WE HAVE BEEN INTERPRETING DIFFUSION TRANSFORMER BACKBONE BEHAVIORS WRONG

Anonymous authors
Paper under double-blind review

ABSTRACT

During a network forward call, diffusion backbones naturally produce many intermediate activations. If we evaluate each activation against an identical downstream benchmarking protocol, we can plot an inverted-U-shaped curve as the depth profile: the activation's performance goes up first and then down as we go into deeper layers. This is common knowledge in the diffusion model community, which inspires many applications including the widely-adopted REPA. However, the common interpretation of this depth profile may be inaccurate and incomplete: (i) The cause of this curve has long been treated as signal granularity changing from coarse to fine, but there has been little evidence in DiT backbones. (ii) The community has been mainly studying this phenomenon in a static way, without investigating how it changes across different training setups. In this work, we show a different mechanism picture for the profile, where the prediction target (*e.g.*, ϵ, u, x_0) sits at the center. The semantically meaningful signal is only an intermediate by-product for predicting the target, while the prediction target is the real goal for the backbone. Furthermore, the choice of prediction targets as part of the training setup can predictably affect the curve's shape. With this updated interpretation, many applications of this depth profile can be better grounded and predicted. As an example, we show how to predict the near-optimal placement of REPA based on the prediction target of the training setup.

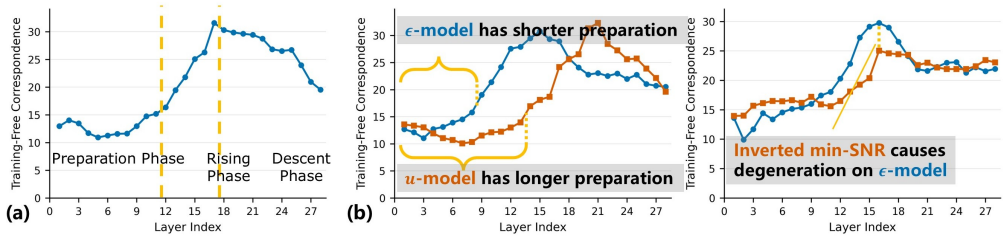

Figure 1: (a) The inverted-U-shaped curve, with layer index as x-axis and correspondence performance (or other downstream task performance) as y-axis, is common knowledge in the diffusion community. (b) Two representative *new* backbone behaviors among our findings, where *the training setups* determine the depth profile. Left: prediction target determines the length of the preparation phase. Right: prediction target can lead to degeneration under certain settings, indicated by the significantly lower peak height.

1 INTRODUCTION

Diffusion models are a trending type of generative model, mainly used for, but not restricted to, image generation, consisting of a scheduler and a backbone network (Ho et al., 2020; Dhariwal & Nichol, 2021; Rombach et al., 2022; Podell et al., 2023; Wu et al., 2025). As a neural network, the diffusion backbone produces many intermediate activations during a forward call, which can be denoted as diffusion features (Baranchuk et al., 2022; Xu et al., 2023; Zhao et al., 2023; Tang et al., 2023a; Zhang et al., 2023; Luo et al., 2023; Meng et al., 2024; Gan et al., 2025; Chen et al.,

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

2025). Evaluating diffusion features' performance on downstream vision benchmarks can be used as a *probe* to understand diffusion backbone behaviors (Yu et al., 2025). One important instance of these behaviors is the depth profile: the performance first goes up with layer depth and then down, forming an inverted U-shape, as shown in Figure 1(a). To name its importance, REPA (Yu et al., 2025), optimal layer selection (Zhou et al., 2024; Luo et al., 2023), and post-hoc normalization (Gan et al., 2025), are all directly or indirectly based on the interpretation of the depth profile.

However, the current interpretation of the depth profile is potentially inaccurate and incomplete. On one hand, the popular opinion is that the semantic signal gradually changes its granularity from coarse to fine, and the peak is at the exact position where the granularity best suits the benchmarking task (Baranchuk et al., 2022; Luo et al., 2023; Yu et al., 2025). However, this is not supported by strong evidence after the community moves to DiTs, where there are no structural biases to force such a behavior as in U-Nets. On the other hand, previous interpretations are mostly static, focusing on one model and overlooking the potential influence of the training setups on the depth profile. Therefore, the current application of the depth profile property relies on a foundation that is still partially black-box. In this work, we provide a more complete and likely more accurate picture, where the *prediction target* sits at the center. Here, prediction target means what type of signal the backbone actually predicts: not all diffusion backbones predict noise ϵ, as they can also predict u[1] or x_0 instead, which can all be utilized for denoising with proper formulas (Li & He, 2026).

We reveal that the whole depth profile can be divided into three phases, as shown in Figure 1(a), and that the real goal of the backbone is to predict the target, with a semantically meaningful inner representation as only a by-product. (i) **Preparation phase**: The semantic signal helpful for the downstream task has *not* started to form. Instead, the model extracts ϵ and/or x_0 from the residual[2] and stores them back into subspaces for later use. (ii) **Rising phase**: A highly abstract and high-level representation x_s of the clean image gradually forms, peaking at a certain layer, which is exactly the semantic signal for the downstream task. (iii) **Descent phase**: The model combines the semantic x_s with ϵ and/or the input-space x_0 to encode the prediction target. The semantic signal is therefore still within the residual stream, with a new signal corresponding to the prediction target being added to the residual stream. Hence, at least one major cause for the curve going down is the new signal overrunning the semantic signal, beside the granularity change.

Furthermore, we find that the training setups can affect, or even *determine*, the depth profile in a way that the prediction targets play a vital role, as shown in Figure 1(b). One example corollary from the findings above is that the length of the preparation phase is determined by the prediction target, where a more *hybrid* target leads to longer preparation than a *purer* target. Here, we treat targets like x_0 or ϵ as pure, and targets like $u = \epsilon - x_0$ as hybrid, as it mixes the two pure targets into one. Another example is that highly pure targets can lead to *degeneration* when paired with certain loss weighting. Both corollaries align well with the actual observations.

Lastly, to exemplify the practical implication of our findings, we show how to predict the near-optimal placement for REPA (Yu et al., 2025) based on our findings. We even make an aggressive prediction that the REPA alignment layer should be gradually moved to earlier layers as the training goes on, which is also validated. These demonstrations serve as falsification tests of the mechanism interpretation rather than as method contributions. Our contribution is as follows:

- We propose and validate a more complete and likely more accurate mechanism interpretation for the diffusion backbone depth profile, with prediction targets playing a vital role.
- We reveal how training setups, especially prediction targets, determine the depth profile.
- Practically, our interpretation makes certain practice ambiguities less black-box and more predictable, alleviating the need for exhaustive search or intuition.

