Why DiT Models Underperform as Representation Learners without Long Skip Connections

Benyuan Meng[1,2] **Yiliang Zhang**[3] **Jin-Wen Wu**[3]
Qianqian Xu[4,5] **Qingming Huang**[6,4,7] **Longtao Huang**[3]*
[1]Institute of Information Engineering, CAS
[2]School of Cyber Security, University of Chinese Academy of Sciences
[3]Alibaba Group
[4]Key Lab. of Intelligent Information Processing, Institute of Computing Technology, CAS
[5]Beijing Academy of Artificial Intelligence
[6]School of Computer Science and Tech., University of Chinese Academy of Sciences
[7]Key Laboratory of Big Data Mining and Knowledge Management, CAS
mengbenyuan@iie.ac.cn {suiyang.zyl,bage.wjw,kaiyang.hlt}@alibaba-inc.com
xuqianqian@ict.ac.cn qmhuang@ucas.ac.cn

Abstract

Diffusion feature — intermediate activations extracted from diffusion model backbones — has emerged as a promising approach for unifying generation and vision understanding (discrimination). Despite the advancement of diffusion backbones into Diffusion Transformers (DiT), it has been found in previous literature that DiT models tend to yield weaker features despite their stronger generation capability. This poses a threat to good generation-understanding unification. In this work, we show via controlled from-scratch training that long skip connections (LSCs) affect feature quality: a DiT with LSCs can produce better features. Furthermore, through detailed diagnosis, we reveal the mechanism causing this effect. Specifically, the input to a diffusion backbone contains both the latent image and the diffusion noise, where the diffusion noise, though appearing in the form of noises, actually reflects the information of the diffusion trajectory. To make the final prediction of a noisy variable (v or ϵ), the diffusion backbone needs both signals. However, carrying both signals through the main backbone effectively "pollutes" the image signal with noises, leading to poor feature quality. In comparison, LSCs can shortcut diffusion noises directly to where they are needed, leaving much weaker noise signals in the main backbone, thus improving feature quality. To name the broader impact, this reveals that LSCs are not a mere trick but closely related to the diffusion objective. The code is here.

1 Introduction

Diffusion models are a trending type of generative models nowadays [16, 10, 30, 29]. Beyond generation, it is also an important and emerging topic to explore the unification of generation and vision understanding tasks, such as semantic segmentation [9], depth estimation [4], or semantic correspondence [26] with diffusion models [2, 51, 45, 17, 19, 20, 7]. *Diffusion feature* is an important sub-family of this research trend. Every diffusion model *pre-trained for generation*, *e.g.*, Stable Diffusion v1.5 [30], has a neural network backbone for ϵ- or v-prediction [16, 33], and many intermediate activations are created in the computation process. If we extract these activations from the backbone, they could serve as versatile vision features, essentially similar to how people extract activations from ResNet [13] backbones as features. High-quality diffusion features can enable better generation-understanding unification [40, 28].

*Corresponding author.

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

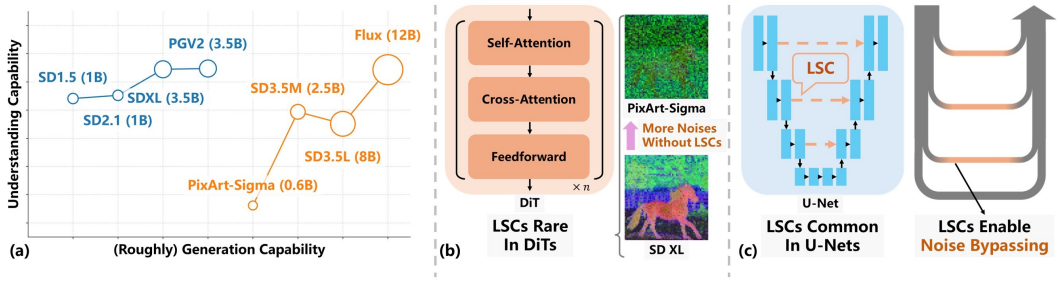


Figure 1: (a) Compared with U-Net diffusion models, DiT models yield diffusion features with relatively low understanding capability. (b) Most DiTs do not have LSCs, so the diffusion noise has to go through the target reconstruction section, causing features in this section to degrade. (c) We reveal a unique function of long skip connections (LSCs) in diffusion backbones: they work as shortcuts to transport the diffusion noise directly from the downsampling stage to the upsampling stage, thus *bypassing* the middle section, where the most performative diffusion features can be extracted.

When studying diffusion features, the basic belief and hypothesis are that the advancement of generative capability naturally brings the advancement of understanding capability [25]. Here, the generative capability can be measured with FID score [15], CLIP score [14], or user study [29], and the understanding capability can be measured using the performance on standard dense perception tasks [9, 4, 26]. With this belief in mind, and considering that the mainstream architecture for diffusion models is turning to Diffusion Transformers (DiT) [27] from U-Nets, it is important that we can extract high-quality DiT features. However, it is reported that DiT models tend to yield weaker features despite their stronger generation [11] (Figure 1(a)). This both poses a threat to generation-understanding unification and suggests a representation degradation phenomenon hidden behind the good generation, which may manifest one day and limit the ceiling of generation.

Notably, although the phenomenon is reported [11], the cause is still only preliminarily understood: the massive activation explanation appears to be applicable mostly to correspondence tasks (Section A.1), while the feature degradation exists universally (Section A.3). This indicates that the massive activation explanation is not exhaustive, and there should be a more universal cause. On the other hand, many recent studies report that adding long skip connections (LSCs), a structure in U-Nets that shortcut information from the downsampling stage directly to the upsampling stage, can enhance DiT generation. Inspired by this, we confirm via a controlled comparison experiment that the wide absence of LSCs from standard DiTs is a more universal cause, or at least part of it.

We do not stop at knowing LSCs are good but further ask *why*. It is common sense that such skip connections can carry high-frequency information like edges, and this is also used to explain the effectiveness of LSCs for DiT generation in some literature [1]. However, suppose LSCs only diverge some high-frequency semantics like edges from the middle part; why are features here improved simply by *removing* some high-frequency semantics from them? Through further hypothesizing and validating, we reveal a deeper mechanism that is fundamentally new compared to commonsense. Specifically, [25] shows that diffusion backbones do not *directly* predict a noisy variable from a noisy input, but instead reconstruct a semantic representation of the clean image z_0 in the middle of the network, based on which the final noise prediction is made. To predict the final noisy variable, *both* the reconstructed representation *and* the diffusion noise carried in the noisy input should be transported to the final noise prediction section. Here, the diffusion noise is *not* a random noise: it appears in the form of noises but reflects certain information, *i.e.*, the current denoising trajectory. *Without LSCs*, the only possible passage for the diffusion noise is through the entire target reconstruction section, thus "*polluting*" the features extracted from this section, since it still takes the form of noises (Figure 1(b)). *With LSCs*, however, the diffusion noise can take the shortcut and thus bypass the target reconstruction section, maintaining high feature quality in this section (Figure 1(c)). Hence, we add to the knowledge that LSCs can carry high-frequency semantics that they also carry diffusion noises. This reveals that LSCs are not a mere trick to boost generation but are closely related to the diffusion objective.

In general, our contribution is as follows:

- We provide one of the first systematic studies focusing on the discriminative quality of DiT features and notice a significant feature quality discrepancy between DiTs and U-Nets.
- We attribute the feature quality discrepancy to the LSC structure and provide a detailed, rigorously supported mechanistic explanation.
- Despite the initial representation learning motivation, we hope our mechanistic findings could make LSCs in DiTs a more widely adopted practice rather than an optional heuristic.

2 Related Work

2.1 DiT Feature

There are currently mainly three approaches to exploiting the understanding potential from a generative diffusion model: diffusion classifier [20, 7], generative modeling [17, 19, 40], and diffusion feature [2, 51, 45, 25, 35, 11]. Diffusion feature extracts intermediate activations from a diffusion backbone as features, regarding the diffusion backbone as an information bottleneck that compresses an image into a latent embedding. These features are dense and rich in information, but not directly interpretable by humans. Thus, a downstream model is required to process these features and produce the final predictions.

Most previous knowledge about diffusion feature is based on U-Net diffusion backbones. [11] is the first to extend this methodology to DiT and surprisingly find that DiT features are generally worse. The explanation proposed in [11] is *massive activations* [34], where the scale of a small number of elements in the feature is extremely large.

To some extent, diffusion feature can be viewed as a representation learning method. However, the model is not directly trained for representation learning, but a generatively trained diffusion model is adapted to this task. Compared to typical representation learning methods like DINO [5], diffusion feature performs especially well in scenarios where the background is relatively identical and the information is sparse [44].

2.2 LSC-Augmented DiT Models

Conventionally, long skip connections (LSCs) are a typical structure in U-Nets. Since U-Nets adopt a downsampling-upsampling architecture, it is believed mandatory to utilize such skip connections to shortcut low-level/high-frequency semantics like edges, which will otherwise be lost during downsampling. At first, when the diffusion model community shifts from U-Net backbones to DiT backbones, LSCs were naturally discarded, since DiTs have identical resolution across blocks. However, quite a few recent studies find that adding LSCs to DiT can enhance generation [1, 6, 41]. Compared to them, our findings are fundamentally different in two ways: (i) We find LSCs are beneficial not only to generation but also to representation learning. (ii) We reveal the *diffusion noise* bypassing behavior of LSCs, which is different from the high-frequency semantic shortcutting [1] and loss smoothing [6] explanations.

Additionally, there are also LSC-augmented DiT models with actual resolution changes [38, 37]. To isolate the contribution of LSCs, we use *isotropic* DiTs as the backbone to add LSCs onto. For further comparison with models with actual resolution changes, please refer to Section A.2.

3 Preliminaries

3.1 Diffusion Models

Based on [22], the diffusion process starts at a point sampled from the source distribution, progressively moving it towards the target distribution, where the two distributions are typically noises and images. Each step during this process is called a timestep $t \in [0, 1]$. At each timestep, a backbone network is used to predict the velocity of the movement. We can denote this network as v_θ, parameterized by θ. This network used to be U-Nets, but now it is mostly ViTs, known as DiTs.

