WHAT AND WHEN TO ALIGN: HIERARCHY-DECOUPLED REPRESENTATION ALIGNMENT FOR PIXEL-SPACE DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Representation alignment accelerates the training of diffusion models by leveraging pretrained visual representations. However, existing methods treat these pretrained visual representations as uniform alignment targets, overlooking differences in semantic granularity and their roles in diffusion training. We investigate these differences in pixel-space diffusion models and find that alignment targets with stronger *component-level semantics* are associated with faster early improvements in generation quality, whereas *category-level semantics* provide complementary object-level supervision throughout training. Based on these findings, we propose HierarchY-Decoupled Representation Alignment (**HYDRA**), which decouples supervision into *Scaffold* and *Anchor alignments* depending on their semantic roles. HYDRA further introduces Position-wise Feature Centering to mitigate position-dependent biases in pretrained features used as alignment targets, and a timestep-conditioned projector to adapt alignment to representations that vary across diffusion timesteps. HYDRA improves generation performance across model scales on ImageNet with classifier-free guidance. JiT-B/16 reaches baseline-level FID with $3\times$ fewer training epochs and further achieves 3.15, while JiT-H/16 improves the final FID from 1.86 to 1.74. Our results highlight that effective representation alignment requires considering not only what representations to align, but also when they should guide generative learning.

1 INTRODUCTION

Diffusion models (Ho et al., 2020; Song et al., 2021) and flow matching models (Lipman et al., 2023; Liu et al., 2023) have demonstrated remarkable image generation performance through a generative process that progressively transforms a noise distribution into the data distribution. Early diffusion models (Ho et al., 2020; Dhariwal & Nichol, 2021) directly modeled high-dimensional pixel distributions, but suffered from rapidly increasing computational costs as image resolution scaled up. To alleviate the computational burden of pixel-space modeling, latent diffusion models (Rombach et al., 2022) compress images into a low-dimensional latent space using a pretrained autoencoder (Kingma & Welling, 2014) and perform the generative process in this latent space. While this design substantially improves training and sampling efficiency, the quality of the generated images inherently depends on the representational capacity and reconstruction fidelity of the pretrained autoencoder. Specifically, visual information lost during the encoding process of the autoencoder may not be fully recoverable by the decoder, potentially limiting the generation of fine-grained visual details. Motivated by these limitations, pixel-space generative models have recently regained attention. A representative example is Just image Transformers (JiT) (Li & He, 2026), which adopts the clean image as the prediction target, thereby reducing the burden of directly preserving and predicting high-dimensional noise information. By focusing the learning objective on the low-dimensional image manifold, JiT demonstrates that high-dimensional pixel-space generation can be effectively modeled with a simple Transformer architecture (Vaswani et al., 2017; Dosovitskiy et al., 2021) employing large image patches. By avoiding the reconstruction bottleneck introduced by latent compression, pixel-space generation enables direct modeling of the image distribution, but requires the generative model to learn the entire visual hierarchy, from global semantic structure to fine-grained local appearance.

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.

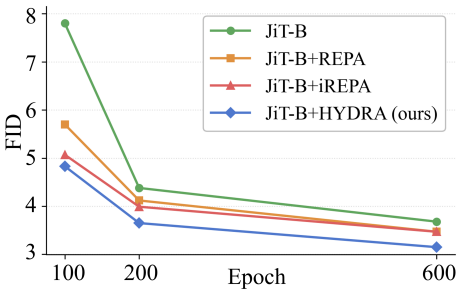


Figure 1: FID comparison of JiT-B/16 variants across training epochs on ImageNet at 256×256 resolution.

Therefore, pixel-space generative models must efficiently learn multi-level visual representations.