2 Related Work

Diffusion Feature. Diffusion feature has become an acknowledged research area through years of advancements. Early on, the concept of using diffusion backbones as feature extractors was vali-

[1]The so-called v in DDPM literature (Salimans & Ho, 2022) and flow matching literature (Lipman et al., 2023) is different, and we define $u = \epsilon - x_0$ in place of the ambiguous v.

[2]For a residual connection $x \leftarrow x + \text{Module}(x)$, x is residual and $\text{Module}(x)$ is increment.

dated by pioneer studies such as Baranchuk et al. (2022); Xu et al. (2023); Zhao et al. (2023); Tang et al. (2023a). In such early studies, the depth profile of diffusion backbones is already observed and utilized to select the near-optimal layer to boost downstream performance. However, this selection process relies on heavy ablation or intuition, so researchers have developed some techniques for the selection, such as an automatic layer selector (Tang et al., 2023a; Zhou et al., 2024). This shows that the depth profile still remains partially black-box in nature. Additionally, whilst most diffusion feature studies focus on U-Net-based backbones, new findings on DiT backbones are emerging (Gan et al., 2025; Helbling et al., 2025).

Probing Diffusion Models. There are different ways to probe a diffusion model and gain a better understanding of its behaviors and properties. (i) Diffusion feature, as introduced before, can serve as a probe. (ii) Another common probe is specifically the attention maps within cross-attention layers, calculated via the matrix multiplication of Q and K (Tang et al., 2023b). This could help interpret how a diffusion backbone aligns textual semantics to pixels (Meng et al., 2026). (iii) Sparse Auto Encoder (SAE) is a recently emerging tool for diffusion model probing (Surkov et al., 2025; Tinaz et al., 2025). (iv) REPA (Yu et al., 2025) uses DINOv2 features as an external alignment target and measures how well each diffusion feature aligns with them.

Our work lies in the intersection of the two aforementioned directions: we use diffusion feature on downstream vision tasks as a way to probe diffusion models. In this study, we do *not* aim to improve diffusion feature performance, but only to better understand model behavior and mechanisms.

3 Preliminary & Notation

Different from the common formalization of DDPM (Ho et al., 2020; Song et al., 2021) or flow matching (Lipman et al., 2023), we will use a more general way to describe the diffusion theoretical framework. A diffusion model consists of a scheduler and a backbone model (Net). The scheduler is used to produce training data at training time and apply the model prediction to the data point at inference time. To construct one data sample, we need an input image x_0, a sampled random noise ϵ, a timestep t, and optionally a condition c. In this manuscript, we use $t = 0$ to denote the clean end and $t = 1$ for the noisy end. Note that the input image is commonly compressed into VAE space (Rombach et al., 2022), but we do not explicitly use another notation such as z_0 for this, for simplicity. With x_0 and ϵ, the scheduler constructs an $x_t = \alpha(t)x_0 + \beta(t)\epsilon$ as linear combination. Next, the scheduler also constructs a ground truth GT $= \alpha(t)'x_0 + \beta(t)'\epsilon$, where GT and x_t can use different linear combination recipes. The formulation of the ground truth is what we call *prediction target*, as it is exactly what the backbone learns to predict. Thus, we obtain a data point (x_t, c, t, GT). Finally, the training objective is $L = ||\text{Net}(x_t, c, t) - \text{GT}||^2$. When generating, the scheduler iteratively calls the backbone at different t. When extracting diffusion features, however, we mimic the *training* procedure, meaning that we only feed x_t into the backbone once at a selected t.

4 Probing Setups

4.1 Aligned Training Framework

4.1.1 Alignment

The motivation for the aligned training framework is to align most factors and only expose certain ones of interest within the training setups. We implement the following alignment designs:

Cross-Target Loss Weighting Alignment. Different prediction targets can be transformed into each other, but this process produces certain loss weighting terms that are functions of SNR (signal-to-noise ratio) (Hang et al., 2023). For example, $||\epsilon - \hat{\epsilon}||^2 = \text{SNR}(t)||x_0 - \hat{x_0}||^2$, with the weighting term being $f(\text{SNR}) = \text{SNR}$. To align this difference, we cast all prediction targets into x_0-space and cancel the derived coefficient. For example, we can multiply ϵ loss with $\frac{1}{\text{SNR}}$. Furthermore, for more stable training, we apply min-SNR weighting (Hang et al., 2023) after the alignment, which emphasizes cleaner samples. A visualization of the weighting curve in the $(u = \epsilon - x_0)$-space can be found in Figure 2(a).

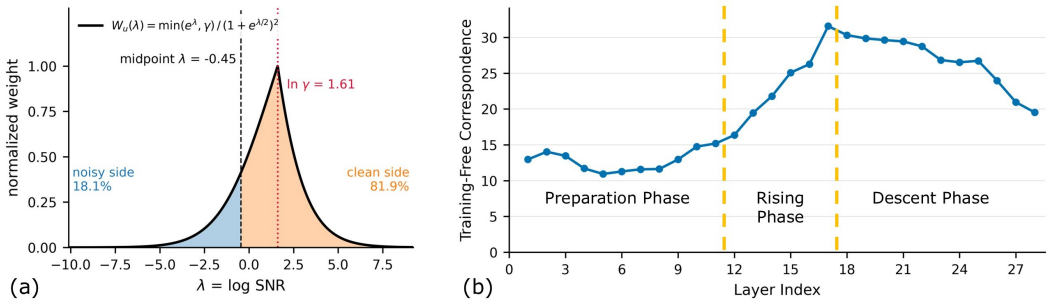

Figure 2: (a) A visualization of the default min-SNR weighting ($\gamma = 5$) in the u-space. (b) A typical depth profile of diffusion backbones. The curve is obtained from FM+u model with clean weighting.

SNR Sampling. Conventionally, diffusion training setups sample timestep t uniformly. However, considering the different contours of sampling paths, uniform t actually leads to non-uniform SNR, which can effectively be seen as implicit loss weighting (Kingma & Gao, 2023). Hence, we sample uniformly in the logSNR space instead, and the sampling space is [-10.12, 9.21] (which is why the midpoint in Figure 2(a) is -0.45).

Training Steps. We train all models for 200k steps, which strikes a balance between time cost and model convergence, unless otherwise stated.

Backbone Architecture. We use a backbone derived from DiT-XL/2 (Peebles & Xie, 2023) and SiT-XL/2 (Ma et al., 2024) as the main backbone, unless otherwise stated. Specifically, the timestep embedding layer is modified such that both DDPM and flow matching paths use the same embedding protocol. In this way, the backbone architecture is entirely aligned, only exposing the difference in training setups. The DiT backbone, compared to U-Net backbones, is cleaner with fewer confounding factors such as resolution changes, making it suitable for model probing studies.