To actually train the model, we need to sample an image x_0, its corresponding user prompt c, and a timestep t, calculating a ground truth based on them. First, the image is compressed into the latent

space by a pre-trained VAE as $z_0 = \text{VAE}(x_0)$, and we sample a Gaussian noise z_1. Then, we have

$$\begin{aligned} z_t &= tz_0 + (1-t)z_1 \\ v_t &= \frac{dz_t}{dt} = z_0 - z_1 \end{aligned} \tag{1}$$

Afterward, we can train the backbone network to predict the velocity:

$$\mathcal{L} = \mathbb{E}_{(z_0,c)\sim\mathcal{D},z_1,t}||v_\theta(z_t,t,c) - v_t||^2 \tag{2}$$

The formulation above is denoted as *flow matching*, and the more conventional DDPM [16] and score matching [33] frameworks can be seen as special forms of flow matching. Notably, although the difference in the theoretical framework may not be that important, the prediction objective of the backbone model, *i.e.*, ϵ-, v-, or x-prediction, does matter. In the main body of this manuscript, we will use ϵ-prediction for experiments, and we will show in Section A.4 how our findings extend to v- and x-prediction.

3.2 Diffusion Feature

To extract a feature from a diffusion backbone, we need an image, a selected timestep, and, optionally, a prompt describing the image content. Denoting them as x_0, t, c, we have:

$$\begin{aligned} z_0 &= \text{VAE}(x_0) \\ z_t &\sim q(z_t|z_0,t) \\ v &= v_\theta(z_t,t,c) \end{aligned} \tag{3}$$

Respectively, the three lines indicate encoding the image into latent space with a VAE, sampling a noisy latent corresponding to the timestep t, and sending all three inputs into the diffusion backbone. We discard the output v. Instead, we gather some intermediate activations as features. Such activations can be extracted from a variety of spots, such as the residual flow between two adjacent layers or the query vector used to compute the attention matrix in an attention layer [25]. In DiT models, these activations are patchified 1D features, and we can reshape them back to 2D features and resize them to the original image size using bilinear interpolation.

Obviously, there are many legal feature extraction spots, but they cannot all be harvested due to limited computational resources, so there is a need to select a few near-optimal activations as features. To this end, a three-step strategy can be taken [25].

(i) Based on both PCA analysis and quantitative comparison within small ranges, study the properties of features in this backbone.

(ii) Based on the understanding of feature properties, rule out low-quality features qualitatively.

(iii) Perform a filtered quantitative comparison among the remaining features on a small and fast downstream task to find the most performative features. In this study, we adopt training-free semantic correspondence [26] as the task.

After selecting activations as features, diffusion features can be used in a very straightforward way. Basically, suppose the original input to a visual perception task is a raw image; we could replace the image with the corresponding diffusion feature. Commonly, such replacement requires the dedicated training of a compatible downstream task head [2], and we could fine-tune the diffusion backbone on the task for better performance [51], or, in much rarer cases, use diffusion features in a totally training-free manner [49].

4 Long Skip Connections Matter

4.1 Core Hypothesis

With the recent emergence of LSC-augmented DiTs [1, 6, 41], it gradually becomes acknowledged that LSCs are beneficial for DiT *generation*. Furthermore, the basic belief of diffusion feature studies tells us a stronger generation model should generally lead to stronger features. Hence, it is natural to hypothesize that LSCs are beneficial for DiT *features* as well. For a visual illustration of the architectures of U-Net and DiT diffusion models, refer to Figure 2(a,b).

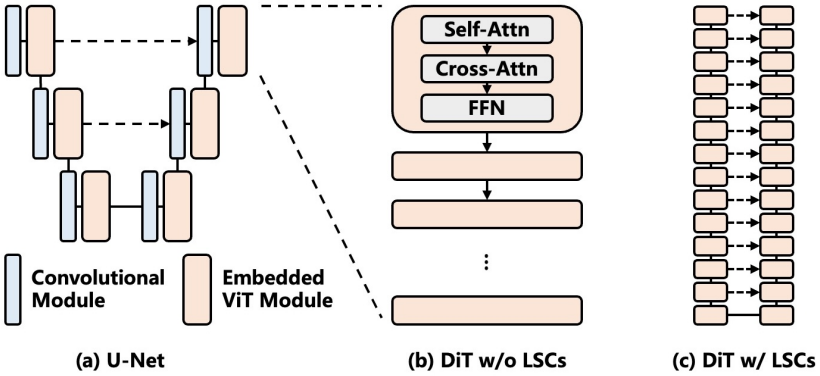


Figure 2: (a) A typical U-Net architecture of modern diffusion models is not purely convolutional but contains many large embedded ViT modules. (b) A DiT backbone can, from a certain perspective, be viewed as extracting an embedded ViT module and greatly extending it. (c) We can fold the DiT chain in half to form an LSC-augmented DiT.

As preliminary evidence, we test on pre-trained DiT-XL/2 [27] and U-ViT [1] models where the generative capability roughly aligns. The result that U-ViT (with LSCs) yields significantly better features than DiT-XL/2 (without LSCs) (53.53 v.s. 44.01 on SPair (a)) implies the benefit of LSCs. Nevertheless, the full results in Section A.5 also suggest a controlled experiment to exclude the confounders.

4.2 Hypothesis Validation

We further conduct a controlled from-scratch training experiment to validate our hypothesis above. Basically, we train two DiT [27] models from scratch, with the only difference being whether there are LSCs or not, and compare them on six downstream benchmarking settings.

Training Dataset. We train our models on ImageNet [18] 256×256. This dataset contains 1,281,166 images in 1000 classes, all in 256×256 resolution.

Models. We choose the DiT-XL/2 option in [27]. The only modification we make to the models is adding LSC structures. The LSC structure we use is formulated as

$$x \leftarrow \text{Linear}(\text{Norm}(\text{Concat}(x, \text{skip}))) \tag{4}$$

where x is the inter-layer residual flow activation, skip is the activation shortcut here through LSCs, Linear is a single fully connected layer, Norm is LayerNorm, and Concat is concatenation along the channel dimension. A standard DiT architecture is linear, and intuitively, our modification folds the architecture in half from the middle to enforce it into a conceptual "U" shape. For a visual illustration, please refer to Figure 2(c).

Training Details. We simply keep all hyperparameters in the original script unchanged, using a global batch size of 256. Additionally, both models are trained for 400k steps by default. To further isolate the contribution of LSCs from the models' generative capability, we also provide a *w/ LSC 100k* setting trained for fewer steps to align the FID scores.

Evaluation. For the generation capability, we use FID scores as the metric, measured under CFG=1.5. We use six benchmarking settings for the discriminative performance, which are listed in Table 1.

(i) *SPair (a)* to *SPair (c)*: these use SPair-71k [26] dataset as the benchmark. The difference between the three settings is the parameters trainable on the downstream task: *SPair (a)* is training-free, *SPair (b)* trains only a downstream head, and *SPair (c)* fine-tunes the DiT model.

(ii) *Horse-21* and *CityScapes*: these use semantic segmentation task, one on Horse-21 [2] dataset and the other on CityScapes [8] dataset, both only training a downstream head.

(iii) *Depth*: this uses depth estimation task on NYU Depth v2 [31], and we fine-tune the DiT model on this setting.

Table 1: Comparison of the six benchmarking settings we use throughout this study.

Setting Name	Task	Dataset	Metric	Trainable Params
SPair (a)	Semantic Correspondence	SPair-71k	PCK@0.1	None
SPair (b)	Semantic Correspondence	SPair-71k	PCK@0.1	Downstream Head
SPair (c)	Semantic Correspondence	SPair-71k	PCK@0.1	All
Horse-21	Semantic Segmentation	Horse-21	mIoU	Downstream Head
CityScapes	Semantic Segmentation	CityScapes	mIoU	Downstream Head
Depth	Depth Estimation	NYU Depth v2	d1	All

Table 2: Discriminative performance evaluated on all six settings as listed in Table 1. Plus, the FID scores are obtained under CFG=1.5. Best/second marked as **bold** and underlined.

LSC Setting	FID	SPair (a)	SPair (b)	SPair (c)	Horse-21	CityScapes	Depth
w/o LSC	20.87	34.68	52.62	64.70	30.35	50.22	0.70
w/ LSC	14.39	**41.10**	**60.68**	**65.69**	**36.32**	**51.55**	**0.74**
w/ LSC 100k	21.67	38.85	57.36	64.43	35.24	50.70	0.73

Unlike in generation, classifier-free guidance (CFG) is not commonly used for feature extraction.

The results are shown in Table 2. We can observe that *w/ LSC* is significantly stronger than *w/o LSC* across all settings, even for *w/ LSC 100k* where the FID scores are aligned. Thus, we could causally confirm that *LSCs are indeed an important contributing factor*. More importantly, even *w/ LSC 100k* outperforms *w/o LSC*, where the two models have similar generation capability.

For additional information, we also adopt Correlation Decay Slope (CDS) [32] for evaluation. The results in Figure 3 are the negative slopes, where a moderate slope indicates stronger feature expressiveness, while a slope that is either too low or too high is not favored. We can observe that in middle layers (12-16), *w/ LSC* shows higher slope values compared to *w/o LSC*. This supports the benefit of LSCs.

5 WHY Long Skip Connections Matter

5.1 Why Do We Not Fully Understand LSCs Yet

At first glance, the mechanics behind LSCs' contribution should be already known. However, we will show that two competing explanations cannot fully explain our previous experiments.

(i) Feature enhancement does not solely stem from better generation. In Table 2, even when we manually align the generation capability with *w/ LSC 100k*, the feature quality is still better than *w/o LSC*. (ii) LSCs shortcutting high-frequency semantics (*i.e.*, edges) [1] cannot fully explain the observation. Since LSCs can shortcut high-frequency semantics, the features extracted from the middle part should have more concentration of low-order semantics, which means they might perform better for coarse-grained tasks like image classification. However, what we have tested the models on are fine-grained tasks such as semantic segmentation. Hence, there should be some additional mechanisms that co-govern the gain brought by LSCs.

5.2 Target Reconstruction Behavior

Our hypothesis to explain why LSCs have such a positive effect is based on a finding in [25]. Here, we first re-formulate the "diffusion noise" property found in [25] as "target reconstruction". Note that we do *not* claim originality for this subsection's content. It is here to smooth the narrative later.