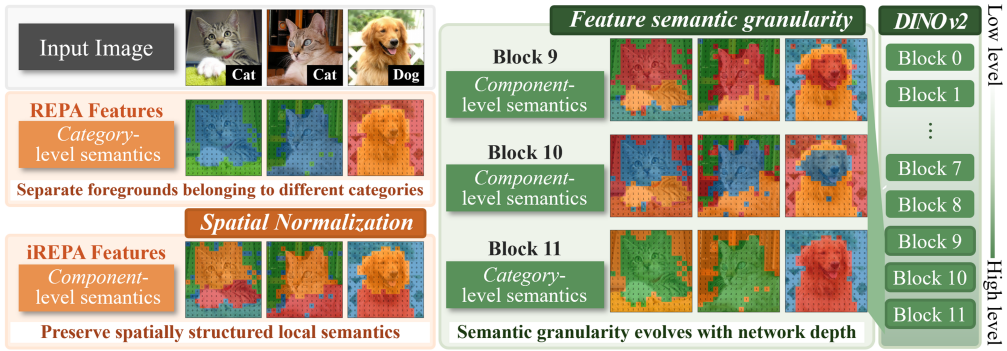
Meanwhile, Representation Alignment (REPA) (Yu et al., 2025) was introduced as an effective approach for improving both the training efficiency and generation performance of latent diffusion models. REPA aligns hidden states of a diffusion model with clean-image representations from pretrained visual encoders (Zhai et al., 2023; Radford et al., 2021), such as DINOv2 (Oquab et al., 2024), providing supervision beyond the denoising objective and accelerating convergence. However, it remains unclear which properties of pretrained representations contribute to rapid convergence and which are critical for final generation performance. Understanding this relationship is crucial for elucidating how representation supervision benefits diffusion models and for designing more effective alignment strategies. Motivated by this importance, iREPA (Singh et al., 2026) shows that spatial contrast among patch tokens is closely associated with fast Fréchet Inception Distance (FID) convergence. Building on this finding, iREPA applies spatial normalization to pretrained visual representations to enhance spatial contrast across patches. As shown in Fig. 1, iREPA also effectively accelerates early FID convergence in pixel-space diffusion models such as JiT. However, as training progresses, the performance gap between iREPA and REPA gradually narrows, and the two methods eventually converge to similar performance. This trend suggests that spatially structured representations are particularly effective for accelerating early convergence, although their relative advantage diminishes as training progresses. This raises a broader question: ***which properties of visual representations underlie their distinct roles over the course of training?***

To address this question, we examine the semantic granularity of alignment representations by comparing how REPA and iREPA organize patch representations across object categories and semantic components. We find that the target features used by REPA exhibit category-level semantics, where patch representations are primarily organized according to object category, whereas iREPA exhibits component-level semantics, where semantically corresponding components are represented similarly even across different object categories. Taken together with the training dynamics in Fig. 1, these findings suggest that component-level and category-level semantics play distinct roles in representation alignment: the former facilitates rapid early convergence, whereas the latter provides continuous object-level semantic supervision throughout training.

Importantly, our analysis reveals that both semantic properties are naturally present within the DINOv2 representation hierarchy. Representations from late-intermediate blocks exhibit stronger component-level semantics, whereas final-block representations exhibit stronger category-level semantics, as detailed in Sec. 3.1. This suggests that the representation hierarchy itself contains alignment targets with distinct semantic properties, providing an opportunity to exploit different levels of the hierarchy according to their roles during training.

Building on these observations, we propose HierarchY-Decoupled Representation Alignment (HYDRA), which decouples the pretrained representation hierarchy into two distinct alignment roles, Scaffold and Anchor, according to semantic granularity. Scaffold alignment leverages component-level semantics to accelerate early representation learning by promoting fine-grained structural organization, and is therefore applied only during early training, whereas Anchor alignment uses category-level semantics to provide object-level semantic supervision throughout training. In addition, HYDRA incorporates Position-wise Feature Centering to mitigate position-dependent bias in pretrained representations and a timestep-conditioned projector to account for timestep-dependent variation in diffusion representations. Our main contributions are summarized as follows:

- We characterize the semantic granularity of REPA and iREPA, and uncover a DINOv2 hierarchy that transitions from component-level features to category-level final-block features.
- We show that alignment with late-intermediate DINOv2 features, which exhibit stronger component-level semantics, yields faster early FID convergence than iREPA.
- We propose HYDRA, which decouples the pretrained representation hierarchy into distinct alignment roles according to semantic granularity, together with a timestep-conditioned projector.