Probe-Time SNR Alignment. It is a known property that the depth profile changes with the timestep for probing (Baranchuk et al., 2022; Zhou et al., 2024; Luo et al., 2023). Furthermore, the timestep definition differs between DDPM and flow matching. Hence, we do not probe models at matched timesteps, but matched SNR instead.

4.1.2 Exposed Factors

Prediction Targets. We include practical targets for diffusion training: ϵ, u, and x_0. Furthermore, we design a diagnostic target $a(\theta) = cos\theta\epsilon - sin\theta x_0$, where θ is a hyperparameter fixed for the entire training process. This allows testing finer-grained prediction targets.

SNR Weighting. The base min-SNR weighting emphasizes the cleaner samples, so we denote it as the *clean* weighting. We can *replace* the base min-SNR weighting with two alternative implementations: (i) a mirrored distribution to the curve shown in Figure 2(a) (the *noisy* weighting), and (ii) a curve that is completely symmetric in the logSNR sampling space (the *neutral* weighting).

Sampling Path. We test the classic DDPM path (denoted as variance-preserving, VP) and the straight flow matching path (denoted as FM).

4.2 Probes

Downstream Task Performance. (i) Following DIFT (Tang et al., 2023a), we use **training-free semantic correspondence** on SPair-71k dataset (Min et al., 2019) as the first probe in our study, detecting semantically rich representations. As a training-free probe, it can only detect signals that are directly interpretable without decoding. (ii) To complement its capability, we also optionally **add one linear convolutional layer** to the probe, giving it *linear* decoding capability. Note that only the decoding capability changes, but the signal granularity requirement, which is determined by the semantic correspondence task, does not change. (iii) We can also use **post-hoc transforms** on the features before correspondence computation, including whitening (Su et al., 2021) (roughly similar to normalization) and transplantation (Mikolov et al., 2013) (a free-form linear transform

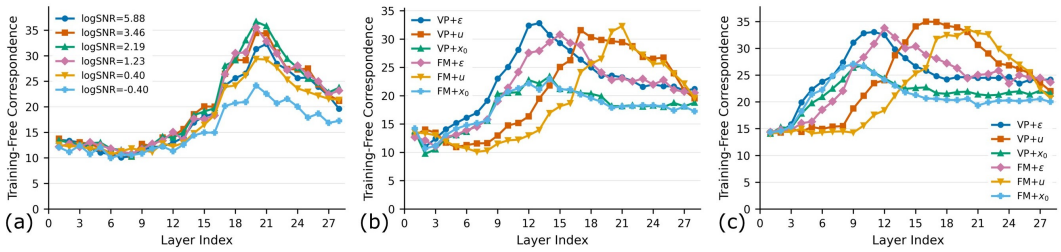

Figure 3: (a) Probing the same FM+u model at different SNR yields highly similar depth profiles. (b) Basic training-free correspondence probe results on different sampling paths. (c) Whitening correspondence probe results on different sampling paths.

that projects a feature into the linear space of another given feature). All these probes are measured using logSNR=5.88 unless otherwise stated.

Other Probes. (i) **Linear regression** (R^2) predicts a target signal from an input signal via linear training or closed-form solutions (Huh et al., 2024; Alain & Bengio, 2017). (ii) **Separability probe** (Elazar et al., 2021) removes a direction from a given feature, but with the restriction of minimizing the impact on other given features that should be protected. The cleaner the target direction can be removed under this restriction (indicated by *lower* remaining R^2), the more separable these features are. (iii) **Geometry probe** detects the relative norm and the angle between two features. These probes are measured under logSNR=0 unless otherwise stated.

Causal Intervention. We can manipulate certain features during inference and observe the influence on the outcome, including MSE loss and the main probe's curve. Such manipulation can be done in two ways: (i) Simply removing a certain direction from a feature; (ii) Replacing a certain direction within one feature with a random noise of the same mean and variance as the replaced direction.

5 Mechanism Findings

As shown in Figure 2(b), the common depth profile of diffusion backbones is in an inverted U shape. We divide this shape into three phases: preparation phase, rising phase, and descent phase. *The overall picture* is that the actual task for the diffusion backbone is to regress the prediction target GT $= \alpha(t)'x_0 + \beta(t)'\epsilon$, and the forming of an inner semantic representation of the clean image is only a *by-product*. As shown in Figure 4(a): (i) During the preparation phase, the model extracts x_0 and ϵ from the residual and stores them back into subspaces within the main residual. (ii) During the rising phase, a semantic, abstract inner representation of the clean image, denoted as x_s ("s" for semantic), is formed. (iii) During the descent phase, the model combines the semantic x_s and the VAE-space signal x_0, ϵ to encode the prediction target and write it back into the residual, waiting to be decoded by the final linear layer.

Before going into details, we find that the depth profile is mainly static with respect to the inference-time SNR choice (except for the height of the peak), almost solely determined by the training dynamics (Figure 3(a)). The peak height difference is related to the timestep profile, where a noisier timestep yields coarser semantics and a cleaner timestep yields finer-grained features (Tang et al., 2023a). In addition, we find that the difference between sampling paths is relatively slight. Considering the difference caused by prediction targets and loss weighting is already too heavy, we will neglect the difference between sampling paths for now and leave it for future work (Figure 3(b,c)).

5.1 Preparation Phase

Based on the overall picture, the preparation phase prepares x_0 and ϵ by storing them into subspaces, and the prepared signals are mainly consumed *not* in the rising phase but in the descent phase.

First, we locate how the preparation is done. Within each layer of a DiT backbone, there are an attention layer and an MLP layer. With the geometry probe results in Figure 4(b), attention increments align better with the residual's direction but have smaller norms, while MLP increments are

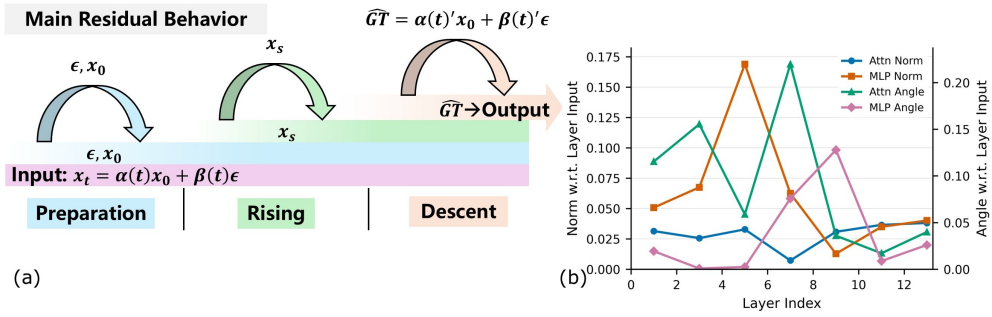

Figure 4: (a) An illustration of what each phase writes into the main residual and *remains there afterward*. (b) Geometry probe results of FM+$a45$ model, angle measured in cosine. Each feature is measured relative to the layer input.