Based on the flow matching formulation (Section 3), both the input variable and the predicted variable of the diffusion backbone network are noisy variables. A natural interpretation of this would be that diffusion activations should be mostly noisy. However, the diffusion noise property states that, in the middle section of the network, intermediate activations are rather clean from noises. Actually, these activations are an important source of high-quality diffusion features. Now, considering that diffusion

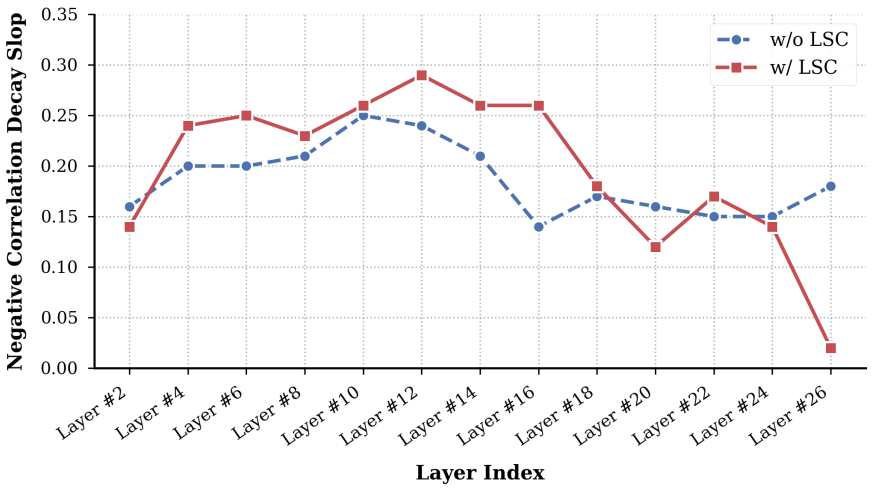


Figure 3: Correlation Decay Slope measuring results, where we report the negative slope of the linearly regressed lines.

features extracted from the middle section in fact reflect the semantics of the target image z_0, this middle section's role is to reconstruct an internal representation for the target image based on the noisy input z_t. Then, the network could encode its final prediction based on this reconstructed target representation. In conclusion, this target reconstruction behavior shows that diffusion backbones take *two rather than one* step to make their final prediction.

To facilitate the narrative, we conceptually divide the diffusion backbone into three coarse sections based on their roles, which are indicated by three different colors, red, green, and blue, in Figure 4:

(i) **Noise suppression**: To prepare for the later acquisition of the target representation, the noises in the inputs are gradually refined and suppressed to some extent.

(ii) **Target reconstruction**: Next, the backbone reconstructs a latent representation of the target image latent within activations.

(iii) **Noise prediction**: Finally, the model predicts the velocity, which is a noisy quantity.

5.3 Diffusion Noise Bypassing Hypothesis

Next, we develop upon this target reconstruction behavior to form our diffusion noise bypassing hypothesis. Consider the flow matching formulation: $v_t = z_0 - z_1$. We hypothesize that, to make the final prediction, *both* the target information (z_0) *and* the diffusion noise (z_1) are required, just like how both the starting and terminating points are required to determine a direction. Of the two, the target information is provided by the target reconstruction section, and the diffusion noise is carried within the input variable.

Furthermore, in order for the noise prediction section to make the prediction, both target and diffusion noise signals should be accessible to the noise prediction section. Since the target reconstruction section and the noise prediction section are directly adjacent, the target information can be transported to the noise prediction section simply through the main residual flow. However, how the diffusion noise is routed is where LSCs play their role. *Without LSCs*, as shown in Figure 4(right), the diffusion noise must also go through the main residual, but there is a target reconstruction section in between the noise suppression and noise prediction sections. Therefore, the activations in the target reconstruction section have to spare some capacity to carry the diffusion noise, which makes their original goal, *i.e.*, target reconstruction, degrade. *With LSCs*, as shown in Figure 4(left), however, the diffusion noise can directly take the shortcut and *bypass* the target reconstruction section, thus maintaining strong feature quality in the target reconstruction section. Thus, we make a hypothesis that could explain why LSCs can improve diffusion feature quality.

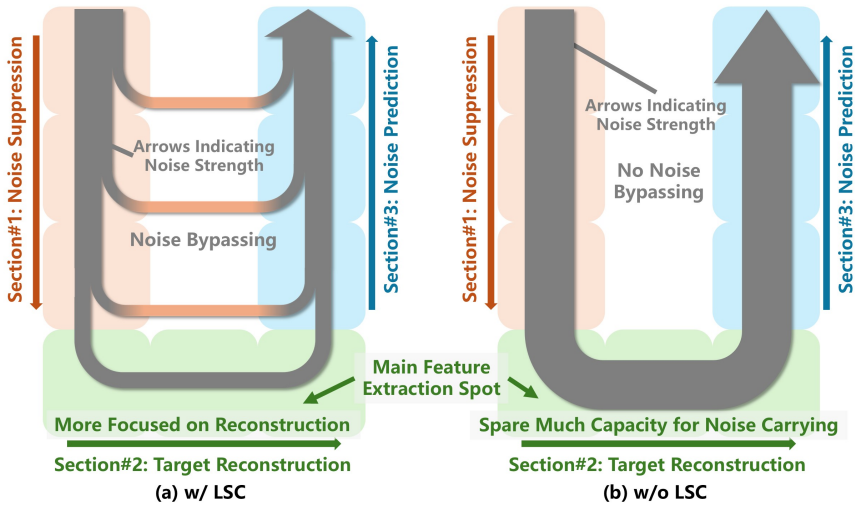


Figure 4: The noise bypassing hypothesis, where we use gray arrows to indicate noises and their *width as noise strength*. With LSCs, the diffusion noise is shortcut through LSCs, leaving only weak noises into the target reconstruction section. Without LSCs, the diffusion noise is carried through the target reconstruction section, leading to feature degradation.

Table 3: Causal intervention experiment to separate the effect of high-frequency semantics and diffusion noises. All settings are trained for 400k steps before evaluation.

Setting	Loss@1k	Loss@5k	FID	SPair (a)	SPair (b)	SPair (c)	Horse-21
w/o LSC	-	-	20.87	34.68	52.62	64.70	30.35
w/ LSC	0.1744	-	14.39	41.10	60.68	65.69	36.32
$z_0^{\text{high}} + \epsilon$	0.1189	0.0705	NaN	NaN	NaN	NaN	NaN
$z_0^{\text{high}} + \epsilon'$	0.1309	0.1222	37.04	26.76	44.43	58.43	22.59
$z_0'^{\text{high}} + \epsilon$	0.1196	0.1023	306.92	20.68	39.65	50.97	18.67

5.4 Hypothesis Validation

First, we use mutual information probing to actually measure the amount of the target information and diffusion noise described in the noise bypassing hypothesis, hoping to provide *correlation* evidence. The target information contained in each activation is defined as the mutual information between this activation and the target latent (z_0), and the diffusion noise contained in each activation is defined as the mutual information between this activation and the noise added to the latent image (z_1). To measure the mutual information, we adopt the Mutual Information Neural Estimator (MINE) [3], with a setting of (MINE hidden channel as 512, projection head channel as 256, training steps as 600). The results are plotted in Figure 5, and we can see that LSCs can indeed reduce the amount of diffusion noise and increase the amount of target information in the target reconstruction section (roughly the middle section). We also provide an alternative probing using a different MINE setting in Section A.6.

Next, to provide relatively *causal* evidence, we conduct a controlled study. Here, what we already know is that LSCs mainly carry high-frequency information. What we want to validate is that LSCs carry not only high-frequency semantics but also diffusion noises. The experiments we design include three settings besides the baseline:

(i) Replacing the $skip$ variable in Equation (4) with the linear addition of the high-frequency part of the original latent (z_0^{high}) and the original noise (z_1). This serves as a replacement baseline.

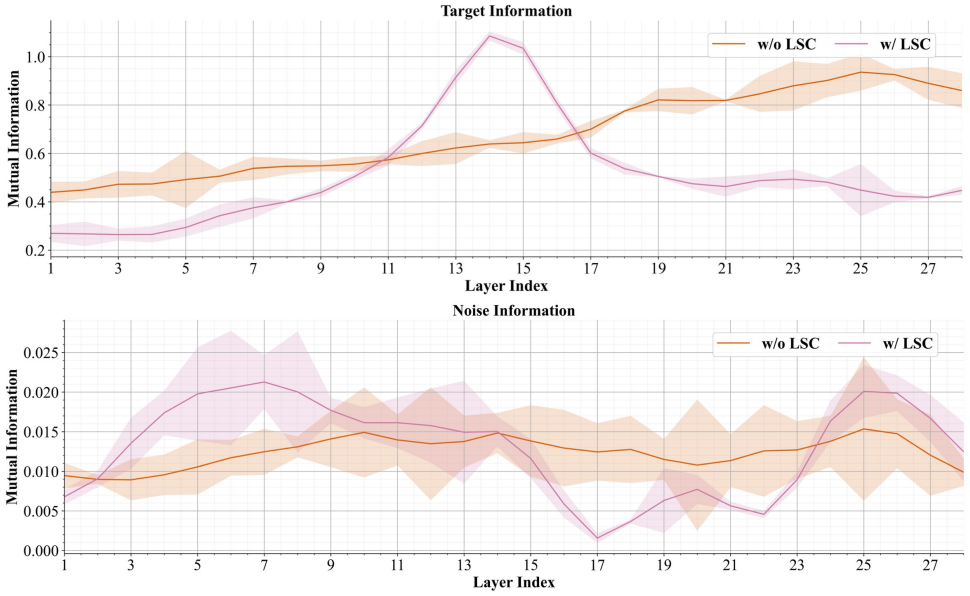


Figure 5: Mutual information between each layer's activation and target/diffusion noise, measured using MINE. Shadows indicate variance across three seeds.

(ii) Compared to setting (i), replacing the original noise with a random noise (z_1'). This isolates the contribution of high-frequency semantics.

(iii) Compared to setting (i), replacing the original latent with a latent from a different image ($z_0'^{high}$). This isolates the contribution of diffusion noises.