(a) Spatial normalization in alignment features (b) Block-wise semantic granularity in DINOv2

Figure 2: Comparison of semantic granularity in the alignment features of REPA, iREPA, and different DINOv2 blocks using K-means clustering ($k = 4$). (a) REPA exhibits category-level clustering, whereas iREPA exhibits component-level clustering. (b) Intermediate DINOv2 blocks exhibit component-level characteristics, while the final block shows category-level characteristics.

2 Preliminaries

2.1 Just Image Transformers

Our method builds on Just image Transformers (JiT) (Li & He, 2026), a pixel-space generative model trained with the Flow Matching (FM) objective (Lipman et al., 2023; Liu et al., 2023). Let p_0 denote the Gaussian noise distribution and p_1 the data distribution, with $x_0 \sim p_0$ and $x_1 \sim p_1$. FM defines a continuous interpolation between x_0 and x_1 as

$$x_t = t \cdot x_1 + (1 - t) \cdot x_0, \qquad v_t = x_1 - x_0, \tag{1}$$

where $t \in [0, 1]$ denotes the timestep, x_t the intermediate image state, and v_t the corresponding target velocity. The model learns a parameterized velocity field $v_\theta(x_t, t)$ by minimizing

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{x_0, x_1, t} \| v_\theta\left(x_t, t\right) - v_t \|_2^2, \tag{2}$$

where θ denotes the model parameters. During sampling, images are generated by integrating the ODE $dx_t/dt = v_\theta(x_t, t)$ from $t = 0$ to 1 using an ODE solver (Hairer et al., 1993).

Unlike standard FM models that directly predict the velocity, JiT adopts x-prediction (Salimans & Ho, 2022), in which the network predicts the clean image $x_\theta(x_t, t)$ from the intermediate state x_t at timestep t. The corresponding velocity $v_\theta\left(x_t, t\right)$ is obtained as

$$v_\theta(x_t, t) = \big(x_\theta(x_t, t) - x_t\big)/(1 - t). \tag{3}$$

Thus, JiT directly predicts the clean image while retaining the standard FM sampling formulation.

2.2 Representation Alignment

Representation Alignment (REPA) (Yu et al., 2025) facilitates representation learning in diffusion models by aligning their hidden features with representations from pretrained vision encoders. REPA uses rich visual representations from self-supervised vision encoders trained on large-scale image datasets, such as DINOv2 (Oquab et al., 2024), as additional supervision. Specifically, REPA projects intermediate hidden features of the diffusion model into the feature space of the vision encoder and maximizes their cosine similarity with target features at corresponding spatial locations. The REPA loss is defined as follows:

$$\mathcal{L}_{\text{REPA}} = -\mathbb{E}_{x_0, x_1, t} \left[\frac{1}{N} \sum_{i=1}^{N} \text{sim}\left(P_\phi\left(h_\theta^{(i)}\left(x_t, t\right)\right), y^{(i)}\left(x_1\right)\right)\right], \tag{4}$$

where N denotes the number of spatial tokens and i denotes the spatial token index. $h_\theta^{(i)}(x_t, t)$ denotes the i-th intermediate hidden feature of JiT at timestep t, and P_ϕ denotes a trainable projector with parameters ϕ that maps the hidden feature into the feature space of the pretrained vision encoder. $y^{(i)}(x_1)$ denotes the target feature at the i-th spatial location extracted from the clean image x_1 by the pretrained vision encoder, and $\text{sim}(\cdot, \cdot)$ denotes cosine similarity.

