisciplinary reviews: computational statistics **2**(4), 433–459 (2010)

[2] Albergo, M.S., Vanden-Eijnden, E.: Building normalizing flows with stochastic interpolants. In: International Conference on Learning Representations (2023)

[3] Bao, F., Nie, S., Xue, K., Cao, Y., Li, C., Su, H., Zhu, J.: All are worth words: A vit backbone for diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 22669–22679 (2023)

[4] Bao, H., Dong, L., Piao, S., Wei, F.: BEiT: BERT pre-training of image transformers. In: International Conference on Learning Representations (2022)

[5] Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., Ramesh, A.: Video generation models as world simulators. OpenAI Blog (2024)

[6] Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: Proceedings of the European conference on computer vision (ECCV). pp. 132–149 (2018)

[7] Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. Advances in neural information processing systems **33**, 9912–9924 (2020)

[8] Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the International Conference on Computer Vision (2021)

[9] Chen, H., Han, Y., Chen, F., Li, X., Wang, Y., Wang, J., Wang, Z., Liu, Z., Zou, D., Raj, B.: Masked autoencoders are effective tokenizers for diffusion models. arXiv preprint arXiv:2502.03444 (2025)

[10] Chen, J., Yu, J., Ge, C., Yao, L., Xie, E., Wu, Y., Wang, Z., Kwok, J., Luo, P., Lu, H., Li, Z.: Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In: International Conference on Learning Representations (2024)

[11] Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15750–15758 (2021)

[12] Chen, X., Liu, Z., Xie, S., He, K.: Deconstructing denoising diffusion models for self-supervised learning. arXiv preprint arXiv:2401.14404 (2024)

[13] Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9640–9649 (2021)

[14] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). vol. 1, pp. 886–893. Ieee (2005)

[15] Dao, T.: Flashattention-2: Faster attention with better parallelism and work partitioning. arXiv preprint arXiv:2307.08691 (2023)

[16] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)

[17] Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in neural information processing systems **34**, 8780–8794 (2021)

[18] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale (2021)

[19] Elfwing, S., Uchibe, E., Doya, K.: Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. Neural networks **107**, 3–11 (2018)

[20] Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-first international conference on machine learning (2024)

[21] Fan, D., Tong, S., Zhu, J., Sinha, K., Liu, Z., Chen, X., Rabbat, M., Ballas, N., LeCun, Y., Bar, A., Xie, S.: Scaling language-free visual representation learning. arXiv preprint arXiv:2504.01017 (2025)

[22] Frenkel, Y., Vinker, Y., Shamir, A., Cohen-Or, D.: Implicit style-content separation using b-lora. In: European Conference on Computer Vision. pp. 181–198. Springer (2024)

[23] Geirhos, R., Jacobsen, J.H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., Wichmann, F.A.: Shortcut learning in deep neural networks. Nature Machine Intelligence **2**(11), 665–673 (2020)

[24] Gou, J., Yu, B., Maybank, S.J., Tao, D.: Knowledge distillation: A survey. International Journal of Computer Vision **129**(6), 1789–1819 (2021)

[25] Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Advances in neural information processing systems **33**, 21271–21284 (2020)

[26] He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16000–16009 (2022)

[27] He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9729–9738 (2020)

[28] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)

[29] Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)

[30] Ho, J., Saharia, C., Chan, W., Fleet, D.J., Norouzi, M., Salimans, T.: Cascaded diffusion models for high fidelity image generation. Journal of Machine Learning Research **23**(47), 1–33 (2022)

[31] Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)

[32] Hoogeboom, E., Heek, J., Salimans, T.: simple diffusion: End-to-end diffusion for high resolution images. In: International Conference on Machine Learning. pp. 13213–13232. PMLR (2023)

[33] Hunter, J.S.: The exponentially weighted moving average. Journal of quality technology **18**(4), 203–210 (1986)

[34] Jiang, D., Wang, H., Zhang, L., Wei, W., Dai, G., Wang, M., Wang, J., Zhang, Y.: Unbiased general annotated dataset generation. arXiv preprint arXiv:2412.10831 (2024)

[35] Jiang, L., Yan, Q., Jia, Y., Liu, Z., Kang, H., Lu, X.: InfiniteYou: Flexible photo recrafting while preserving your identity. arXiv preprint **arXiv:2503.16418** (2025)

[36] Karras, T., Aittala, M., Lehtinen, J., Hellsten, J., Aila, T., Laine, S.: Analyzing and improving the training dynamics of diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 24174–24184 (2024)

[37] Kingma, D., Gao, R.: Understanding diffusion objectives as the elbo with simple data augmentation. Advances in Neural Information Processing Systems **36**, 65484–65516 (2023)

[38] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4015–4026 (2023)

[39] Krause, F., Phan, T., Hu, V.T., Ommer, B.: Tread: Token routing for efficient architecture-agnostic diffusion training. arXiv preprint arXiv:2501.04765 (2025)