We can observe from Table 3 that:

(i) All three settings show smaller losses yet weaker representation *and* generation performance compared to *w/ LSC*. Setting (i) even collapses to NaN, and Setting (ii) fails to converge (very high FID scores). This is a sign of representation collapse: injecting external signals creates some information shortcut and thus trivializes the task, making training worse.

(ii) Settings (ii) and (iii) both show different degradation severity compared to setting (i), meaning that removing either of the two signals will affect the training process. This further confirms that both high-frequency semantics and diffusion noises play a non-negligible role in making LSCs work.

Nevertheless, since the training process is highly complex, we cannot say for sure from the results that diffusion noises are more important than high-frequency semantics or vice versa. In conclusion, LSCs actually carry a *mixture* of high-frequency semantics and diffusion noises, where both of them are non-negligible. This effectively extends our previous knowledge that only covers high-frequency semantics but not diffusion noises.

5.5 Discussion

Our diffusion noise bypassing hypothesis serves as knowledge of the inner workings of diffusion model backbones. Based on this new knowledge, we can see that LSCs can enable diffusion backbones to work in a more optimized way: the activations in the target reconstruction section no longer need to spare much capacity to carry the diffusion noise. Hence, *even if one's goal is not related to features, but only generally better diffusion models, we believe LSCs are still beneficial*. For example, Skip-DiT [6] and U-ViT [1] extensively show that LSCs can enhance *generation*, where we believe there is a certain causal relation. To be specific, REPA [48] finds that the quality of inner representations in diffusion backbones is essential to generation, where better representations lead to stronger generation and faster convergence. Hence, it is likely that the stronger generation brought by LSCs *partially* stems from the stronger feature they enable. Furthermore, compared to

explicitly aligning diffusion features to an external representation model in REPA, LSCs introduce less constraint and can thus better exploit the advantage of the diffusion training objective itself.

6 Conclusion & Limitation

This work reveals that long skip connections (LSCs) are beneficial for DiT, with regard to not only generation but also feature quality. This can also explain why current large-scale DiT models tend to yield weaker features than U-Net diffusion models, despite their stronger generation capability. More importantly, we propose a mechanistic hypothesis termed *diffusion noise bypassing* to explain why LSCs have such an effect. In general, we show that LSCs are not merely an engineering trick, but actually have a close relation to the diffusion process itself. Thus, we hope this work can persuade more researchers and developers to consider adding LSCs back to the backbone models in the DiT era.

Unfortunately, we were not able to effectively augment existing pre-trained DiT models with LSCs, which has been found to be very difficult (Section A.8). Additionally, our controlled experiments are based on the conventional DiT architecture rather than the MMDiT architecture, which remains underexplored. We also do not claim exhaustiveness of our LSC explanation as the cause of DiT feature degradation.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China: 62525212, U23B2051, 62236008, 62441232, 62521007 and U21B2038, in part by Youth Innovation Promotion Association CAS, in part by the Strategic Priority Research Program of the Chinese Academy of Sciences, Grant No. XDB0680201, and in part by the project ZR2025ZD01 supported by Shandong Provincial Natural Science Foundation.

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*, pages 22669–22679, 2023.

[2] Dmitry Baranchuk, Andrey Voynov, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. In *International Conference on Learning Representations*, pages 1–15, 2022.

[3] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International conference on machine learning*, pages 531–540, 2018.

[4] Alexey Bochkovskiy, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R. Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. In *International Conference on Learning Representations*, 2025.

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision*, pages 9630–9640, 2021.

[6] Guanjie Chen, Xinyu Zhao, Yucheng Zhou, Xiaoye Qu, Tianlong Chen, and Yu Cheng. Towards stabilized and efficient diffusion transformers through long-skip-connections with spectral constraints. In *IEEE/CVF International Conference on Computer Vision*, pages 17708–17718, 2025.

[7] Huanran Chen, Yinpeng Dong, Shitong Shao, Zhongkai Hao, Xiao Yang, Hang Su, and Jun Zhu. Your diffusion model is secretly a certifiably robust classifier. *CoRR*, abs/2402.02316:1–36, 2024.

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016.

[9] Gabriela Csurka, Riccardo Volpi, and Boris Chidlovskii. Semantic image segmentation: Two decades of research. *CoRR*, abs/2302.06378, 2023.

[10] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat gans on image synthesis. In *Annual Conference on Neural Information Processing Systems*, pages 8780–8794, 2021.

[11] Chaofan Gan, Yuanpeng Tu, Xi Chen, Tieyuan Chen, Yuxi Li, Mehrtash Harandi, and Weiyao Lin. Unleashing diffusion transformers for visual correspondence by modulating massive activations. In *Advances in Neural Information Processing Systems*, pages 1–24, 2025.

[12] Shijie Hao, Yuan Zhou, and Yanrong Guo. A brief survey on semantic segmentation with deep learning. *Neurocomputing*, 406:302–321, 2020.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[14] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7514–7528. Association for Computational Linguistics, 2021.

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Annual Conference on Neural Information Processing Systems*, pages 1–12, 2020.

[17] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9492–9502, 2024.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1106–1114, 2012.

[19] Hsin-Ying Lee, Hung-Yu Tseng, Hsin-Ying Lee, and Ming-Hsuan Yang. Exploiting diffusion prior for generalizable dense prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7861–7871, 2024.

[20] Alexander C. Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. Your diffusion model is secretly a zero-shot classifier. In *IEEE/CVF International Conference on Computer Vision*, pages 2206–2217, 2023.

[21] Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 36115–36125, 2026.

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

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF International Conference on Computer Vision*, pages 9992–10002, 2021.

[24] Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, volume 15135, pages 23–40, 2024.

[25] Benyuan Meng, Qianqian Xu, Zitai Wang, Xiaochun Cao, and Qingming Huang. Not all diffusion model activations have been evaluated as discriminative features. In *Annual Conference on Neural Information Processing Systems*, pages 55141–55177, 2024.

[26] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Spair-71k: A large-scale benchmark for semantic correspondence. *CoRR*, abs/1908.10543:1–5, 2019.

[27] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision*, pages 4172–4182, 2023.

[28] Yuxin Peng, Zishuo Wang, Geng Li, Xiangtian Zheng, Sibo Yin, and Hulingxiao He. A survey on fine-grained multimodal large language models. *Chinese Journal of Electronics*, 35:771–803, 2026.

[29] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: improving latent diffusion models for high-resolution image synthesis. *CoRR*, abs/2307.01952:1–21, 2023.

[30] 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*, pages 10674–10685, 2022.

[31] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *European Conference on Computer Vision*, volume 7576, pages 746–760, 2012.

[32] Jaskirat Singh, Xingjian Leng, Zongze Wu, Liang Zheng, Richard Zhang, Eli Shechtman, and Saining Xie. What matters for representation alignment: Global information or spatial structure? In *International Conference on Learning Representations*, 2026.

[33] 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.

[34] Mingjie Sun, Xinlei Chen, J. Zico Kolter, and Zhuang Liu. Massive activations in large language models. *CoRR*, abs/2402.17762, 2024.

[35] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. In *Annual Conference on Neural Information Processing Systems*, pages 1–27, 2023.

[36] Ye Tian, Ying Fu, and Jun Zhang. Transformer-based under-sampled single-pixel imaging. *Chinese Journal of Electronics*, 32:1151–1159, 2023.

[37] Yuchuan Tian, Hanting Chen, Mengyu Zheng, Yuchen Liang, Chao Xu, and Yunhe Wang. U-repa: Aligning diffusion u-nets to vits. In *Advances in Neural Information Processing Systems*, pages 1–22, 2025.

[38] Yuchuan Tian, Zhijun Tu, Hanting Chen, Jie Hu, Chao Xu, and Yunhe Wang. U-dits: Downsample tokens in u-shaped diffusion transformers. In *Advances in Neural Information Processing Systems*, volume 37, pages 98321–98345, 2024.

[39] Zijie J. Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. DiffusionDB: A large-scale prompt gallery dataset for text-to-image generative models. *arXiv:2210.14896 [cs]*, 2022.

[40] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Shengming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report. *CoRR*, abs/2508.02324, 2025.

[41] Linzhi Wu, Aoran Mei, Xiyue Wang, Guo-Niu Zhu, and Zhongxue Gan. U-dit policy: U-shaped diffusion transformers for robotic manipulation. *CoRR*, abs/2509.24579, 2025.

[42] Linhui Xiao, Guiping Cao, Mingyue Guo, Xianchao Guan, Fan Yang, Ming Tao, Xin Li, Yuxin Peng, and Yaowei Wang. A survey on the green development of large models: From resource-efficient architectures to hardware-software co-design. *Chinese Journal of Electronics*, 35:1–24, 2026.

[43] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *European Conference on Computer Vision*, pages 418–434, 2018.

[44] Yingpeng Xie, Hao Chen, Jing Qin, Yongtao Zhang, Lei Dong, Jie Du, Tianfu Wang, and Baiying Lei. Towards semantically faithful diffusion representation for generalizable retinal image segmentation. *IEEE Transactions on Medical Imaging*, pages 668–680, 2025.

[45] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2955–2966, 2023.

[46] Zhiwen Yang and Yuxin Peng. Gala-2.5d: Global-local alignment with 2.5d semantic guidance for camera-based 3d semantic scene completion in autonomous driving. *Chinese Journal of Electronics*, 35:817–828, 2026.

[47] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. *CoRR*, abs/1506.03365, 2015.

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

[49] Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. A tale of two features: Stable diffusion complements DINO for zero-shot semantic correspondence. In *Annual Conference on Neural Information Processing Systems*, pages 1–15, 2023.

[50] Tao Zhang, Ying Fu, and Jun Zhang. Deep guided attention network for joint denoising and demosaicing in real image. *Chinese Journal of Electronics*, 33:303–312, 2024.

[51] Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. Unleashing text-to-image diffusion models for visual perception. In *IEEE/CVF International Conference on Computer Vision*, pages 5706–5716, 2023.

Table 4: Comparing the massive activation suppression method (MAS) proposed in [11] on different tasks.

LSC Setting	SPair (a)	Horse-21
w/o LSC	34.68	30.35
w/o LSC + MAS	36.32	31.27
w/ LSC	41.10	36.32
w/ LSC + MAS	44.55	36.51

Table 5: Measuring the massive activation phenomenon without and with LSCs.