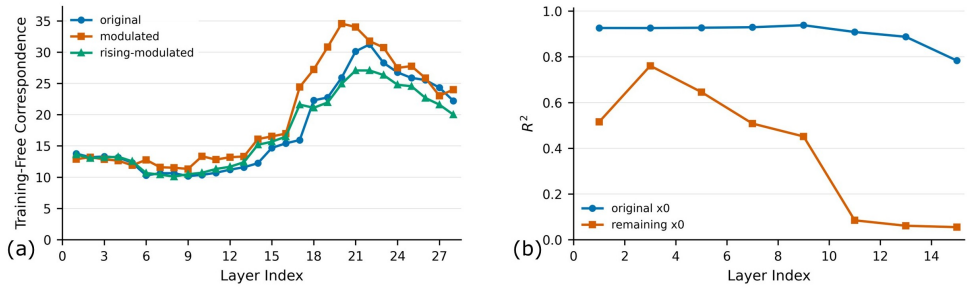

Figure 5: (a) Training-free correspondence probe results from the original and the modulated FM+$a45$ models. (b) Separability of FM+$a45$ model's MLP increments, measured using the remaining x_0 signal after deleting x_0 from the feature while minimizing the damage to ϵ.

more orthogonal to the residual but have larger norms. Then, we selectively zero out the attention increments or the MLP increments, finding that zeroing out the MLP increments leads to a more significant performance drop (MSE loss=0.3863 for attention and MSE loss=0.4176 for MLP). This shows that MLP increments are more important to the preparation behavior.

Then, to validate that the prepared information is mainly consumed by the descent phase, we zero out MLP increments in the preparation phase and test both prediction loss and correspondence performance. The prediction loss worsens (from 0.3852 to 0.4176), but the correspondence probe even improves (Figure 5(a)). The interpretation of the latter is that the stored information effectively acts as a perturbation for the *training-free* probe. On the other hand, the worsening of the prediction loss demonstrates that the stored information is actively utilized during the descent phase. For comparison, when we zero out the same number of layers in the rising phase instead, the loss becomes 0.4255, but the correspondence performance drops, indicating that it has a different mechanism.

With the role of the preparation phase clarified, we next provide a more robust way to measure the length of the preparation phase instead of visually observing the raw correspondence curve. If we use the separability probe to separate x_0 and ϵ signals in MLP increments, we can observe from Figure 5(b) a generally descending trend of the remaining R^2 values after deletion. We further define the last layer before the remaining R^2 drops below 0.1 as the final layer of the preparation phase. This is a more reliable definition compared to the training-free correspondence probe curve, which is sensitive to post-hoc transforms such as whitening (Figure 12 of Appendix).

5.2 Rising Phase

The key characteristic of the rising phase is the formation of the semantic abstract of the clean image, which we denote as x_s. Note that x_s is largely different from x_0, which is a VAE- or RGB-space

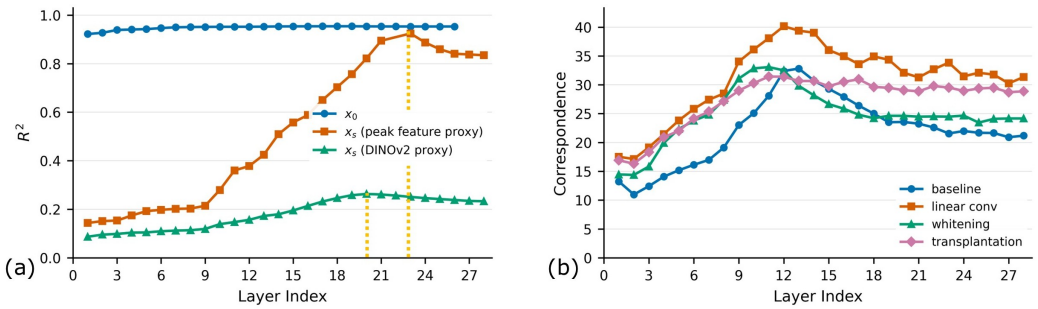

Figure 6: (a) Linear regression probe results of x_0, x_s via peak feature proxy, and x_s via DINOv2 proxy. (b) Correspondence probe results from the FM+ϵ (clean weighting) models, using the baseline training-free probe, linear convolution probe, whitening probe, and transplantation probe (projecting features into the peak feature's space).

signal. Imagine doing nearest-neighbor correspondence on an RGB image directly. The result will naturally be bad compared to doing it on a DINOv2 (Caron et al., 2021) feature. This explains the difference between an abstract, conceptual, semantic x_s and a VAE- or RGB-space x_0.

The behavior in this phase largely aligns with the current knowledge. To provide additional information, we utilize the linear regression probe, where the source feature is the per-block features, and the target feature includes x_0, the peak-layer feature as a proxy of x_s, and DINOv2 (Caron et al., 2021) features as another external proxy of x_s. We can observe from Figure 6(a) that the x_0 curve constantly rises, while the x_s curves (especially the DINOv2 curve) peak at roughly the same position as the correspondence peak. This further strengthens what we have already known.

5.3 Descent Phase

The characteristic behavior of the descent phase is the drop of the correspondence performance. The current interpretation is that the granularity of x_s further becomes finer until the network exit. However, our overall picture gives a different explanation: the drop is because the model writes the encoded prediction target into the residual, making the semantic x_s harder to decode.

To validate our explanation, we add a trainable convolutional layer to the correspondence probe, granting it linear decoding ability. With results in Figure 6(b), whilst the trainable layer brings overall improvement, the improvement is more significant in the descent phase, indicating a "fixing" effect introduced by the linear decoding ability. A similar effect can be observed with whitening and transplantation (into the peak feature's linear space) probes. In other words, the x_s signal still exists to a large extent, only covered under more complex encoding. Nevertheless, this behavior can coexist with the granularity change explanation.

5.4 Effect of Prediction Target

Next, we discuss how the overall picture predicts the potential impact of training setups on the depth profile. The first prediction, related to prediction targets, is that a more hybrid target (such as $u = \epsilon - x_0$) leads to a longer preparation phase, forming a roughly symmetric pattern.