[40] Kynkäänniemi, T., Aittala, M., Karras, T., Laine, S., Aila, T., Lehtinen, J.: Applying guidance in a limited interval improves sample and distribution quality in diffusion models. Advances in Neural Information Processing Systems **37**, 122458–122483 (2025)

[41] Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., Aila, T.: Improved precision and recall metric for assessing generative models. Advances in neural information processing systems **32** (2019)

[42] Labs, B.F.: Flux. https://github.com/black-forest-labs/flux (2024)

[43] Li, D., Ling, H., Kar, A., Acuna, D., Kim, S.W., Kreis, K., Torralba, A., Fidler, S.: Dreamteacher: Pretraining image backbones with deep generative models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16698–16708 (2023)

[44] Li, T., Katabi, D., He, K.: Return of unconditional generation: A self-supervised representation generation method. Advances in Neural Information Processing Systems **37**, 125441–125468 (2024)

[45] Lin, W., Wei, X., Zhang, R., Zhuo, L., Zhao, S., Huang, S., Xie, J., Qiao, Y., Gao, P., Li, H.: Pixwizard: Versatile image-to-image visual assistant with open-language instructions. In: International Conference on Learning Representations (2025)

[46] Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022)

[47] Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In: European Conference on Computer Vision. pp. 38–55. Springer (2024)

[48] Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. In: International Conference on Learning Representations (2023)

[49] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021)

[50] Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11976–11986 (2022)

[51] Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)

[52] Ma, N., Goldstein, M., Albergo, M.S., Boffi, N.M., Vanden-Eijnden, E., Xie, S.: Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In: European Conference on Computer Vision. pp. 23–40. Springer (2024)

[53] Mukhopadhyay, S., Gwilliam, M., Agarwal, V., Padmanabhan, N., Swaminathan, A., Hegde, S., Zhou, T., Shrivastava, A.: Diffusion models beat gans on image classification. arXiv preprint arXiv:2307.08702 (2023)

[54] Nash, C., Menick, J., Dieleman, S., Battaglia, P.W.: Generating images with sparse representations. arXiv preprint arXiv:2103.03841 (2021)

[55] Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: International conference on machine learning. pp. 8162–8171. PMLR (2021)

[56] Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)

[57] Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4195–4205 (2023)

[58] Preechakul, K., Chatthee, N., Wizadwongsa, S., Suwajanakorn, S.: Diffusion autoencoders: Toward a meaningful and decodable representation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10619–10629 (2022)

[59] Qin, Q., Zhuo, L., Xin, Y., Du, R., Li, Z., Fu, B., Lu, Y., Li, X., Liu, D., Zhu, X., et al.: Lumina-image 2.0: A unified and efficient image generative framework. arXiv preprint arXiv:2503.21758 (2025)

[60] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)

[61] Richemond, P.H., Grill, J.B., Altché, F., Tallec, C., Strub, F., Brock, A., Smith, S., De, S., Pascanu, R., Piot, B., et al.: Byol works even without batch statistics. arXiv preprint arXiv:2010.10241 (2020)

[62] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)

[63] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)

[64] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. Advances in neural information processing systems **29** (2016)

[65] Sun, Q., Wang, J., Yu, Q., Cui, Y., Zhang, F., Zhang, X., Wang, X.: Eva-clip-18b: Scaling clip to 18 billion parameters. arXiv preprint arXiv:2402.04252 (2024)

[66] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)

[67] Wang, S., Tian, Z., Huang, W., Wang, L.: Ddt: Decoupled diffusion transformer. arXiv preprint arXiv:2504.05741 (2025)

[68] Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 568–578 (2021)

[69] WanTeam: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)

[70] Xiang, W., Yang, H., Huang, D., Wang, Y.: Denoising diffusion autoencoders are unified self-supervised learners. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15802–15812 (2023)

[71] Yang, X., Wang, X.: Diffusion model as representation learner. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18938–18949 (2023)

[72] Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al.: Cogvideox: Text-to-video diffusion models with an expert transformer. arXiv preprint arXiv:2408.06072 (2024)

[73] Yao, J., Wang, X.: Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. arXiv preprint arXiv:2501.01423 (2025)

[74] Yu, S., Kwak, S., Jang, H., Jeong, J., Huang, J., Shin, J., Xie, S.: Representation alignment for generation: Training diffusion transformers is easier than you think. In: International Conference on Learning Representations (2025)

[75] Zhang, Y., Dong, W., Tang, F., Huang, N., Huang, H., Ma, C., Lee, T.Y., Deussen, O., Xu, C.: Prospect: Prompt spectrum for attribute-aware personalization of diffusion models. ACM Transactions on Graphics (TOG) **42**(6), 1–14 (2023)

[76] Zhang, Z., Zhang, Q., Lin, H., Xing, W., Mo, J., Huang, S., Xie, J., Li, G., Luan, J., Zhao, L., et al.: Towards highly realistic artistic style transfer via stable diffusion with step-aware and layer-aware prompt. arXiv preprint arXiv:2404.11474 (2024)