Setting	Channel Scale Median	Channel Scale Mean	Channel Scale Top5
w/o LSC	7.36	8.41	555.50, 386.00, 302.00, 282.75, 50.43
w/ LSC	7.32	10.82	273.75, 233.12, 217.00, 141.12, 91.56

A Appendix

A.1 Relation to Massive Activation Suppression

In the original [11], massive activations are found to be harmful to DiT feature quality. However, they mostly focus on correspondence tasks, which conventionally use training-free nearest neighbor for prediction, leading to a natural sensitivity to massive activations. Here, we evaluate the massive activation suppression method proposed in [11] on the semantic segmentation task, with the results in Table 4. It can be observed that the performance gain is much smaller on semantic segmentation than on semantic correspondence. This suggests that massive activations are not that harmful to tasks beyond semantic correspondence, especially when a trained downstream head that can refine the features is involved. Additionally, comparing the results on SPair (a), we can see that when adding massive activation suppression to *w/o LSC* and *w/ LSC*, their performance is both improved, with the gap between them remaining. This indicates that the effects of massive activation suppression and LSCs are roughly orthogonal.

As additional evidence, we also measure the massive activation phenomenon when LSCs are added. From Table 5, we can observe that LSCs have a slight effect on suppressing massive activations, but the effect is limited. This shows LSCs have additional, separate mechanics.

In general, massive activation as the cause of DiT feature degradation is not exhaustive.

A.2 DiT Models with Actual Resolution Changes

We wish to first clarify the two factors: LSCs and resolution changes. LSCs can be applied to *both* isotropic DiTs *and* DiTs with resolution changes. However, a DiT backbone with resolution changes has to utilize LSCs to preserve information that would otherwise be lost during downsampling. Hence, the two factors are only partially entangled, thanks to the transition and wide adoption of the transformer architecture [36, 50].

For comparison, we modify the DiT [27] code to implement resolution change in a minimal way, leading to two alignment options: (i) Down-Up-DiT-Param: this setting roughly aligns the parameter size to the DiT-XL/2 option we use in the main manuscript. (ii) Down-Up-DiT-Flops: this setting has more parameters, but the overall FLOPs roughly align to the DiT-XL/2 option. Besides the two alignment options, we also provide two downsampler/upsampler design variations: (i) Convolutional-based; (ii) Swin-Transformer-style [23] patch merging/expanding design. This leads to four variations in total, and we compare all of them against *w/ LSC* on generation and all six benchmark settings in Table 1.

The results are shown in Table 6. We can observe roughly two effects from resolution changes: (i) By forcing the backbone to actively utilize LSCs via resolution downsampling, the feature quality tends to improve. (ii) However, as the middle-layer features have a lower resolution, they are not that suitable for fine-grained tasks such as semantic segmentation, leading to a quality drop. Overall,

Table 6: Generation and discrimination performance of DiT models with actual resolution changes. Best and second marked as **bold** and underlined.

Model	FID	SPair (a)	SPair (b)	SPair (c)	Horse-21	CityScapes	Depth
w/ LSC	**14.39**	41.10	60.68	65.69	**36.32**	**51.55**	0.74
Param-Conv	21.31	38.62	55.83	65.04	31.62	45.90	0.76
Flops-Conv	19.09	39.04	58.31	67.03	32.05	45.84	**0.79**
Param-Swin	18.53	44.35	60.65	68.22	28.02	43.98	0.76
Flops-Swin	16.20	**45.96**	**62.52**	**69.56**	29.81	45.27	0.75

Table 7: Comparison of diffusion features extracted from prevalent large-scale text-to-image models on all six settings in Table 1. Best and second are marked as **bold** and underlined.

Model	#Params	SPair (a)	SPair (b)	SPair (c)	Horse-21	CityScapes	Depth
SD1.5	1B	62.79	77.78	86.04	57.90	64.10	0.86
SD2.1	1B	65.20	80.00	86.73	58.21	65.23	0.88
SDXL	3.5B	63.14	81.72	88.68	**63.20**	74.47	0.83
PGV2	3.5B	65.66	83.44	**92.31**	61.73	74.74	0.93
PixArt-Sigma	0.6B	60.23	72.77	81.72	55.28	26.17	0.77
HunyuanDiT	1.5B	67.88	80.50	90.02	62.93	**77.50**	**0.94**
SD3.5M	2.5B	56.75	68.72	65.93	51.77	59.38	0.79
SD3.5L	8B	65.09	76.86	79.16	51.68	55.13	0.78
Flux	12B	**72.62**	**85.93**	89.67	61.58	74.32	0.92

there are two effects that fall in opposite directions, and the general effect relies on what task the features are used for.

A.3 Preliminary Observation on Real-World-Scale Models

Here we provide a comparison table of real-world-scale U-Net/DiT diffusion models in Table 7. Since there are many confounders, such as model architectures, parameter sizes, or training recipes, we do not aim to draw very detailed conclusions from it. However, we can observe that most DiT models yield weaker features than U-Net models, and this gap extends across many different tasks, not restricted to semantic correspondence.

A.4 Effect of Backbone Prediction Type

The diffusion backbone can be trained to predict three different targets: ϵ, v, and x [21]. ϵ-prediction is the common choice for DDPM, while v-prediction has become more popular with the shift into flow matching. In comparison, x-prediction is only preliminarily explored [21]. In this subsection, we will show that the prediction target can affect the effect of adding LSCs to some extent. However, the prediction target is relatively superficial, while the peak position of the quality-layer curve is more essential.

In the main manuscript, we are using an ϵ-prediction code to train our models. Here, we switch to a v-prediction code and repeat the experiment in Table 2. This code is based on SiT [24], which is largely derived from the codebase of [27], using the same model architecture but switched to flow matching. Additionally, we propose a new LSC implementation as $x \leftarrow x + \gamma \cdot \text{LayerNorm}(\text{skip})$, with γ initialized to 0, because of training stability concerns. We will later denote this new implementation as the one with *residual connection*. The results are shown in Table 8. As for x-prediction models, we use the JiT [21] code, where we also use the LSC implementation with a residual connection. The results are shown in Table 9.

Before further discussion, we introduce two new concepts. (i) The peak position of quality-layer curves. To plot a quality-layer curve, we use SPair (a) performance as the y-axis and the layer index as the x-axis. This curve peaks at a certain layer, which can be at the middle layer or not, determined by complex factors. The quality-layer curves of models without LSCs are plotted in

Table 8: Comparing v-prediction DiTs without and with LSCs.

Model	FID	SPair (a)	SPair (b)	SPair (c)	Horse-21	CityScapes	Depth
w/o LSC	22.00	33.65	48.79	60.19	29.73	50.54	0.74
w/ LSC	**21.53**	**34.91**	**51.28**	**60.30**	**30.37**	**50.65**	**0.75**

Table 9: Comparing x-prediction DiTs without and with LSCs.

Model	FID	SPair (a)	SPair (b)	SPair (c)	Horse-21	CityScapes	Depth
w/o LSC	40.42	14.49	**26.38**	**45.16**	11.91	41.65	0.66
w/ LSC	39.49	**14.87**	**26.38**	43.40	**11.93**	41.80	**0.67**
skewed LSC	**38.35**	14.76	25.92	44.74	11.58	**42.21**	**0.67**

Figure 6, showing a clear shift of peak position. (ii) The topology of LSCs. By default, the LSC connections are symmetric, but it is legal to design a topology that is not. For instance, for a 28-layer DiT backbone, we could put the symmetry center between layer 10 and 11, instead of layer 14 and 15, creating a *skewed* topology. This effectively introduces a structural bias that tells the model to put its peak position at this skewed symmetry center.

From the results above, we can see that the feature quality is not significantly affected, but generation is overall better. Furthermore, the skewed LSC topology is better than the symmetric one when the vanilla peak position is already skewed. In addition, we have also tried to train SiT and JiT models using the same LSC structure (the one without residual connection) as used in the main manuscript. This structure tends to bring more significant representation quality enhancement and force the quality-layer curve to align with the LSC topology, but does not guarantee generation improvement. In general, we believe that whether and how the vanilla quality-layer curve skews is more essential, while the exact prediction target is relatively superficial. Nevertheless, how this interaction actually works remains an open question. For practical suggestions, we believe it is more beneficial if the LSC topology is designed to match the vanilla peak position of quality-layer curves.

A.5 Comparing DiT-XL/2, U-ViT, and Skip-DiT

To preliminarily support our hypothesis that LSCs contribute to better feature quality, we compare two open-source LSC-augmented DiT models from the literature:

(i) ImageNet 256x256 (U-ViT-H/2) of U-ViT [1].

(ii) DiT-XL/2-skip for Skip-DiT [6].

As a baseline with no LSCs, we also include the DiT-XL/2 (256) model for DiT [27]. We have confirmed that they share the same architecture hyperparameters except for LSCs. However, it should be noted that their training hyperparameters, such as training steps and batch size, are likely to differ, but the final FID scores roughly align.

From Table 10, we can make the following observations:

(i) Comparing U-ViT and DiT, U-ViT achieves overall better representation quality under a similar generation capability to DiT. We can attribute this gain to the LSC structure with some confidence, since the difference between the two models is LSCs *and* the training recipe. This motivates us to hypothesize that LSCs contribute to diffusion feature quality, but more evidence is still required to further exclude the effect of the training recipe.

(ii) Skip-DiT does not yield comparable results under certain settings. Note that, as argued as a main advantage, Skip-DiT has been through much fewer training steps before reaching a 2.29 FID score, which might cause its inner representations to be weaker. In general, this suggests the necessity of a from-scratch training experiment in a better-controlled environment to provide more concrete evidence.