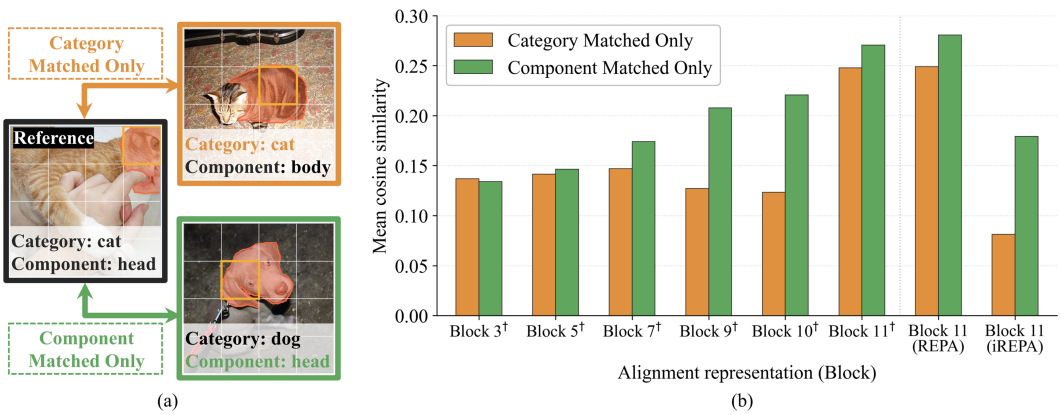


Figure 3: (a) Illustration of the two patch-pair relations defined using PartImageNet annotations: Category Matched Only and Component Matched Only. (b) **Semantic granularity across the DINOv2 hierarchy.** Mean similarity between part representations for the two relation types across DINOv2 blocks, REPA, and iREPA. † denotes features with Position-wise Feature Centering.

3 Method

3.1 Motivation

3.1.1 Analysis of Semantic Granularity of Various Feature Representations

Semantic granularity of REPA and iREPA features. iREPA applies spatial normalization to the DINOv2-derived alignment features, thereby amplifying spatial contrast across patches. While iREPA has primarily focused on how spatial normalization enhances spatial structure, its effect on the semantic organization of patch representations remains underexplored. In this work, we therefore analyze the alignment features of REPA and iREPA from the perspective of semantic granularity, which characterizes the semantic units that govern similarity across patch representations.

Figure 2(a) presents clustering results of DINOv2 patch features extracted from two images of the same class and one image from a different class to compare the target features used for alignment in REPA and iREPA. Specifically, clusters represented by the same color across the three images can be interpreted as regions that DINOv2 considers semantically similar. As shown in Fig. 2(a), foreground patches in REPA are primarily grouped depending on object category, whereas iREPA represents semantically corresponding components similarly even across images from different categories. This suggests that spatial normalization shifts the semantic granularity of the alignment representation from category-level semantics toward component-level semantics. Together with the differing FID trends over training observed in Fig. 1, this difference suggests that the effectiveness of representation alignment may depend on the semantic granularity of the alignment target.

Semantic granularity across the DINOv2 hierarchy. To systematically investigate alignment targets with varying semantic granularities, we extend our analysis to all Transformer blocks of DINOv2. Figure 2(b) shows clustering results of patch features extracted from different levels of the DINOv2 hierarchy. In intermediate blocks, semantically corresponding components tend to be grouped similarly even across different object categories, whereas the final block exhibits more pronounced grouping according to object category. These qualitative results indicate that the semantic granularity of the representations changes along the DINOv2 hierarchy. To quantify this observation, we use PartImageNet (He et al., 2022) to categorize patch pairs into two relation types, as illustrated in Fig. 3(a): Category Matched Only (e.g., cat head–cat body) and Component Matched Only (e.g., cat head–dog head). Based on these relations, Fig. 3(b) compares their average cosine similarities across DINOv2 blocks. From Blocks 3 to 10, the similarity of Component Matched Only pairs gradually increases with network depth. This indicates increasingly consistent representations of the same semantic components across different object categories, reflecting stronger component-level semantics in the late-intermediate hierarchy. In contrast, at Block 11, the similarity of Category Matched Only pairs exhibits a marked increase. This indicates stronger category-level semantics

Table 1: FID (↓) comparison across different alignment targets using JiT-B/16. † denotes features after Position-wise Feature Centering.

Epoch	Alignment Target							
	Block 3†	Block 5†	Block 7†	Block 9†	Block 10†	Block 11†	Block 11 (REPA)	Block 11 (iREPA)
50	16.94	13.77	11.92	11.61	**11.52**	14.36	15.18	12.38
100	6.30	5.29	4.74	**4.70**	**4.70**	5.56	5.69	5.07

across different components of the same object category. Overall, these results reveal a hierarchical transition in semantic granularity, from increasingly pronounced component-level semantics in intermediate blocks to strengthened category-level semantics in the final block.