[77] Zheng, H., Nie, W., Vahdat, A., Anandkumar, A.: Fast training of diffusion models with masked transformers. arXiv preprint arXiv:2306.09305 (2023)

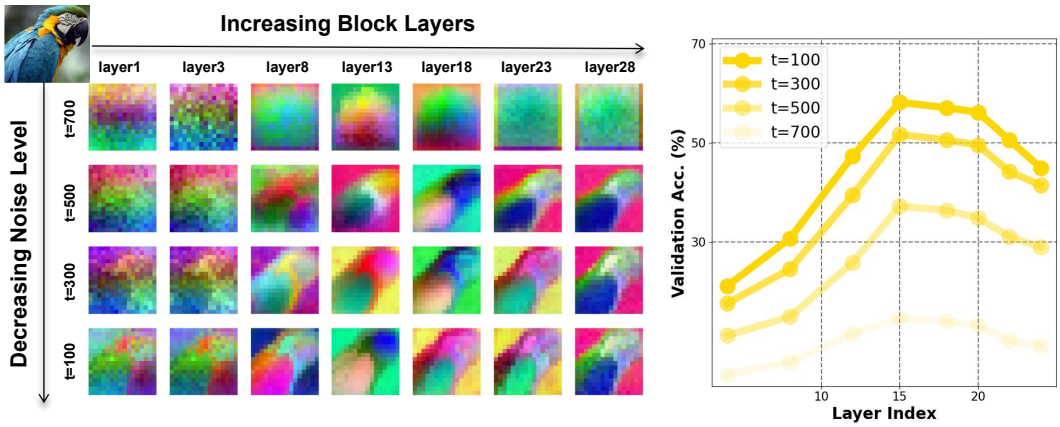
[78] Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., Kong, T.: ibot: Image bert pre-training with online tokenizer. In: International Conference on Learning Representations (2022)

[79] Zhu, R., Pan, Y., Li, Y., Yao, T., Sun, Z., Mei, T., Chen, C.W.: Sd-dit: Unleashing the power of self-supervised discrimination in diffusion transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8435–8445 (2024)

Appendix

A Investigation of Representations in DiT

We also perform a similar analysis with DiT like those have done in Figure 2a (PCA visualization) and Figure 2b (linear probing), the results are showed in Figure 7. In short, we also observe that the representations in DiT basically lead a process from coarse to fine when increasing block layers and decreasing noise level as SiT's.



(a) PCA Visualization Result

(b) Linear Probing Result

Figure 7: We also empirically investigate the representations in diffusion transformers across different blocks and timesteps with the original DiT-XL/2 checkpoint trained for 7M iterations. Similar to SiT, the latent representations in DiT basically follow the coarse to fine process, as block layers increase and noise level decreased.

B Descriptions for Two Types of Baseline Models

In this paper, we use DiT and SiT as our baseline models. We now provide an overview of two types of generative models that are variants of denoising autoencoders and are used to learn the target distribution. Specifically, we discuss Denoising Diffusion Probabilistic Models (DDPMs) like DiT in Section B.1 and Stochastic Interpolants Models like SiT in Section B.2.

B.1 Denoising Diffusion Probabilistic Models (DiT)

Denoise diffusion-based models [29, 55] aim to model the target distribution ($p(\mathbf{x})$) by learning a gradual denoising process that transforms a Gaussian distribution ($\mathcal{N}(\mathbf{0}, \mathbf{I})$) into $p(\mathbf{x})$. Formally, diffusion models learn a reverse process ($p(\mathbf{x}_{t-1}|\mathbf{x}_t)$) corresponding to a pre-defined forward process ($q(\mathbf{x}_t|\mathbf{x}_0)$), which incrementally adds Gaussian noise to the data starting from $p(\mathbf{x})$ over a sequence of time steps ($t \in 1, \ldots, T$), with $T > 0$ fixed.

For a given $\mathbf{x}_0 \sim p(\mathbf{x})$, the forward process ($q(\mathbf{x}_t|\mathbf{x}_{t-1})$) is defined as: [$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_0, \beta_t^2\mathbf{I})$,] where $\beta_t \in (0, 1)$ are pre-defined small hyperparameters. DDPM [29] formalizes the reverse process ($p(\mathbf{x}_{t-1}|\mathbf{x}_t)$) as:

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}\Big(\mathbf{x}_{t-1}; \frac{1}{\sqrt{\alpha_t}}\big(\mathbf{x}_t - \frac{\sigma_t^2}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\big), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\Big), \tag{8}$$

where $\alpha_t = 1 - \beta_t$, $\bar{\alpha}t = \prod i = 1^t \alpha_i$, and $\boldsymbol{\epsilon}\boldsymbol{\theta}(\mathbf{x}_t, t)$ is parameterized by a neural network.