This prediction can be derived from two separate hypotheses. (i) A prediction target that is closer to ϵ naturally requires more ϵ for the encoding, and vice versa, which further affects what should be prepared in the preparation phase. Then, a hybrid prediction target requires more information to be prepared during the preparation phase, as it needs *both* ϵ and x_0 for hints. This gives the preparation phase a harder task, which results in a longer preparation phase. (ii) The difference from the first hypothesis is that what needs to be prepared is determined by inference-time SNR. For example, at a noisy SNR, the descent phase is able to infer ϵ directly from the residual, so it does not need to be prepared in advance. On the other hand, the overall prediction task trivializes for purer targets at certain SNR, *e.g.*, predicting x_0 will be too easy at a clean SNR to form meaningful optimization pressure, so the x_0-model's depth profile is mainly shaped by noisy SNR. Since at noisy SNR, ϵ is not required to be prepared in advance, the x_0-model only learns to prepare x_0. Then, a more

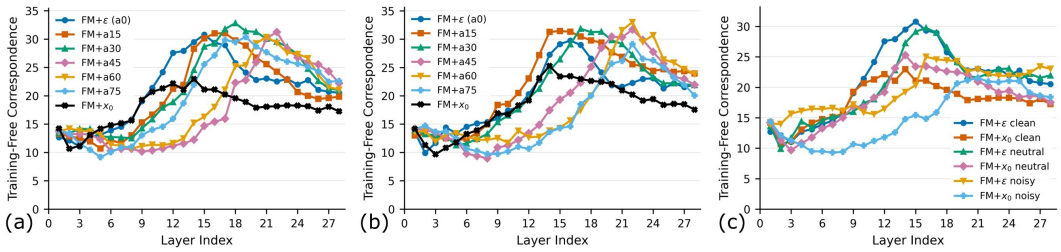

Figure 7: (a) Models trained using the basic min-SNR weighting (emphasizing cleaner samples) with different prediction targets. (b) Models trained using the neutral weighting with different prediction targets. (c) x_0- and ϵ-models trained with three loss weighting methods.

Table 1: Causal intervention by replacing a certain component within the MLP increments with random noise. The more severe loss increase is marked as **bold**. *The same seed is shared across the same column* that controls both image sampling and noise sampling. The absolute loss value is not comparable across models or SNR. All interventions are done with FM+x_0 and FM+ϵ models trained under the neutral weighting.

Setting	logSNR=1.69				logSNR=-1.69			
	seed 1	seed 2	seed 3	avg	seed 1	seed 2	seed 3	avg
x_0 (original)	0.0741	0.0720	0.0734	0.0732	0.3352	0.3347	0.3344	0.3348
ϵ (original)	0.3969	0.3969	0.3941	0.3960	0.0618	0.0642	0.0618	0.0626
x_0 (replacing x_0)	0.1767	**0.1833**	0.1816	0.1805	**0.8191**	**0.8599**	**0.8336**	**0.8375**
x_0 (replacing ϵ)	**0.1838**	0.1813	**0.1841**	**0.1831**	0.7542	0.7758	0.7828	0.7709
ϵ (replacing x_0)	0.6127	0.6162	0.6274	0.6188	0.2258	0.2323	0.2314	0.2298
ϵ (replacing ϵ)	**0.6245**	**0.6277**	**0.6305**	**0.6276**	**0.2350**	**0.2360**	**0.2441**	**0.2384**

hybrid prediction target creates more meaningful optimization pressure on both noisy and clean SNR, leading to longer preparation phases.

First, we observe the predicted roughly symmetric pattern from two groups of models, one trained with the clean weighting (Figure 7(a)) and the other with the neutral weighting (Figure 7(b)). We also complement the two figures with Figure 10 in Appendix, which depicts the pattern using the separability probe. Then, although the two hypotheses lead to the same result, their inner workings are different. The first one suggests that the "importance" of ϵ and x_0 changes solely with the prediction target, while the second one predicts that the importance is more related to the SNR at which we do the inference. Therefore, we make another causal intervention. The target of intervention is still MLP increments in the preparation phase, but we do not zero out the entire increment. Instead, we selectively replace the ϵ part with random noise while minimizing the damage to the x_0 part, or vice versa. The result in Table 1 uses the same random seed across the same column for better alignment. It shows a mixed pattern, suggesting the actual case might be a mixture of both hypotheses.

5.5 Effect of Loss Weighting

The second prediction, related to loss weighting, is two-fold. (i) A pure target might undergo certain degeneration under certain weighting. (ii) A prediction target where x_0 and ϵ are not entirely symmetric will have the length of the preparation phase change.

Degeneration: Taking ϵ as an example of pure targets, it has a certain degeneration scenario. At a very noisy SNR where the input is almost noise, the task of predicting the noise becomes trivial, as the model can almost do nothing and still get a low loss, thus creating weak optimization pressure. When the loss weighting emphasizes an SNR section that corresponds to this degeneration scenario, the model will become degenerated, as a good internal x_s signal is unnecessary. This prediction is validated by the observation in Figure 7(c) that the ϵ-model is normal at the neutral and the clean weighting, but degenerated at the noisy weighting.

Table 2: Generation performance of aligning REPA at different layers during training. Best marked as **bold** and second marked as underlined. The models are trained for 100k steps, and the evaluation is done on 10k samples generated using CFG=1.5.

REPA Alignment Layer	5	7	9	11	13	13→11→10→9
FID↓	15.56	13.4	12.49	12.31	12.33	**11.43**
IS↑	101.96	116.6	124.56	125.63	123.6	**133.2**

Length of Preparation Phase: When the loss weighting emphasizes a certain part of the SNR-axis, the task for the preparation phase may become harder. For example, for a target that has more ϵ, the task is *harder* at a cleaner SNR, since there is *less* amount of ϵ in the model input, which results in a *longer* preparation phase. This prediction is validated by the observation that the $a60$-model has a shorter preparation phase with an SNR-weighting that focuses on the clean part (Figure 7(a)) than with a neutral weighting (Figure 7(b)). Different from the first effect that only applies to highly pure prediction targets, the second effect is more general.

Inherent Asymmetry of x_0 and ϵ: We can also observe from Figure 7(c) that there is a certain inherent asymmetry between x_0 and ϵ. For instance, at a neutral weighting, ϵ is normal while x_0 is degenerated. We have confirmed the cause to be the VAE scaling factor of 0.18215. This number is initially derived by the Stable Diffusion team (Rombach et al., 2022) *from their data* to align the variance of x_0 and ϵ, but it actually leads to misalignment on a different dataset such as ImageNet (Krizhevsky et al., 2012). We will make a more detailed discussion in Section A.

6 Practical Corollaries and Illustrative Applications

REPA proposes to introduce an auxiliary loss that explicitly aligns diffusion features to external self-supervised features, such as DINOv2 (Caron et al., 2021) features. The optimal placement of the REPA loss can be derived from our findings. Specifically, we notice the reported optimal layer in the original paper (Yu et al., 2025) is around the interface between the preparation phase and the rising phase. To test our mechanistic findings in a real application scenario, we measure the length of the preparation phase of the $a60$-model trained with the basic min-SNR weighting. The length is 9 at 200k training steps, so we put the REPA loss at the last layer of the preparation phase. The results in Table 2 show that our prediction is around the optimal placement but not the exact optimum.