(iii) If the diffusion backbone itself is fine-tuned during the discriminative benchmark (SPair (c)), the gap between non-LSC- and LSC-DiT models can become narrower and even reversed. This may be because the backbone is supervised entirely using the discriminative training

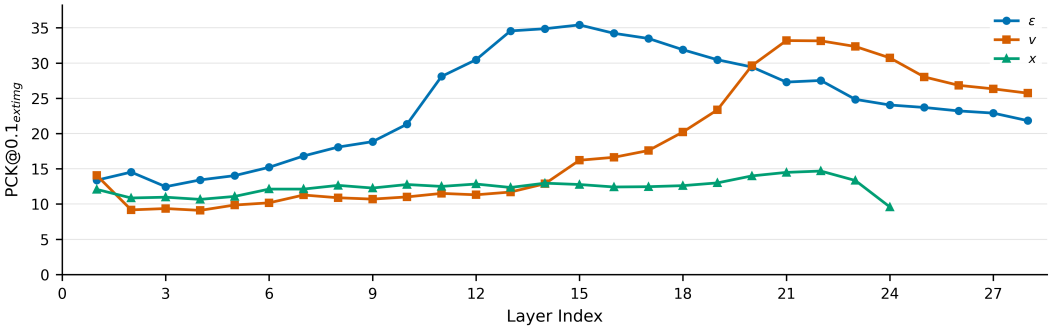


Figure 6: Quality-layer curves from three different prediction types.

Table 10: Comparing open-source DiT models with LSCs. Best results marked as **bold**. The FID scores are taken from the official reports, and the discriminative capability is evaluated using the six settings in Table 1.

Model	FID	SPair (a)	SPair (b)	SPair (c)	Horse-21	CityScapes	Depth
DiT	**2.27**	44.01	64.89	**77.62**	39.29	53.04	0.76
U-ViT	2.29	**53.53**	66.69	75.02	**40.53**	**53.97**	**0.84**
Skip-DiT	2.29	41.91	**68.29**	73.61	39.08	52.53	0.73

signal. This effectively discards the latter half of the backbone and no longer requires the final noise prediction capability. Hence, even for a DiT backbone with no LSCs, it can safely lower the diffusion noise intensity in middle layers, reaching a similar behavior to LSC-DiT models.

A.6 Mutual Information Probing Results

Considering that MINE can be sensitive, we repeat the experiment under two settings: (MINE hidden channel as 512, projection head channel as 256, training steps as 600) and (MINE hidden channel as 1024, projection head channel as 512, training steps as 1000). Moreover, for each setting, we repeat this experiment under three different seeds, each time re-extracting all features. The result from the first setting is reported in Figure 5, and the results from the second setting are reported in Figure 7. We can draw the same conclusions from the two experiments that our noise bypassing hypothesis could be supported.

Additionally, we also report the implementation details of the MINE probing. The input to the system is a 2D feature map of multiple channels. It is first projected into the MINE estimator dimension with a 1×1 convolutional layer. Then, we use average pooling to shrink it into a global vector per sample.

A.7 Two Corollaries of Noise Bypassing Hypothesis

A.7.1 Duo-Peak and Single-Peak Quality-Layer Curves

If we plot a figure with the layer index of a diffusion backbone as the x-axis and the understanding capability of the feature extracted from this layer (evaluated under a certain discriminative task) as the y-axis, we obtain a quality-layer curve. We observe that diffusion models without LSCs tend to manifest a quality-layer curve with a single peak, and models with LSCs tend to manifest a duo-peak curve, as shown in Figure 8.

Next, we provide a hypothesized explanation for this observation. Based on the three-section division of a diffusion backbone, features through the noise suppression section gradually grow in quality as they go into deeper layers in this section, as the noises are gradually suppressed, which forms the rising slope before the first peak. Then, in the noise prediction section, noise increases, causing the final descending slope. Importantly, the behavior of the target reconstruction section differs based on whether there are LSCs. According to another finding in [25], when trying to reconstruct an internal representation for the target image, activations capturing mostly abstract and coarse-grained

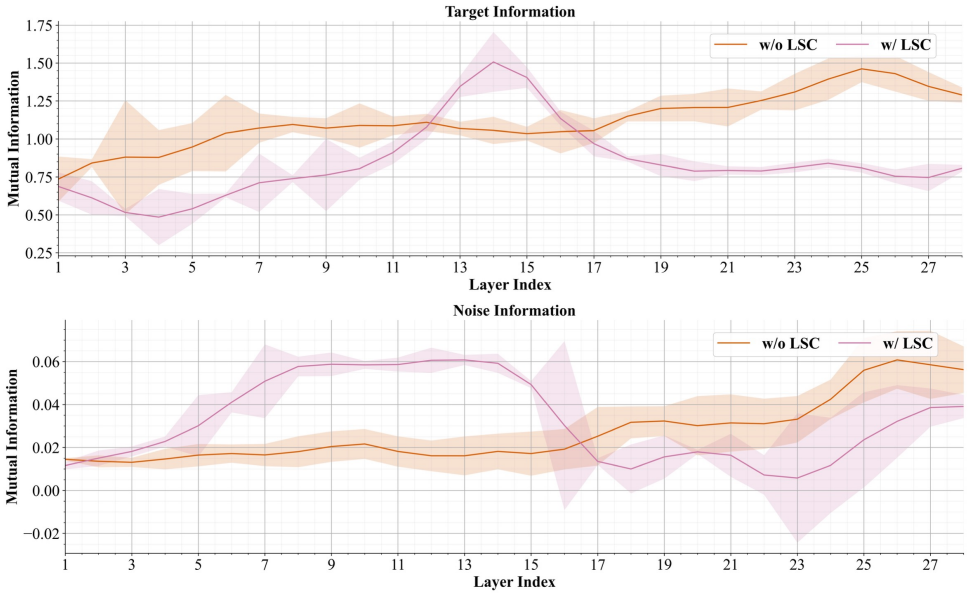


Figure 7: Mutual information between each layer's activation and noise/target information, measured using MINE, under the second setting.

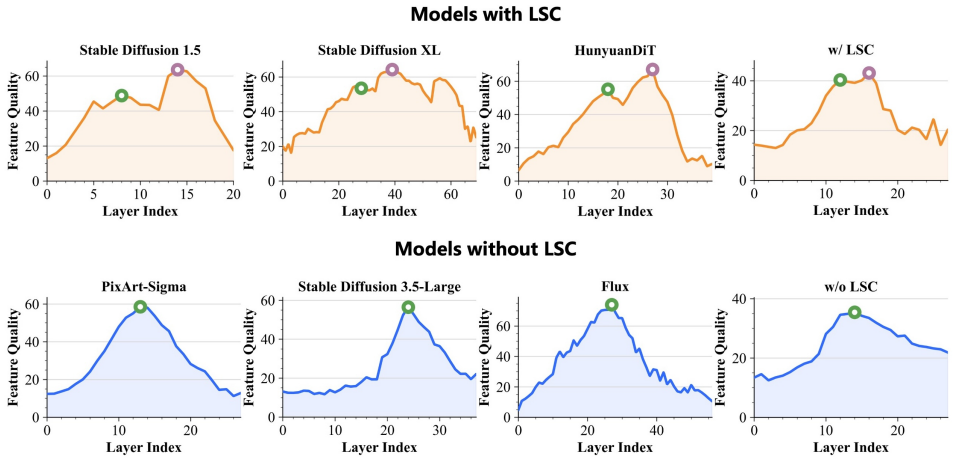


Figure 8: Quality-layer curves (x-axis for layer depth and y-axis for feature quality) of different diffusion backbones. Green dots mark the first peaks, and purple dots mark the second peaks. Quality is evaluated under the *Spair (a)* setting of Table 1.

information, such as positional and overall shape, are first formed, after which activations become more and more detailed. Notably, activations containing abstract and coarse-grained information, when used as a diffusion feature, tend to be inferior to the other features. Therefore, when there are LSCs, there will be a slight drop between the two peaks. In comparison, with no LSCs, the activations in the target reconstruction section would be relatively noisy, and the noises overrun the aforementioned detailed change, showing only one peak in the curve.

A.7.2 Upper and Lower LSCs

According to the noise bypassing hypothesis, LSCs in the upper part of the "U" shape have access to stronger noises, so they should be able to transport stronger noise information, making them more important compared to lower LSCs. Thus, if we keep only half of the LSCs, keeping the upper half should yield a better result than keeping only the lower half.

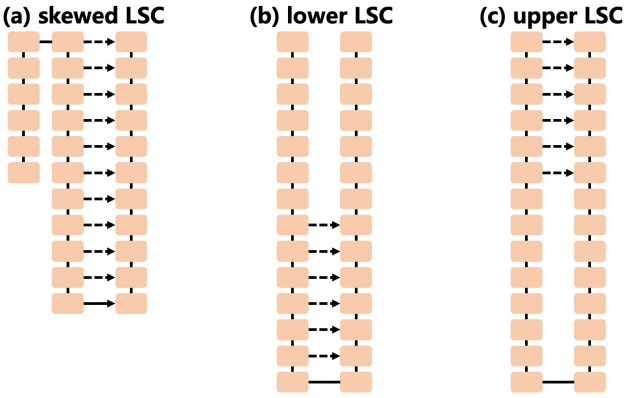


Figure 9: Different LSC designs to be tested in experiments. We use dotted lines to indicate LSCs. In this figure, the number of layers and position of LSCs *exactly* reflect the architecture of the modified DiT models.

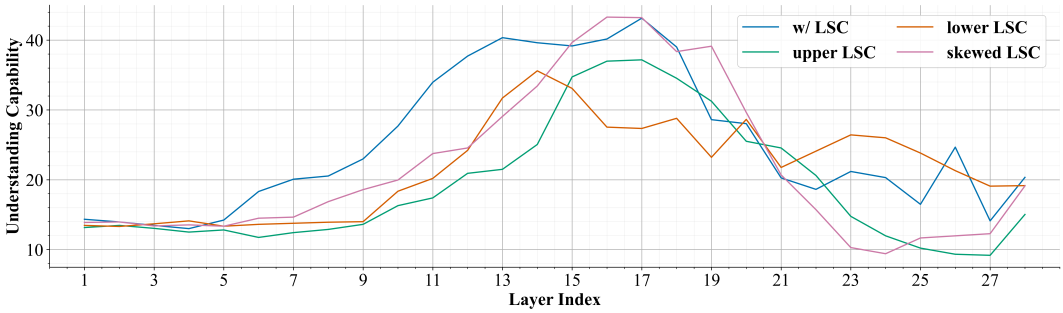


Figure 10: Quality-layer curves under three LSC variants. The quality is evaluated using *SPair (a)* setting.