The model is trained using a simple denoising autoencoder objective:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{\mathbf{x}_*, \boldsymbol{\epsilon}, t}\Big[||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)||_2^2\Big]. \tag{9}$$

For the covariance term ($\mathbf{\Sigma\theta}(\mathbf{x}_t, t)$), DDPM[29] demonstrated that setting it as ($\sigma_t^2\mathbf{I}$) with ($\beta_t = \sigma_t^2$) is sufficient. Subsequently, Improved-DDPM[55] showed that performance can be enhanced by jointly learning ($\mathbf{\Sigma\theta}(\mathbf{x}_t, t)$) along with ($\boldsymbol{\epsilon\theta}(\mathbf{x}_t, t)$) in a dimension-wise manner using the following objective:

$$\mathcal{L}_{\text{vlb}} = \exp(v \log \beta_t + (1 - v) \log \tilde{\beta}_t), \tag{10}$$

where v is a component per model output dimension, and $\tilde{\beta}t = \frac{1-\bar{\alpha}t-1}{1-\bar{\alpha}_t}\beta_t$.

With a sufficiently large T and an appropriate scheduling of β_t, the distribution $p(\mathbf{x}_T)$ approaches an isotropic Gaussian distribution. Thus, sampling is achieved by starting from random Gaussian noise and iteratively applying the reverse process ($p(\mathbf{x}_{t-1}|\mathbf{x}_t)$) to recover a data sample $\mathbf{x}_0$ [29].

B.2 Stochastic Interpolants Models (SiT)

Unlike DDPMs, flow-based models [48, 46] describe a continuous time-dependent process involving data ($\mathbf{x}_* \sim p(\mathbf{x})$) and Gaussian noise ($\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$) over $t \in [0, 1]$:

$$\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \epsilon, \quad \alpha_0 = \sigma_1 = 1, \ \alpha_1 = \sigma_0 = 0, \tag{11}$$

with α_t and σ_t being decreasing and increasing functions of t, respectively. The process is governed by a probability flow ordinary differential equation (PF ODE):

$$\dot{\mathbf{x}}_t = \mathbf{v}(\mathbf{x}_t, t), \tag{12}$$

where the distribution of the ODE at time t matches the marginal distribution $p_t(\mathbf{x})$.

The velocity $\mathbf{v}(\mathbf{x}, t)$ is expressed as:

$$\mathbf{v}(\mathbf{x}, t) = \mathbb{E}[\dot{\mathbf{x}}_t|\mathbf{x}_t = \mathbf{x}] = \dot{\alpha}_t \mathbb{E}[\mathbf{x}_*|\mathbf{x}_t = \mathbf{x}] + \dot{\sigma}_t \mathbb{E}[\epsilon|\mathbf{x}_t = \mathbf{x}], \tag{13}$$

and can be approximated by a model $\mathbf{v\theta}(\mathbf{x}_t, t)$ trained to minimize the objective:

$$\mathcal{L}_{\text{velocity}} = \mathbb{E}_{\mathbf{x}_*, \boldsymbol{\epsilon}, t}\Big[||\mathbf{v}_\theta(\mathbf{x}_t, t) - \dot{\alpha}_t \mathbf{x}_* - \dot{\sigma}_t \epsilon||^2\Big]. \tag{14}$$

This also corresponds to a reverse stochastic differential equation (SDE) given by:

$$d\mathbf{x}_t = \mathbf{v}(\mathbf{x}_t, t)dt - \frac{1}{2} w_t \mathbf{s}(\mathbf{x}_t, t)dt + \sqrt{w_t} d\bar{\mathbf{w}}_t, \tag{15}$$

where the score $\mathbf{s}(\mathbf{x}_t, t)$ is defined as:

$$\mathbf{s}(\mathbf{x}_t, t) = -\frac{1}{\sigma_t}\mathbb{E}[\epsilon|\mathbf{x}_t = \mathbf{x}]. \tag{16}$$

The score $\mathbf{s}(\mathbf{x}_t, t)$ can also be approximated using a model $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}, t)$, trained with the objective:

$$\mathcal{L}_{\text{score}}(\theta) = \mathbb{E}_{\mathbf{x}_*, \boldsymbol{\epsilon}, t}\Big[||\sigma_t \mathbf{s}_\theta(\mathbf{x}_t, t) + \epsilon||^2\Big]. \tag{17}$$

As shown in [2], any functions α_t and σ_t that satisfy the following three conditions:

1. $\alpha_t^2 + \sigma_t^2 > 0, \ \forall t \in [0, 1]$
2. α_t and σ_t are differentiable, $\forall t \in [0, 1]$
3. $\alpha_1 = \sigma_0 = 0, \ \alpha_0 = \sigma_1 = 1,$

lead to an unbiased interpolation process between $\mathbf{x}_0$ and ϵ. Example choices include linear interpolants ($\alpha_t = 1 - t, \sigma_t = t$) or variance-preserving (VP) interpolants ($\alpha_t = \cos(\frac{\pi}{2}t), \sigma_t = \sin(\frac{\pi}{2}t)$) [52].

An advantage of stochastic interpolants is that the diffusion coefficient (w_t) can be independently selected during sampling with the reverse SDE, even after training. This capability simplifies the design space compared to score-based diffusion models [36].