Additionally, we have also noticed that the length of the preparation phase gradually shortens throughout the training (Figure 9 in Appendix). So a more aggressive prediction of our findings is that we can dynamically move the REPA placement to earlier layers as the training goes on. For the concrete $a60$-model, we adopt a setting of $\{13@0\rightarrow11@30k\rightarrow10@40k\rightarrow9@50k\}$ where the REPA loss stays at layer 9 after 50k steps. This choice is based on the measured preparation phase length at different training steps rather than being tuned. The rightmost column in Table 2 achieves 11.43 FID score and 133.2 IS score, surpassing the previous optimum and validating the prediction. This can also explain why our previous prediction of layer 9 is not the exact optimum: in earlier training steps, a deeper layer would be better. If we use a static alignment layer, it might be better to take the average across the entire training process. As for why REPA can work, our findings may provide some new angles (Figure 11 in Appendix), but we leave that for future work.

7 Conclusion & Limitation

This work studies the depth profile of diffusion backbones. We provide new empirical results that cover how the depth profile changes with training setups. Furthermore, we offer novel and more complete mechanism findings, validated by the new observations. The new findings may potentially impact applications that rely on this model property, such as the layer selection of REPA.

Note that our work is based on DiT/SiT-style backbone architectures, which creates a gap between our work and larger-scale models such as Flux or Qwen-Image. Additionally, although we have tried to employ more diverse probes to get a more complete picture, these probes are still biased in nature, as all model probing studies are. Overall, our findings are falsifiable as the scope expands.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools to: (i) Propose the necessary components for the alignment training framework, though the idea of alignment itself is proposed by humans. (ii) Code, with necessary human reviewing. (iii) Help interpret probing results and design further probes to dig deeper into the backbones, though quite often human intelligence is still required where LLMs fail. For instance, 2 in 4 of the major breakthroughs during the research process is mainly made by humans while the other 2 by LLMs. (iv) Provide feedback for the writing, though we do not directly use LLMs to refine writing, because of personal taste. Also, we notice that the LLM feedback often focuses too much on details that contribute little to our main claim/contribution, requiring generally less meaningful ablation experiments to be added. We have not used generative AI tools for finding motivations and ideas / help develop theoretical models or conceptual frameworks, and generate synthetic data sets / formulate mathematical claims / provide critical ingredients for proving mathematical claims / assist in the writing of proofs / propose or refine hypotheses /assist with translation / clean and reformat dataset / support qualitative and thematic data analysis are not applicable to this work.

Additionally, we used generative AI tools for: (i) Create matplotlib codes to render the plots (but the illustrative figures are drawn by humans). This is reliable by nature. (ii) Summarize or analyse existing literature (using the GPT-Researcher harness to enhance reliability, along with necessary human reviewing). (iii) Format references. For example, when directly copying the DBLP bibtex citations, the conference titles are often not directly usable, or the page number may be missing for certain conferences or ArXiV papers. We use an agentic LLM rather than a pure-text-based LLM and develop a skill to enhance reliability.

We have reviewed all AI-assisted work or use other means to enhance reliability, as described immediately after each usage. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This is a general diffusion model study and does not seem to relate to ethical issues.

REPRODUCIBILITY STATEMENT

We plan to release our codes after acceptance, but not during reviewing. When releasing, we will include both training codes and probing codes, along with the command we used.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *International Conference on Learning Representations*, pp. 1–13, 2017.

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

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*, pp. 9630–9640, 2021.

Xinlei Chen, Zhuang Liu, Saining Xie, and Kaiming He. Deconstructing denoising diffusion models for self-supervised learning. In *International Conference on Learning Representations*, 2025.

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

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Trans. Assoc. Comput. Linguistics*, 9:160–175, 2021.

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*, 2025.

Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. In *IEEE/CVF International Conference on Computer Vision*, pp. 7407–7417, 2023.

Alec Helbling, Tuna Han Salih Meral, Benjamin Hoover, Pinar Yanardag, and Duen Horng Chau. Conceptattention: Diffusion transformers learn highly interpretable features. *CoRR*, abs/2502.04320, 2025.

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

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In *International Conference on Machine Learning*, volume 235, pp. 20617–20642, 2024.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P. Kingma and Ruiqi Gao. Understanding diffusion objectives as the ELBO with simple data augmentation. In *Annual Conference on Neural Information Processing Systems*, 2023.

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

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

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.

Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell. Diffusion hyperfeatures: Searching through time and space for semantic correspondence. In *Annual Conference on Neural Information Processing Systems*, pp. 1–11, 2023.

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, pp. 23–40, 2024.

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*, pp. 55141–55177, 2024.

Benyuan Meng, Qianqian Xu, Zitai Wang, Xiaochun Cao, Longtao Huang, and Qingming Huang. Making training-free diffusion segmentors scale with the generative power. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35861–35871, 2026.

Tomás Mikolov, Quoc V. Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168, 2013.

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.

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

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.

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*, pp. 10674–10685, 2022.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

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.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. Whitening sentence representations for better semantics and faster retrieval. *CoRR*, abs/2103.15316, 2021.

Viacheslav Surkov, Chris Wendler, Antonio Mari, Mikhail Terekhov, Justin Deschenaux, Robert West, Caglar Gulcehre, and David Bau. One-step is enough: Sparse autoencoders for text-to-image diffusion models. In *Annual Conference on Neural Information Processing Systems*, 2025.

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*, pp. 1–27, 2023a.

Raphael Tang, Linqing Liu, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Pontus Stenetorp, Jimmy Lin, and Ferhan Ture. What the DAAM: interpreting stable diffusion using cross attention. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 5644–5659, 2023b.

Berk Tinaz, Zalan Fabian, and Mahdi Soltanolkotabi. Emergence and evolution of interpretable concepts in diffusion models. In *Annual Conference on Neural Information Processing Systems*, 2025.

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.

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*, pp. 2955–2966, 2023.

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.

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*, pp. 1–15, 2023.

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*, pp. 5706–5716, 2023.

Jingyi Zhou, Jiamu Sheng, Peng Ye, Jiayuan Fan, Tong He, Bin Wang, and Tao Chen. Exploring multi-timestep multi-stage diffusion features for hyperspectral image classification. *IEEE Trans. Geosci. Remote. Sens.*, 62:1–16, 2024.