3.1.2 Analysis of Complementarity Across the DINOv2 Hierarchy

We next examine how the semantic granularity of DINOv2 features across blocks affects the effectiveness of alignment supervision. Table 1 compares the early-stage FID of JiT using features from different DINOv2 blocks as alignment targets. From Blocks 3 to 10, FID consistently improves at both 50 and 100 epochs as component-level semantics become increasingly pronounced, whereas this trend breaks at Block 11, where category-level semantics become substantially stronger. Together with Fig. 3, these results suggest that the effectiveness of early-stage alignment is driven more by the semantic granularity of late-intermediate representations than by feature depth alone. Notably, Blocks 9 and 10 achieve faster early FID convergence than iREPA despite both exhibiting strong component-level semantics. However, they differ in how well semantic relationships across different components are preserved. As shown in Fig. 3(b), iREPA exhibits markedly lower Category Matched Only similarity than Blocks 9 and 10, suggesting that spatial normalization may enhance component-specific contrast while reducing semantic commonality across different components within the same category. In contrast, late-intermediate features retain such cross-component commonality while maintaining component-level discrimination, which may explain their advantage in the early stage.

However, the rapid early convergence enabled by late-intermediate features does not imply that final-block features are unnecessary. As shown in Fig. 1, iREPA converges faster than REPA during the early stage of training, while the performance gap between the two methods gradually narrows as training progresses. Taken together with our preceding analysis of semantic granularity, this observation suggests that component-level representations from late-intermediate blocks effectively accelerate early representation learning, whereas category-level representations from the final block play a complementary role by providing category-level supervision throughout training. Therefore, these two representations should not be viewed as interchangeable alignment targets, but rather as complementary sources of supervision that serve distinct roles according to their semantic properties. Based on this observation, we propose HYDRA, which jointly exploits late-intermediate and final-block representations according to their respective characteristics.

3.2 Hierarchy-Decoupled Representation Alignment

Our preceding analysis shows that late-intermediate and final-block representations in DINOv2 exhibit different semantic granularities and play distinct roles in representation alignment. Based on these findings, we propose **HierarchY-Decoupled Representation Alignment (HYDRA)**, which decouples and exploits hierarchical representations according to their semantic properties and roles. Figure 4 illustrates the overall architecture of HYDRA. HYDRA first applies **Position-wise Feature Centering** to alleviate position-dependent bias in the patch representations of the pretrained vision encoder. It then uses the component-level late-intermediate and category-level final-block representations from the centered feature hierarchy as targets for **Scaffold alignment** and **Anchor alignment**, respectively, thereby providing supervision at different levels of semantic granularity. In addition, because hidden representations in pixel-space diffusion vary substantially across timesteps, we employ a **timestep-conditioned projector** for each alignment branch. Details of each component are provided in the following subsections.

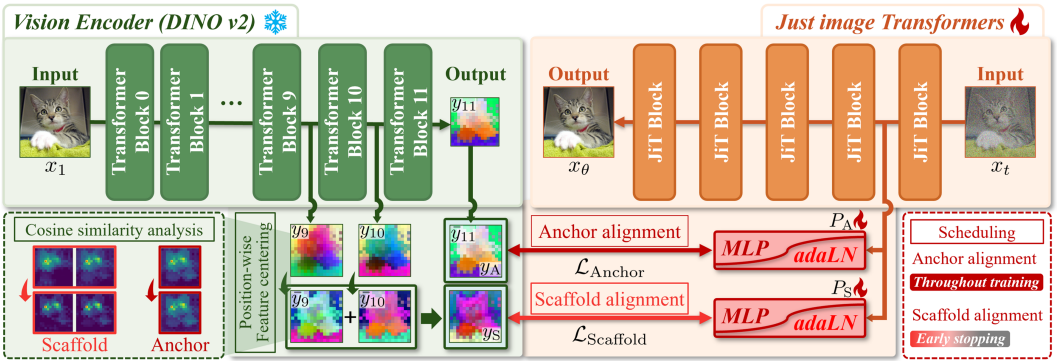


Figure 4: Overview of HYDRA.