Table 3: **Defualt hyperparameter setup.** Unless other otherwise specified, we use these sets of hyperparameters for different models. In our component-wise analysis experiment, settings are also kept the same except those we point out in Table 1.

	SiT-B	SiT-L	SiT-XL	DiT-B	DiT-L	DiT-XL
Architecture						
Input dim.	32×32×4	32×32×4	32×32×4	32×32×4	32×32×4	32×32×4
Patch size	2	2	2	2	2	2
Num. layers	12	24	28	12	24	28
Hidden dim.	768	1024	1152	768	1024	1152
Num. heads	12	16	16	12	16	16
SRA						
Alignment blocks	$3 \rightarrow 8$	$6 \rightarrow 16$	$8 \rightarrow 20$	$3 \rightarrow 7$	$6 \rightarrow 14$	$8 \rightarrow 16$
Alignment time interval	$0 \sim 0.2$	$0 \sim 0.2$	$0 \sim 0.2$	$\lfloor 0 \sim 200 \rfloor$	$\lfloor 0 \sim 200 \rfloor$	$\lfloor 0 \sim 200 \rfloor$
Objective	smooth-ℓ_1	smooth-ℓ_1	smooth-ℓ_1	smooth-ℓ_1	smooth-ℓ_1	smooth-ℓ_1
EMA decay	0.999	0.999	0.999	0.999	0.999	0.999
Using projection head	✓	✓	✓	✓	✓	✓
Optimization						
Batch size	256	256	256	256	256	256
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW
lr	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
(β_1, β_2)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)
Interpolants or Denoising						
α_t	$1-t$	$1-t$	$1-t$	-	-	-
σ_t	t	t	t	-	-	-
w_t	σ_t	σ_t	σ_t	-	-	-
T	-	-	-	1000	1000	1000
Training objective	v-prediction	v-prediction	v-prediction	noise-prediction	noise-prediction	noise-prediction
Sampler	Euler-Maruyama	Euler-Maruyama	Euler-Maruyama	DDPM	DDPM	DDPM
Sampling steps	250	250	250	250	250	250
Guidance Scale	-	-	1.8 (if used)	-	-	-

C Hyperparameter and More Implementation Details

More implementation details. We implement our models based on the original DiT and SiT implementation. To speed up training and save GPU memory, we use mixed-precision (fp16) with a gradient clipping and FusedAttention [15] operation for attention computation. We also pre-compute compressed latent vectors from raw pixels via stable diffusion VAE [62] and use these latent vectors. We do not apply any data augmentation, but we find this does not lead to a big difference, as similarly observed in MaskDiT [77] and REPA [74]. We also use `stabilityai/sd-vae-ft-ema` for encoding images to latent vectors and decoding latent vectors to images. For projection head used for nonlinear transformation, we use two-layer MLP with SiLU activations [19]. The λ is set to 0.04 for DiT and 0.2 for SiT initially and followed the rules: $\lambda = \lambda_{\text{init}} * 0.1^{\left(\frac{N_e - 150}{1000}\right)}$, where $N_e > 150$ is the number of epoch, to change after 150 epochs for ensuring $\mathcal{L}_{\text{gen}}$ and $\mathcal{L}_{\text{sa}}$ at the same scale.

When using SiT-XL to generate images with classifier-free guidance [31], the guidance interval introduced in the study [40] with the same setting used in REPA [74] is applied, which has been demonstrated to yield a slight performance improvement.

Linear probing. We follow the setup used in REPA [74] and I-DAE [12], and use the code-base of ConvNeXt [50] to conduct the experiment. Specifically, we use AdaptiveAvgPooling and a batch normalization layer to process the latent features output by the model, then train a linear layer for 80 epochs. The batch size is set to 4096 with cosine decay learning rate scheduler, where the initial learning rate is set to 0.001.

Computing resources. We use 8 NVIDIA A100 80GB GPUs or 8 NVIDIA L40S 48GB GPUs for training largest model (XL); and use 4 NVIDIA A100 80GB GPUs or 4 NVIDIA L40S 48GB GPUs for training smaller model (L, B), our training speed is about 2.12 step/s with a global batch size of 256. When sampling, we use either NVIDIA A100 80GB GPUs or NVIDIA L40S 48GB GPUs or NVIDIA RTX 4090 24GB GPUs to obtain the samples for evaluation.

D Evaluation Metric

In this section we explain the main concept of metrics that we used for the evaluation.

- **FID** [28] measures the feature distance between the distributions of real and generated images. It uses the Inception-v3 network [66] and computes distance based on an assumption that both feature distributions are multivariate gaussian distributions.
- **sFID** [54] proposes to compute FID with intermediate spatial features of the Inception-v3 network to capture the generated images' spatial distribution.
- **IS** [64] also uses the Inception-v3 network but use logit for evaluation of the metric. Specifically, it measures a KL-divergence between the original label distribution and the distribution of logits after the softmax normalization.
- **Precision and recall** [41] are based on their classic definitions: the fraction of realistic images and the fraction of training data manifold covered by generated data.