A.7.3 Validation for the Two Corollaries

To validate the two corollaries, we conduct an additional controlled experiment. In this experiment, we design several special LSC structures, as depicted in Figure 9:

- (a) *skewed LSC*: The U-shape is skewed to one direction, possibly affecting the position of the second peak.
- (b) *upper LSC*: To test the relative contribution of the upper LSCs, we actually keep only the upper ones.
- (c) *lower LSC*: As a symmetric setting, only the lower LSCs are kept.

The quality-layer curves are visualized in Figure 10, and the quantitative results are shown in Table 11.

(i) First, we can observe from Figure 10 that the skewed LSC design indeed makes the second peak skewed as well, thus validating the causal relation between LSCs and the duo-peak observation.

(ii) Then, we can observe from Table 11 that the upper LSCs are indeed more important than lower LSCs: if we only keep part of the LSCs, keeping the upper ones leads to stronger downstream performance than keeping the lower ones. Nevertheless, the lower LSCs still matter, as both variants are worse than the previous complete LSC setting in Table 2.

In general, this experiment validates the two corollaries of the noise bypassing hypothesis, indirectly strengthening the evidence supporting this hypothesis.

Table 11: Discriminative performance evaluated on all six settings the same as Table 2. Best/second marked as **bold** and underlined.

LSC Setting	SPair (a)	SPair (b)	SPair (c)	Horse-21	CityScapes	Depth
skewed LSC	**42.33**	**58.93**	**66.76**	**37.08**	**51.16**	**0.74**
upper LSC	36.86	52.69	61.56	31.91	50.11	0.71
lower LSC	35.60	52.54	61.18	28.72	49.55	0.70

A.8 Attempts to Add LSCs to Pre-Trained DiTs

A.8.1 Method

We have tried several different methods to add LSCs to pre-trained DiTs, basically divided into two-stage and single-stage training types.

Two-Stage Method. Following the adaptation method in Skip-DiT [6], we divide the training into two stages. At the first stage, we freeze the backbone and only train LSC connectors until they converge, indicated by roughly recognizable generation results. Then, we unfreeze the backbone and train the entire model altogether. Ideally, during the second stage, the generative capability would quickly restore and improve, but we did not observe such behavior. Moreover, we notice that the features in the upsampling stage become very noisy. Therefore, we conclude that the two-stage fine-tuning method might be applicable only to certain diffusion models. We hypothesize that the first-stage training introduces too much degradation to the model, which is hard to correct in the second stage. To tackle the drawback of this method, we further propose a new single-stage method.

Single-Stage Method. Directly adding a vanilla LSC structure into a pre-trained DiT backbone will damage its original behavior and make single-stage fine-tuning hard, as we need to restore the generative capability from a low point. To tackle this, we propose the zero-init LSC design, which aims to make no impact on model behavior at initialization. In the zero-init LSC design, we put a residual connection in the LSC:

$$x = x + \text{Linear}(\text{Norm}(skip)) \tag{5}$$

where Norm is layer normalization, and Linear is a linear layer. The weight and bias of this linear layer are both initialized to zero, making the residual connection degenerate to an identity projection at initialization. Thanks to zero-init LSCs, we can safely fine-tune the entire model from the beginning using the standard diffusion training objective:

$$\mathcal{L} = \mathbb{E}_{(x_0,c)\sim\mathcal{D},x_1,t}||v'_\theta(z_t,t,c) - v_t||^2 \tag{6}$$

where v' is the DiT backbone augmented with LSCs, $v'_\theta(z_t,t,c)$ is the predicted velocity, and v_t is the ground truth.

Note that the initial state of the zero-initialized model is already at a local optimum, so we need to add some additional regularization to force the model out of it. Specifically, we extract features from the main activation residual flow in the noise suppression section and the target reconstruction section. First, we force the features from the target reconstruction to be clean of noises. Then, we encourage the features from the noise suppression section to be noisier. In this way, we hope the model can learn to shortcut the noise information through the newly added LSCs. We find that noises can be easily suppressed by a Gaussian blur, which creates a ground truth for this regularization term:

$$\mathcal{L}_{noise} = \text{MSE}(feat, \text{sg}(\text{GaussianBlur}(feat))) \tag{7}$$

where MSE is the mean squared error loss, $feat$ is the extracted feature, GaussianBlur is the Gaussian blur operation, and sg denotes the stop-gradient operator. We find that a weight of 0.001 / -0.001 is generally stable for section 2 / section 1, where the negative weight should be combined with gradient clipping to alleviate instability.

A.8.2 Training Strategy

We fine-tune PixArt-Sigma with the single-stage method to evaluate the viability. To construct the training data, we adopt a dataset titled DiffusionDB [39]. This dataset contains prompts and the corresponding images used by real diffusion users. We only use its prompt part, letting the original PixArt-Sigma generate images to obtain image-prompt pairs. In this way, we hope to retain the original generation capability with the fine-tuned model.

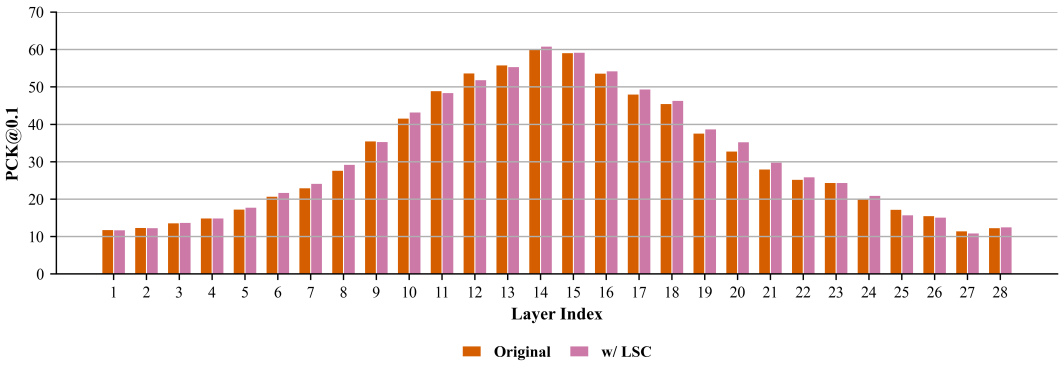


Figure 11: Per-layer feature quality from the original model and the model fine-tuned with LSC inserted. The quality is evaluated using PCK@0.1_{img} ↑.

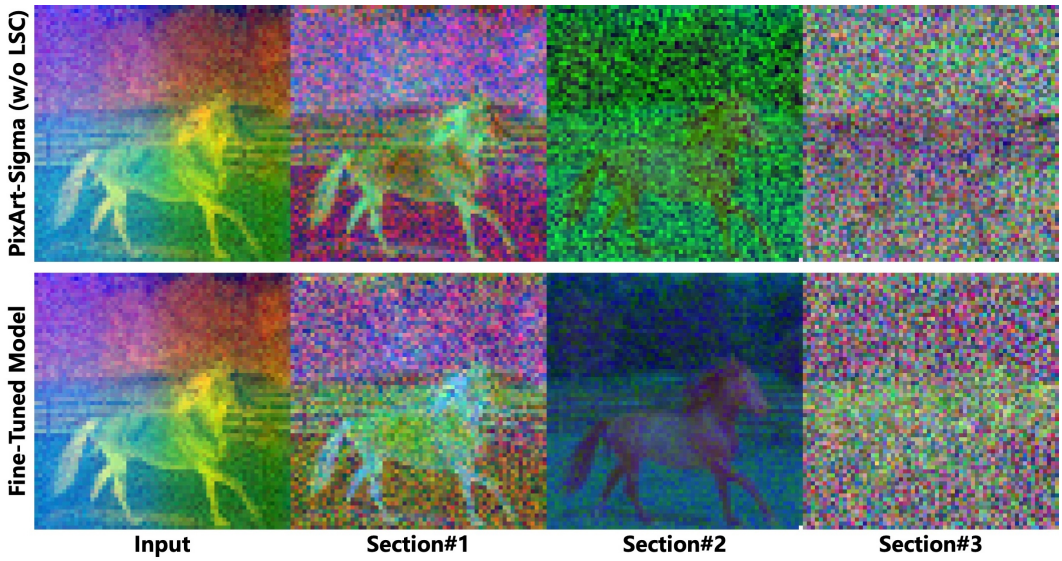


Figure 12: Feature PCA visualization of the original PixArt-Sigma model and the fine-tuned version with LSCs.

A.8.3 Results

We compare the feature quality, evaluated using *Spair (a)* settings in Table 1, of both the original and fine-tuned models in Figure 11. We can notice that:

(i) Feature quality increases, but only slightly.

(ii) The overall quality-layer curve is roughly the same as before, showing that the model's inner behavior is not fundamentally changed to manifest the duo-peak curve.

In addition, we provide some feature PCA visualization in Figure 12. We can observe that the additional noise regularization term can actually toggle the noise strength in different parts of the backbone to some extent, but the result is not satisfying enough. For example, in the target reconstruction section, the features contain less noises than the original PixArt-Sigma model, but still more than HunyuanDiT.

In summary, we believe that the strong path dependence of DiTs' learned representations makes it tough to fundamentally change the behavior of pre-trained DiT models. To be specific, the backbone model has already learnt to transport both noise and target information in the same activations, which might be the optimal behavior when there are no LSCs. Such an ability might conflict with the ability to separately transport the information of two aspects in different paths, making fine-tuning

Table 12: Parameter sizes of different LSC designs, compared to the original *w/o LSC*.

LSC Type	Parameter Size	Percentage w.r.t. w/o LSC
w/o LSC	675,129,632	100.0%
w/ LSC	709,709,216	105.1%
upper LSC	691,089,440	102.4%
lower LSC	691,089,440	102.4%
skewed LSC	701,729,312	103.9%

particularly difficult. Therefore, fundamentally changing how the backbone functions via fine-tuning might require as much training as training from scratch, or even more. Currently, we believe that, to effectively fuse LSCs into a DiT, it is likely that the model needs training from the start, and direct retrofitting is likely to be hard in nature. Nevertheless, it is not impossible that, with a better fine-tuning strategy, we could really add LSCs to a pre-trained DiT model.