3.2.1 Position-wise Feature Centering

Patch token representations from DINOv2 may encode not only the content of an image but also position-dependent components. In particular, the cosine similarity analysis in Fig. 4 shows that DINOv2 features exhibit recurring feature components at the same patch positions across images. Such position-dependent bias can cause tokens at the same spatial location to exhibit high cosine similarity in the feature space regardless of their visual content, interfering with semantic comparisons between patch features. To alleviate this issue, we apply Position-wise Feature Centering. We construct a class-balanced sample set $\mathcal{B}$ by randomly sampling 200 images per class from ImageNet. Let $y_l^{(i)}(x)$ denote the patch feature at spatial position i of image x, extracted from the l-th block of DINOv2. We obtain the centered feature as $\bar{y}_l^{(i)}(x) = y_l^{(i)}(x) - \frac{1}{|\mathcal{B}|}\sum_{x' \in \mathcal{B}} y_l^{(i)}(x')$, thereby suppressing position-dependent components shared across images. We apply this centering to both the late-intermediate and final-block features of DINOv2, which are used as the alignment targets for Scaffold and Anchor alignment, respectively.

3.2.2 Scaffold Alignment and Anchor Alignment

HYDRA formulates Scaffold alignment and Anchor alignment as two complementary loss terms, termed the Scaffold loss and the Anchor loss, to exploit the distinct roles of representations within the centered DINOv2 hierarchy. The Scaffold target is constructed by summing the centered features, $\bar{y}_9^{(i)}(x_1)$ and $\bar{y}_{10}^{(i)}(x_1)$, from DINOv2 Blocks 9 and 10, which provide component-level semantics, while the Anchor target uses the centered feature from Block 11, which provides category-level semantics. For a clean image x_1, the two alignment targets are defined as follows.

$$y_{\mathrm{S}}^{(i)}(x_1) = \bar{y}_9^{(i)}(x_1) + \bar{y}_{10}^{(i)}(x_1), \qquad y_{\mathrm{A}}^{(i)}(x_1) = \bar{y}_{11}^{(i)}(x_1). \tag{5}$$

The Scaffold loss provides representation supervision that encourages the pixel-space diffusion model to rapidly learn component-level visual structure during the early stage of training. The Scaffold loss with projector P_{S} is defined as follows:

$$\mathcal{L}_{\mathrm{Scaffold}} = -\mathbb{E}_{x_0,x_1,t}\left[\frac{1}{N}\sum_{i=1}^{N}\mathrm{sim}\left(P_{\mathrm{S}}\left(h_\theta^{(i)}(x_t,t),t\right), y_{\mathrm{S}}^{(i)}(x_1)\right)\right]. \tag{6}$$

We follow the notation of Sec. 2.2 for N, i, h_θ, and $\mathrm{sim}(\cdot,\cdot)$. The Scaffold loss is used only during early training to accelerate early representation learning. Rather than being applied throughout the entire training process, Scaffold alignment is removed after the early training stage via early stopping. This allows the model to benefit from the rapid early convergence provided by late-intermediate representations, while preventing generative learning from being overly constrained by continued alignment to component-level representations in later stages of training. In contrast, the Anchor loss provides continuous category-level semantic supervision throughout training. The Anchor loss with projector P_{A} is defined as follows:

$$\mathcal{L}_{\mathrm{Anchor}} = -\mathbb{E}_{x_0,x_1,t}\left[\frac{1}{N}\sum_{i=1}^{N}\mathrm{sim}\left(P_{\mathrm{A}}\left(h_\theta^{(i)}(x_t,t),t\right), y_{\mathrm{A}}^{(i)}(x_1)\right)\right]. \tag{7}$$

Unlike the Scaffold loss, the Anchor loss is applied throughout training, as pixel-space diffusion requires prolonged learning to establish high-level semantic representations (Jiang et al., 2026a). In this way, rather than utilizing representations from different hierarchy levels in the same manner, HYDRA decouples the role and application timing of alignment supervision depending on their respective semantic properties. This design directly reflects the complementary roles of late-intermediate and final-block features identified in our preceding analysis. The overall training objective of HYDRA is defined as follows:

$$\mathcal{L}_{\mathrm{HYDRA}} = \mathcal{L}_{\mathrm{FM}} + \lambda_{\mathrm{S}}\mathcal{L}_{\mathrm{Scaffold}} + \lambda_{\mathrm{A}}\mathcal{L}_{\mathrm{Anchor}}, \tag{8}$$

where λ_{S} and λ_{A} are the Scaffold and Anchor loss weights, respectively, detailed in App. C.