E Baselines for Comparison

In what follows, we explain the main idea of baseline methods that we used for the evaluation.

- **ADM** [17] improves U-Net-based architectures for diffusion models and proposes classifier-guided sampling to balance the quality and diversity tradeoff and improve the performance.
- **VDM++** [37] proposes a simple adaptive noise schedule for diffusion models to improve training efficiency.
- **Simple diffusion** [32] proposes a diffusion model for high-resolution image generation by exploring various techniques to simplify a noise schedule and architectures.
- **CDM** [30] introduces cascaded diffusion models and trains multiple diffusion models starting from the lowest resolution and applying one or more super-resolution diffusion models for generating high-fidelity images.
- **LDM** [63] proposes latent diffusion models by modeling image distribution in a compressed latent space to improve the training efficiency without sacrificing the generation performance.
- **DiT** [57] proposes a pure transformer backbone for training diffusion models based on proposing AdaIN-zero modules.
- **SiT** [52] extensively analyzes how DiT training can be efficient by moving from discrete diffusion to continuous flow-based modeling.
- **SD-DiT** [79] leverages IBOT's [78] training paradigm that combines DINO loss [8] and BEIT loss [4] for efficiently training diffusion transformers.
- **MaskDiT** [77] proposes an asymmetric encoder-decoder scheme for efficient training of diffusion transformers, where they train the model with an auxiliary mask reconstruction task similar to MAE [26].
- **TREAD** [39] introduces a dynamic token routing strategy combined with the mask reconstruction task similar to MAE [26] and MaskDiT [77] to accelerate the training of diffusion models.
- **REPA** [74] achieves significant improvements in both training efficiency and generation quality by aligning latent feature of diffusion model with that of a large-scale data pre-trained representation model (*e.g.*, DINOv2 [56]).
- **MAETok** changes the SD-VAE to MAE-Tok which is trained with auxiliary mask reconstruction loss and align loss with three representation targets (HOG's [14], DINOv2's [56], and CLIP's [60]) and obtains diffusion transformer with better generation performance.

F Convergence Speed Against REPA

Here we provide quantitative comparison on convergence speed with SiT-XL baseline and SiT-XL + REPA [74]. It is worth noting that REPA leverages a powerful representation foundation model whose training data and resource are far beyond training a diffusion transformer on ImageNet.

As shown in Figure 8, although REPA converges quickly at the beginning of training due to its strong representation prior, the performance saturates after about 200 epochs. On the contrary, in SRA, since the teacher's ability is constantly enhanced during training and thus better and better

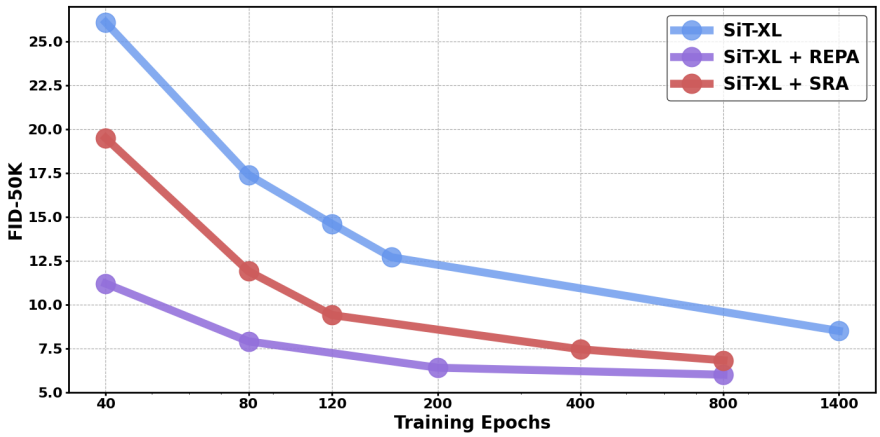


Figure 8: **Training epochs vs. FID plot** without classifier-free guidance (CFG). The benefit of REPA encounters saturation after approximately 200 epochs, while our method provides continuous performance improvement.

guidance is provided, the continuous performance improvement can be achieved. From another perspective, the representation guidance signal in REPA is fixed in every timestep and epoch, while our supervision signal is dynamically changing. This provides a wider variety of learning cues to help the model continuously improve the performance.

G Detailed Setup of Ablation Study

We now give the detailed experimental setup of Figure 6c in our ablation study. The accuracy rate of the horizontal axis in the figure is obtained by using the linear probing results of the original SiT-XL/2 checkpoint training for 7M iterations, while the vertical axis is the FID evaluation result of training 400K iterations with SRA without classifier-free guidance (CFG). Since we do not introduce any representation component and use the representation supervision signal only in generative training process, we consider this experiment can validate the effectiveness of our approach.