A.9 Computational Cost of LSC

Computational cost is essential nowadays [42]. We provide the parameter size of different LSC settings in Table 12. We can observe that, parameter-wise, LSCs only introduce a slight overhead, compared to their huge bonus to both feature quality and generation. Additionally, for more practical DiT architectures with more complex designs, this overhead will become even smaller. In our experiments, the running time of different LSC designs is almost identical.

However, it is possible that LSCs might conflict with some acceleration techniques based on parallelism by introducing additional shortcut computation. This may also be one of the reasons why LSCs are not commonly used in today's DiTs. Nevertheless, there are also methods that utilize LSCs to accelerate diffusion inference, such as caching [6]. By adopting LSCs in future DiT designs, such methods would be enabled for application, which might make LSCs in DiT more appealing efficiency-wise.

A.10 Exact Feature Extraction Specification for DiTs

In this section, we specify the exact optimal feature extraction spots and timesteps for the DiTs evaluated in Table 7. The extraction spots and timesteps are established following the pipeline in Section 3, and once the optimal choice is decided, it is reused across all experiments. The layer index starts at 0, and the timestep is given based on a max timestep of $T = 1000$.

PixArt-Sigma ($t = 100$):

(i) Cross-attention query at layer 13.
(ii) Output of self-attention at layer 13.
(iii) Output of the 2nd normalization at layer 14.

HunyuanDiT ($t = 50$):

(i) The normalized cross-attention query at layer 27.
(ii) Self-attention query at layer 27.
(iii) Output of the 3rd normalization at layer 26.

SD3.5M ($t = 300$):

(i) Output of the 1st attention at layer 9.
(ii) Output of the 2nd normalization at layer 9.
(iii) Query of the 1st attention at layer 11.

SD3.5L ($t = 400$):

(i) Output of the 2nd normalization at layer 23.

(ii) Output of the 1st attention at layer 23.
(iii) Value of the 1st attention at layer 24.

Flux ($t = 200$):

(i) Attention output at layer 27.
(ii) Attention key at layer 29.
(iii) Attention output at layer 25.

w/o LSC ($t = 50$):

(i) Output activation at layer 13.
(ii) Output activation at layer 14.
(iii) Output activation at layer 15.

w/ LSC ($t = 50$):

(i) Output activation at layer 14.
(ii) Output activation at layer 15.
(iii) Output activation at layer 16.

upper LSC ($t = 50$):

(i) Output activation at layer 15.
(ii) Output activation at layer 16.
(iii) Output activation at layer 17.

lower LSC ($t = 50$):

(i) Output activation at layer 13.
(ii) Output activation at layer 14.
(iii) Output activation at layer 15.

skewed LSC ($t = 50$):

(i) Output activation at layer 15.
(ii) Output activation at layer 16.
(iii) Output activation at layer 17.

In addition, the U-Net features are extracted at the optimal spots specified in [25].

Note that the specification above is not applicable to quality-layer curves. For quality-layer curves, we extract features from similar spots of each layer at the optimal timestep as stated above. For example, we can extract the basic block output of each layer to plot the quality-layer curves.

A.11 Experimental Details

Semantic Correspondence [12]: This task aims to predict the corresponding keypoint in the target image given a keypoint in the source image, where the two keypoints should be semantically similar. The dataset used is SPair-71k [26]. PCK@0.1_{img}($\uparrow$) and PCK@0.1_{bbox}($\uparrow$) are commonly used metrics, following the widely adopted protocol reported in [26]. These two metrics measure the percentage of correctly predicted keypoints, where a predicted keypoint is considered to be correct if it lies within the neighborhood of the corresponding annotation with a radius of $0.1 \times max(h, w)$. For PCK@0.1_{img}/PCK@0.1_{bbox}, h, w denote the dimensions of the entire image/object bounding box, respectively. In Setting#2, we train a small downstream network, which is a two-layer convolutional network. In the other two settings for this task, we directly perform nearest neighbor computation, so no downstream model is required. If training is required, we use a learning rate of 1e-4 for the

downstream network and 1e-5 for the diffusion backbone, and the optimizer is AdamW for smaller models and AdamW8bit for larger diffusion models (SD3.5L and Flux). The prompt used for feature extraction is "a highly realistic photo that may contain an aeroplane, a bicycle, a bird, a boat, a bottle, a bus, a car, a cat, a chair, a cow, a dog, a horse, a motorbike, a person, a plant within a pot, a sheep, a train, or a tv monitor".

Semantic Segmentation: This task aims to classify pixels into semantic categories [46]. The dataset used is CityScapes [8]. We use mIoU metric, which is the mean over the IoU performance across all semantic classes [12]. For each image, IoU (Intersection over Union, ↑) is defined by #(overlapped pixels between the prediction and the ground truth) / #(union pixels of them). In this setting, we train a relatively large downstream network, which is UPerNet [43]. We use a learning rate of 0.00016 and AdamW optimizer. The prompt used for feature extraction is "An urban street scene with multiple lanes, various buildings, traffic lights, cars in the lanes, and pedestrians, highly realistic.".

Label-Scarce Semantic Segmentation [2]: This is a special setting for semantic segmentation, where the labeled training data is very scarce ($\tilde{3}0$ sample-annotation pairs). The dataset used is collected in [2], and we experiment on its Horse-21 subset, the data of which is sourced from LSUN [47]. In this setting, we train 10 pixel-classification MLP models and aggregate their predictions by choosing the majority. The learning rate is 0.001, and the optimizer is Adam. The prompt used for feature extraction is "a horse in the field running, high quality, photorealistic".

Depth Estimation [4]: This task aims to regress the depth value, *i.e.*, the distance of each pixel from the camera. The dataset used is NYU Depth v2 [31]. We use d1 as the evaluation metric. It computes the percentage of correctly predicted pixels, where a pixel is counted as correct if the difference between its value and the ground truth value does not exceed 25%. In this setting, we add a downstream model consisting of two regular convolutional layers and two 1×1 convolution layers, training this model along with the diffusion backbone itself. The diffusion backbone is trained with a 10-times smaller learning rate than the downstream network (1e-4/1e-5), and the optimizer is AdamW8bit. The prompt used for feature extraction is "a photo, focusing on the depth of the scene".

A.12 Computation Resources

To train the DiT models under different settings from scratch, we use $2\times$A100 GPUs for each experiment, which takes around five days per training run. For the other experiments, such as evaluating these DiT models or the real-world-scale experiment, we only need a single 3090 or L20 GPU per experiment. Whether we use 3090 or L20 depends on the parameter size of the pre-trained DiT model and the number of trainable parameters in the backbone. Nevertheless, we need to point out that we have multiple cards to run multiple experiments in parallel, without which this study would be somehow hindered.

A.13 Licenses

- SPair-71k: Available at https://cvlab.postech.ac.kr/research/SPair-71k/.
- Label-scarce segmentation datasets (sourced from LSUN): Available at https://github.com/fyu/lsun.
- CityScapes: Custom, at https://www.cityscapes-dataset.com/license/.
- NYU Depth v2: Available at https://cs.nyu.edu/~fergus/datasets/nyu_depth_v2.html.
- Stable Diffusion v1.5: The CreativeML OpenRAIL M license, available at https://huggingface.co/stable-diffusion-v1-5/stable-diffusion-v1-5.
- Stable Diffusion v2.1: CreativeML Open RAIL++-M License, available at https://huggingface.co/Manojb/stable-diffusion-2-1-base.
- Stable Diffusion XL: CreativeML Open RAIL++-M License, available at https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0.
- Playground v2: Playground v2 Community License, available at https://huggingface.co/playgroundai/playground-v2-1024px-aesthetic.
- PixArt-Sigma: CreativeML Open RAIL++-M License, available at https://huggingface.co/PixArt-alpha/PixArt-Sigma-XL-2-1024-MS.

- Stable Diffusion v3.5: the Stability Community License, available at https://huggingface.co/stabilityai/stable-diffusion-3.5-medium and https://huggingface.co/stabilityai/stable-diffusion-3.5-large.
- Flux: flux-1-dev-non-commercial-license, available at https://huggingface.co/black-forest-labs/FLUX.1-dev.
- HunyuanDiT: tencent-hunyuan-community, available at https://huggingface.co/Tencent-Hunyuan/HunyuanDiT.
- ImageNet: Custom terms of access (non-commercial research and educational use only), available at https://image-net.org/.
- DiffusionDB: CC0 1.0, available at https://huggingface.co/datasets/poloclub/diffusiondb.
- DiT: CC-BY-NC 4.0, available at https://github.com/facebookresearch/DiT.
- SiT: MIT License, available at https://github.com/willisma/SiT.
- JiT: MIT License, available at https://github.com/LTH14/JiT.
- U-ViT: MIT License, available at https://github.com/baofff/U-ViT.
- Skip-DiT: Apache-2.0 (code, available at https://github.com/OpenSparseLLMs/Skip-DiT); CC-BY-NC 4.0 (model weights, available at https://huggingface.co/GuanjieChen/Skip-DiT).

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We tried our best to make the abstract and introduction reflect the contributions and scope of this paper.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitations mainly in the conclusion section (Section 6) and explicitly add the phrasing "limitation" in the section title.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: Although, in our own opinion, this work is more like a foundational paper than a method contribution, it is more empirical than theoretical.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We have included many details. If there should be any missing information, it can be easily clarified with the code release.

   Guidelines:

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

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

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

Answer: [No]

Justification: We plan to release the code if this work is accepted.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: We provide many experimental details in both the main body and the appendix.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [No]

Justification: Actually, the answer is partially yes and partially no. For the mutual information measuring experiment, we provide standard variance (in Figure 5), since this method is known to be sensitive. For the other experiments, we do not repeat them under different seeds, as it would be too computationally expensive and cost too much time.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: See Section A.12.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We conform the NeurIPS Code of Ethics in this study.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This is a general AI study, and its societal impact does not go beyond that of a standard AI paper.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This work is not likely to be exploited for misusing.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: See Section A.13.

    Guidelines:

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

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: We do not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

    Answer: [N/A]

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

    Guidelines:

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

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

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

    Answer: [N/A]

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

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: We use LLMs only for writing assistance.

Guidelines:

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