3.2.3 Timestep-conditioned Projector

In pixel-space diffusion, the noise level and visual information vary substantially across timesteps, leading to timestep-dependent diffusion representations. Existing alignment methods (Yu et al., 2025; Singh et al., 2026) with lightweight projection heads use a timestep-independent projector that applies the same transformation across all timesteps, potentially limiting timestep-specific alignment. To address this, we incorporate adaptive layer normalization (adaLN) (Peebles & Xie, 2023) into the lightweight MLP projector used in REPA, with the modulation scale and shift dynamically determined by the diffusion timestep. We use two separate timestep-conditioned projectors for Scaffold and Anchor alignment to account for both their distinct hierarchical targets and timestep-dependent diffusion representations.

4 Experiments

4.1 Experimental Setup

Implementation details. Unless otherwise specified, all experiments follow the training and evaluation protocols of JiT (Li & He, 2026). We conduct class-conditional image generation experiments on ImageNet-1K (Russakovsky et al., 2015) at 256×256 resolution. We evaluate HYDRA on JiT-B/16, JiT-L/16, and JiT-H/16, using JiT-B/16 as the primary backbone for analysis and ablation studies due to its computational efficiency. We use a frozen DINOv2-B (Oquab et al., 2024) for representation alignment, following REPA (Yu et al., 2025). We set the loss weights for Scaffold and Anchor alignment to $\lambda_{\mathrm{S}} = 0.07$ and $\lambda_{\mathrm{A}} = 0.03$, respectively, and terminate Scaffold alignment after 150 epochs. Additional details are provided in App. B.

Evaluation. We evaluate generation quality using Fréchet Inception Distance (FID) (Heusel et al., 2017) and Inception Score (IS) (Salimans et al., 2016), computed over 50,000 generated samples. Following JiT, we use a 50-step Heun (Heun, 1900) ODE solver for sampling. We sweep the classifier-free guidance (CFG) (Ho & Salimans, 2021) scale over $[1.0, 4.0]$ and apply a CFG interval (Kynkäänniemi et al., 2024) of $[0.1, 1.0]$.

Baselines. We compare HYDRA with a broad range of representative image generation models. For latent-space diffusion models, we include DiT (Peebles & Xie, 2023), SiT (Ma et al., 2024), REPA (Yu et al., 2025), LightningDiT (Yao et al., 2025), DDT (Wang et al., 2026b), and RAE (Zheng et al., 2026a). For pixel-space diffusion models, we include ADM (Dhariwal & Nichol, 2021), RIN (Jabri et al., 2023), SiD (Hoogeboom et al., 2023), VDM++ (Kingma & Gao, 2023), SiD2 (Hoogeboom et al., 2025), PixelFlow (Chen et al., 2025), PixNerd (Wang et al., 2026a), and JiT (Li & He, 2026). Since HYDRA is built upon JiT, we use JiT-B/16, JiT-L/16, and JiT-H/16 as the primary baselines across model scales, with PixelREPA (Shin et al., 2026) included for comparison with prior pixel-space representation alignment.

4.2 Quantitative Comparison

Table 2 compares HYDRA with representative generative models on ImageNet at 256×256 resolution. While several existing methods achieve strong FID scores using pretrained latent representations or specialized pixel-space architectures, our goal is to improve the plain JiT backbone through hierarchy-decoupled representation alignment. HYDRA consistently improves JiT across model scales, reducing FID from 3.66 to 3.15 on JiT-B/16, 2.36 to 2.06 on JiT-L/16, and 1.86 to 1.74 on JiT-H/16, demonstrating effective scalability across model sizes. HYDRA also consistently

Table 2: Comparison with latent-space and pixel-space generative models on ImageNet 256×256 using classifier-free guidance.