H Ablation Results of DiT

We also perform a similar analysis with DiT like those have done in Figure 6c, the results are showed in Figure 9. In short, we also observe that the generative capability of DiT with SRA is indeed strongly correlated with the representation guidance as observed in SiT with SRA.

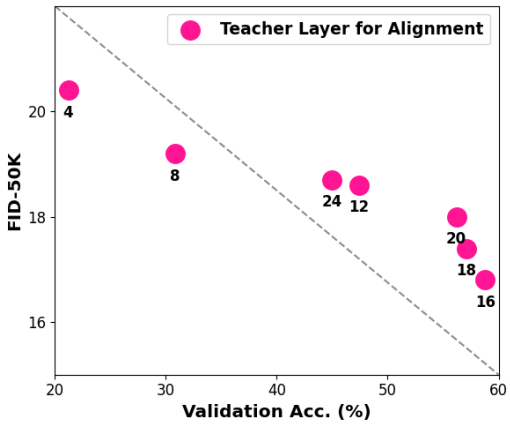


Figure 9: We also investigate the correlation between generation performance and the representation guidance of DiT + SRA. The similar tight coupling can also be seen.

I More Discussion on Related Work

Diffusion transformers. Currently, the diffusion model is progressively transitioning from a U-Net-based architecture to a Transformer-based one, owing to the superior scalability of the latter. At the beginning, U-ViT [3] shows transformer-based backbones with *skip connections* can be an effective backbone for training diffusion models. Then, DiT [57] shows skip connections are not even necessary components, and a pure transformer architecture can be a scalable architecture for training diffusion-based models. Based on DiT, SiT [52] shows the model can be further improved with continuous stochastic interpolants [2]. Moreover, Stable diffusion 3 [20], Lumina-Image 2.0 [59] and FLUX 1 [42] show pure transformers can be scaled up for challenging text-to-image generation, and this characteristic is also verified by Sora [5], CogvideoX [72] and Wan [69] in text-to-video field. Our work fouces on improves the training of DiT (and SiT) architecture based on our proposed self-representation alignment.

Exploring representation capacity of diffusion models. With the success of diffusion models to generate detailed images, many works have attempted to test out whether the discriminative semantic information can be found in diffusion models. GD [53] and DDAE [70] first observe that the intermediate representations of diffusion models have discriminative properties. Driving from this finding, I-DAE [12] deconstructs diffusion models to be a self-supervised Learner. Moreover, Repfusion [71] and DreamTeacher [43] propose knowledge distillation schemes using diffusion models to perform various downstream tasks (*e.g.*, semantic segmentation and object detection). Our work also try to explore representation capacity of diffusion model, but we focus on leveraging the representations in diffusion transformers to enhance their generation capacity.

Applying representation guidance to other generation tasks. In addition to pre-training of the class-conditional diffusion model, applying representation guidance can benefit other generation tasks. For example, lbGen [34] utilizes text feature from CLIP [60] as a low-biased reference to regulate diffusion model for low-biased dataset synthetics. RCG [44] focuses on unconditional generation, which first use the features from self-supervised image encoder to train a representation generator and subsequently used it output as the 'label' for image generation by a second generator. Diff-AE [58] uses a learnable encoder for discovering the high-level semantics and a diffusion model for modeling stochastic, these dual-encoding improves realistic of the generated images on attribute manipulation and image interpolation tasks. Different from these works, our study focus on class-conditional diffusion transformer's pre-training and exploiting representation guidance in itself and its own training paradigm. But we also believe our plug-and-play method can be easily applied to other tasks with benefits.

J More Qualitative Results

Below we show some uncurated generation results on ImageNet 256×256 from the SiT-XL + SRA. We use classifier-free guidance with $w = 4.0$.