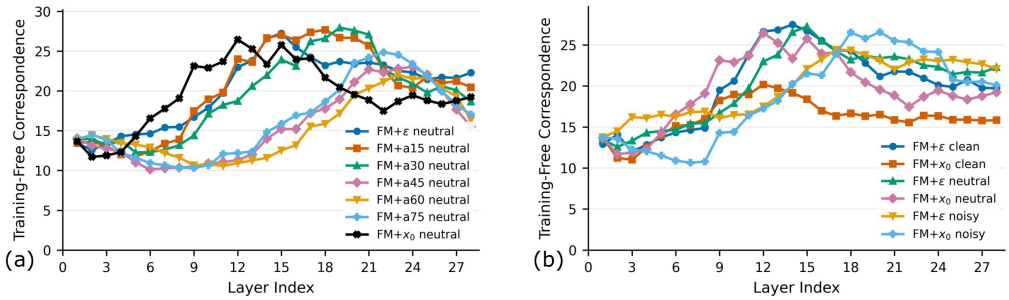

Figure 8: (a) Models of different prediction targets trained with a fixed VAE scaling factor, under the neutral loss weighting. (b) Models trained under different loss weighting with a fixed VAE scaling factor. Models in both figures are trained for 75k steps.

A Fixing VAE Scaling Factor

To expose the misalignment between x_0 and ϵ brought by the VAE scaling factor 0.18215, we use linear regression to measure x_t, which is not even sent into the model yet. The result shows that the x_0 signal is inherently weaker ($R^2 = 0.4451$) compared to the ϵ signal ($R^2 = 0.6052$). This is because the common practice multiplies the VAE-compressed x_0 by 0.18215, which is used by the original Stable Diffusion (Rombach et al., 2022) team to align the variance of x_0 and ϵ and *only applies to their data*. The currently common practice of using the same number is thus in fact improper and will lead to variance misalignment. Nevertheless, the impact on generation performance is often more negligible than the impact on the depth profile, which might explain why the specific number is used so widely, despite the fact that it is derived from a very specific case.

The x_0 signal being inherently weaker can explain two effects: (i) With the neutral weighting, the $a60$-model has a much longer preparation phase than the $a30$-model (Figure 7(b)). This is because $a60$ has more x_0 components, and preparing x_0 is harder, leading to a longer preparation phase. (ii) x_0-models are more prone to degeneration (Figure 7(c)), because they tend to face a task that is too hard, with the x_0 signal being too weak.

To fix this effect and obtain more reliable observations, we have preliminarily trained models with a properly designed scaling factor (0.218265 for ImageNet (Krizhevsky et al., 2012)). The new results include a prediction target comparison and a loss weighting comparison. Since these models are trained for fewer steps due to the time limit, their behaviors are not directly comparable with the model reported in the main paper. Nevertheless, we can clearly observe the alignment effect: (i) In Figure 8(a), $a30$- and $a60$-models achieve stronger symmetry. (ii) In Figure 8(b), x_0- and ϵ-models with the neutral weighting are closer to each other with respect to their peak heights. However, there is still some asymmetry remaining, which might suggest there is a *real* inherent difference between x_0 and ϵ. We leave this for future work.

In conclusion, we want to point out that it is necessary for diffusion backbone behavior studies to use a properly calculated VAE scaling factor, so that the model's behavior is standardized and unbiased.

B Additional Probing Results

We use separability probe to measure how the length of the preparation phase changes with the training steps. The results in Figure 9 suggest that the preparation phase gradually shortens as the training goes on, but the shortening rate slows down. This result also serves as evidence for our dynamic REPA alignment design.

Figure 10 serves as complementary information to Figure 7(a,b) by showing the roughly symmetric patterns via separability probing.

To provide preliminary empirical observations for future work on why REPA can work, we use both separability probe and the training-free correspondence probe to measure the $a60$ model trained with the dynamic REPA. It can be observed from Figure 11 that the separability in early layers improves

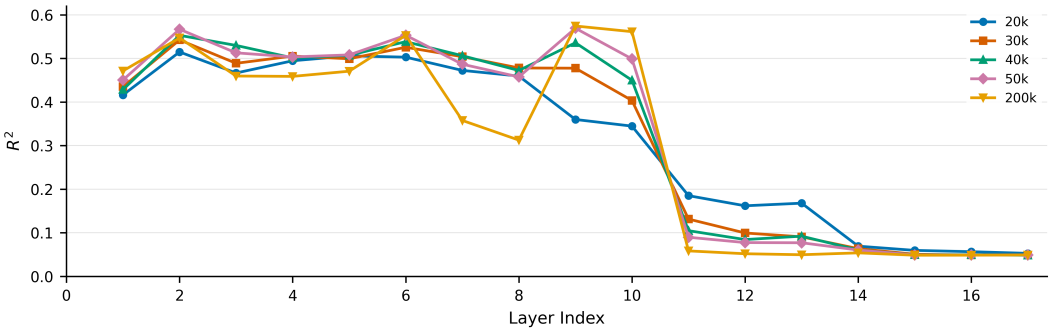

Figure 9: Using separability probe to measure the length of the preparation phase as the training goes on. We can observe that the preparation phase gradually shortens.

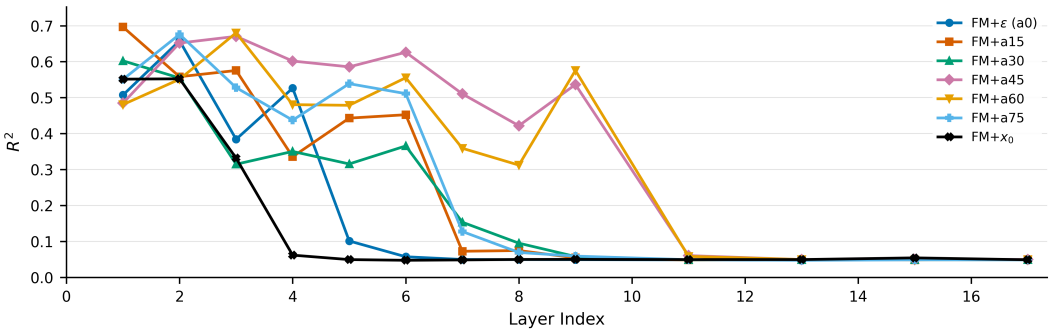

Figure 10: Using separability probe to depict the roughly symmetric pattern caused by prediction targets. The models are trained with the clean weighting.

significantly, and the downstream task performance is also greatly boosted. Hence, separability between x_0 and ϵ in early-layer MLP increments might serve as a way to reveal a new explanation for why REPA can work.

To show that the separability probe dropping below a certain threshold is a robust definition for the length of the preparation phase, we measure the separability of the same model with whitening and with different SNR. The results in Figure 12 clearly demonstrate the robustness.

We have shown that certain loss weighting can cause degeneration. To know whether degeneration will also make generation worse, we evaluate some models with generation metrics. The results in Table 3 suggest that degeneration does not necessarily lead to poor generation, or its impact is relatively small. Hence, if diffusion features are not of interest, degeneration caused by certain training setups is acceptable to some extent.

We have also uploaded additional probing results in .csv format in the supplementary material.

C Implementation Details

C.1 Model & Training

Model Architecture. One notable implementation detail is how the timestep input t is embedded into the network. We first cast the original t into normalized logSNR and embed it instead. This effectively aligns the two sampling paths, which originally use different timestep definitions.

Training. The training uses ImageNet (Krizhevsky et al., 2012) at 256×256 resolution as the training set. We use the same VAE as used in DiT (Peebles & Xie, 2023) and SiT (Ma et al., 2024). We use AdamW (Kingma & Ba, 2014) optimizer, with a constant 1e-4 learning rate, betas as (0.9, 0.999), eps as 1e-8, wd as 0. We add a learning rate scheduler that linearly warms up through the first 5000 steps and then stays constant afterward. The batch size is 256. Additionally, we implement

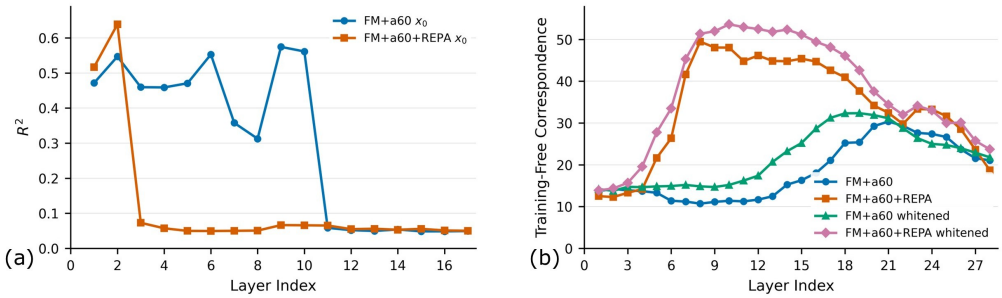

Figure 11: (a) Using separability probe to measure the REPA-augmented model, finding that the separability in early layers significantly improves. (b) The correspondence performance as well as the curve shape are also changed greatly.

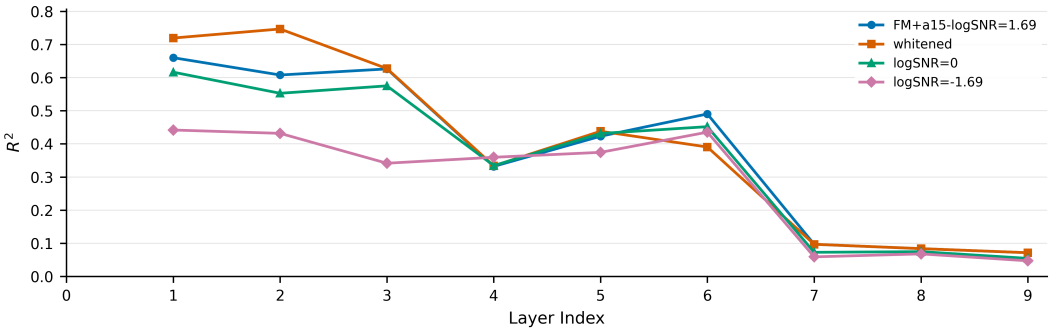

Figure 12: Measuring the separability of the same FM+$a15$ model with whitening and with different SNR. This result shows the robustness of separability probe as the definition of the length of the preparation phase.

gradient clipping with global norm set to 1.0. The logSNR sampling span is [-10.12, 9.21], which is calculated from the original DDPM (Ho et al., 2020) setting.

C.2 Probes

Training-Free & Convolutional Correspondence. Training-free semantic correspondence is evaluated on the SPair-71k (Min et al., 2019) dataset using nearest neighbor matching via cosine similarity. The results are reported under PCK@0.1, which counts the percentage of correctly predicted samples, where a prediction is considered correct if it lies within a radius of the ground truth point based on 0.1 times the image width and height. For the convolutional correspondence probe, we add one linear convolutional layer before nearest neighbor to refine the features, which is trained for 500 steps only.

Whitening. Given a batch of features f, we first accumulate the mean μ and centered covariance C. C is factorized as $C = V\Lambda V^T, \Lambda = \text{diag}(\lambda_1, \cdots, \lambda_d)$. Then, we calculate the whitening matrix as $M = V\text{diag}(\lambda_1 + \kappa\lambda_{\text{max}}, \cdots, \lambda_n + \kappa\lambda_{\text{max}})^{-1/2}V^T$, where κ is a relative truncation threshold. Finally, the per-layer feature is processed as $f' = (f - \mu)M$.

Transplantation. For a source feature f_s and a reference feature f_r, we transplant the source feature into the linear space of the reference feature. First, accumulate the cross-covariance between the two features. Then obtain a projection matrix using a closed-form least-squares problem.

Linear Regression. For this probe and the following probes, the experiments are done on ImageNet (Krizhevsky et al., 2012). The diffusion features within DiT backbones are a token sequence, *i.e.*, a sequence of vectors in the shape of $\mathbb{R}^{c\times hw}$. We first align the size of source and target sequences using bilinear interpolation. Then, on the two aligned features, each token in the sequence

Table 3: We measure the generation performance of well-conditioned and degenerated models. The results appear to suggest that degeneration does not necessarily lead to poor generation. All models use FM paths.

Model	x_0-clean	x_0-neutral	x_0-noisy	ϵ-clean	ϵ-neutral	ϵ-noisy
FID↓	62.66	60.55	72.16	61.62	63.22	82.71
IS↑	50.62	59.61	41.16	55.19	52.81	31.80

serves as a linear regression target. After regression, the reported R^2 is computed separately for each channel over all tokens and then averaged.

Separability Probe. Denote the per-layer feature as f_s, the feature to delete as f_t, and the feature to protect as f_p. First, compute the standardized cross-covariance between f_s and f_t, and take an orthonormal basis of its column space as S_t. Then, f_s is orthogonally projected to remove this subspace:

$$f_s' = f_s(I - S_t S_t^T) \tag{1}$$

Geometry Probe. Given a feature and another reference feature, the geometry probe measures the relative norm and the angle between the given feature and the reference feature. The norm is calculated as follows: first calculate the norm of each token in the sequence, and then take the average over the sequence. The angle is calculated as follows: we first compute the centered covariance GRAM of the entire token sequence, and then use the direction of its top eigenvector as the direction of this whole feature. Afterward, we compute the cosine angle between the two given features.