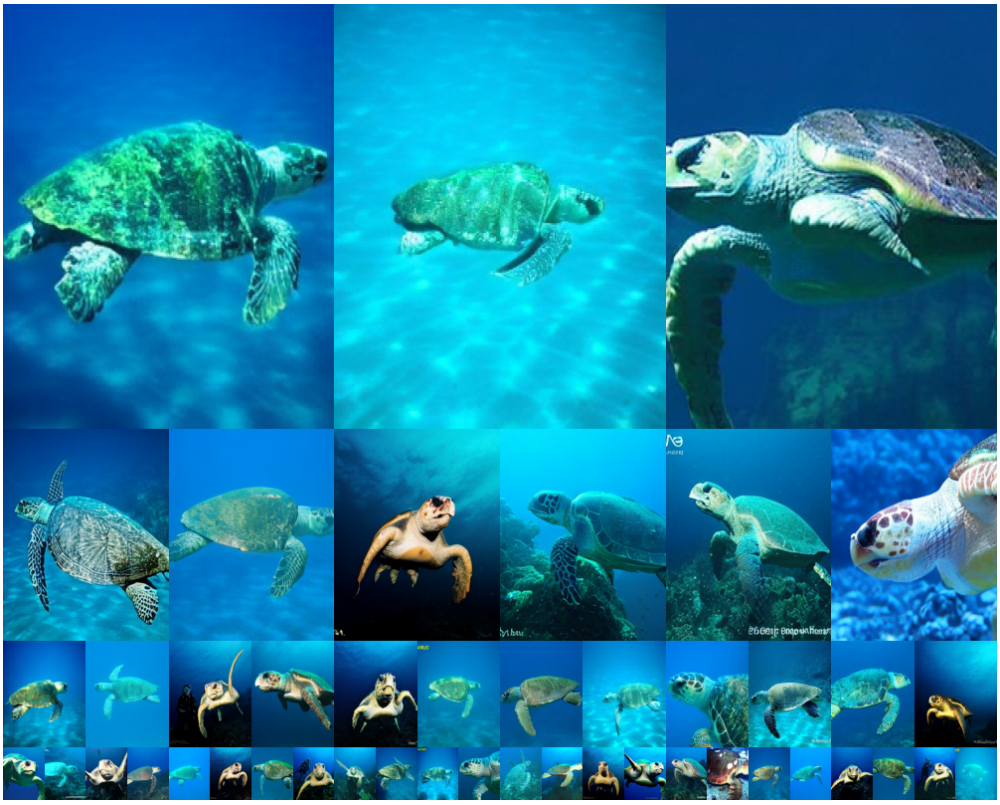


Figure 10: **Uncurated samples of loggerhead turtle (class label: 33).**

Figure 11: **Uncurated samples of sulphur-crested cockatoo (class label: 89).**

Figure 12: **Uncurated samples of golden retriever (class label: 207).**

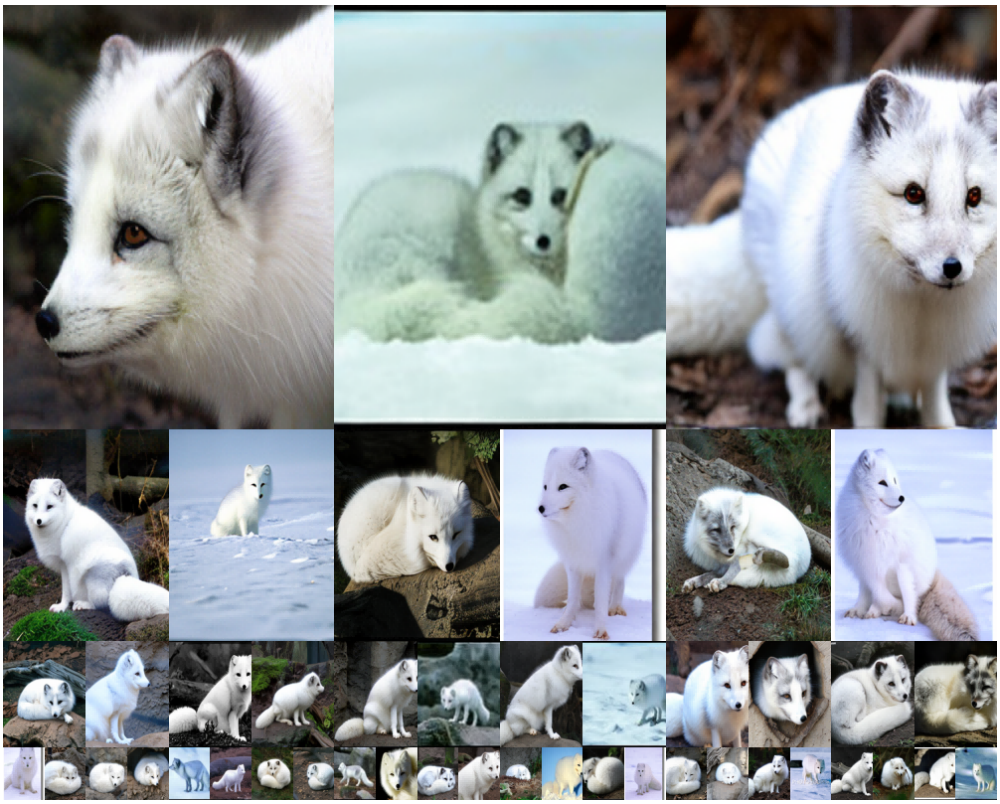


Figure 13: **Uncurated samples of white fox (class label: 279).**

Figure 14: **Uncurated samples of tiger (class label: 292).**

Figure 15: **Uncurated samples of red panda (class label: 387).**

Figure 16: **Uncurated samples of acoustic guitar (class label: 402).**

Figure 17: **Uncurated samples of balloon (class label: 417).**

Figure 18: **Uncurated samples of baseball (class label: 429).**

Figure 19: **Uncurated samples of fire truck (class label: 555).**

Figure 20: **Uncurated samples of ice cream (class label: 928).**

Figure 21: **Uncurated samples of cheeseburger (class label: 933).**

Figure 22: **Uncurated samples of coral reef (class label: 973).**

Figure 23: **Uncurated samples of lakeside (class label: 975).**