Model	Params	Epochs	FID↓	IS↑
Latent-space Diffusion				
DiT-XL/2	675+49M	1400	2.27	278.2
SiT-XL/2	675+49M	1400	2.06	277.5
REPA, SiT-XL/2	675+49M	800	1.42	305.7
LightningDiT-XL/2	675+49M	800	1.35	295.3
DDT-XL/2	675+49M	400	1.26	310.6
RAE, DiT^{DH}-XL/2	839+415M	800	1.13	262.6
Pixel-space Diffusion				
ADM-G	559M	400	4.59	186.7
RIN	320M	800	3.42	182.0
SiD	2B	800	2.44	256.3
VDM++, UViT/2	2B	1120	2.12	267.7
SiD2, UViT/2	–	1600	1.73	–
SiD2, UViT/1	–	1280	1.38	–
PixelFlow, XL/4	677M	320	1.98	282.1
PixNerd, XL/16	700M	160	2.15	297.0
JiT-G/16	2B	600	1.82	292.6
PixelREPA-B/16	131M	600	3.17	284.6
PixelREPA-L/16	459M	600	2.11	309.5
PixelREPA-H/16	953M	600	1.81	317.2
JiT-B/16	131M	600	3.66	275.1
+HYDRA (ours)		200	3.65	270.2
+HYDRA (ours)		600	3.15	284.7
JiT-L/16	459M	600	2.36	298.5
+HYDRA (ours)		200	2.39	302.0
+HYDRA (ours)		600	2.06	293.8
JiT-H/16	953M	600	1.86	303.4
+HYDRA (ours)		200	1.93	293.3
+HYDRA (ours)		600	1.74	305.8

Table 3: Ablation study of the alignment objectives in HYDRA on JiT-B/16 without Scaffold alignment early stopping. FID is evaluated at 200 epochs.

Alignment		FID ↓
$\mathcal{L}_{\text{Scaffold}}$	$\mathcal{L}_{\text{Anchor}}$	
×	×	4.37
✓	×	3.77
×	✓	3.91
✓	✓	**3.71**

Table 4: Ablation study of early stopping for the alignment losses in HYDRA on JiT-B/16. FID is evaluated at 600 epochs.

Early stopping		FID ↓
$\mathcal{L}_{\text{Scaffold}}$	$\mathcal{L}_{\text{Anchor}}$	
×	×	3.22
✓	×	**3.15**
×	✓	3.33
✓	✓	3.29

Table 5: Ablation study of applying timestep conditioning to the HYDRA projectors on JiT-B/16. FID is evaluated at 200 epochs.

t-cond.	Scaffold early stopping	
	×	✓
×	3.81	3.76
✓	3.71	**3.65**

outperforms PixelREPA across all evaluated model scales. In addition to improving final generation quality, HYDRA reaches baseline-level performance with substantially fewer training epochs, achieving an FID of 3.65 after 200 epochs compared with 3.66 for the baseline after 600 epochs. Similarly, after 200 epochs, HYDRA reduces the FID of JiT-L/16 and JiT-H/16 to 2.39 and 1.93, respectively, approaching their corresponding 600-epoch baseline FIDs of 2.36 and 1.86. Together, these results show that HYDRA improves generation quality and accelerates FID convergence across model scales.

4.3 Ablation Study and Analysis

Effect of Scaffold and Anchor alignment. We examine the contributions of Scaffold and Anchor alignment without Scaffold early stopping in Tab. 3. Both objectives individually improve over the baseline, while their combination achieves the best FID of 3.71. Although Scaffold provides a larger improvement when used alone, incorporating Anchor further improves generation quality, indicating that category-level supervision provides complementary information beyond component-level alignment. This result is consistent with the semantic analysis in Sec. 3.1, which reveals distinct semantic granularities in late-intermediate and final-block representations. To examine their interaction during optimization, App. C.5 analyzes the cosine similarity between the gradients of the JiT block parameters induced by the two alignments. Their gradients remain positively aligned across the evaluated epochs and timesteps, while generally becoming less similar as training progresses. Thus, the objectives induce directionally compatible but increasingly differentiated updates. Together, these results support combining the two objectives, while their training schedules are examined below.

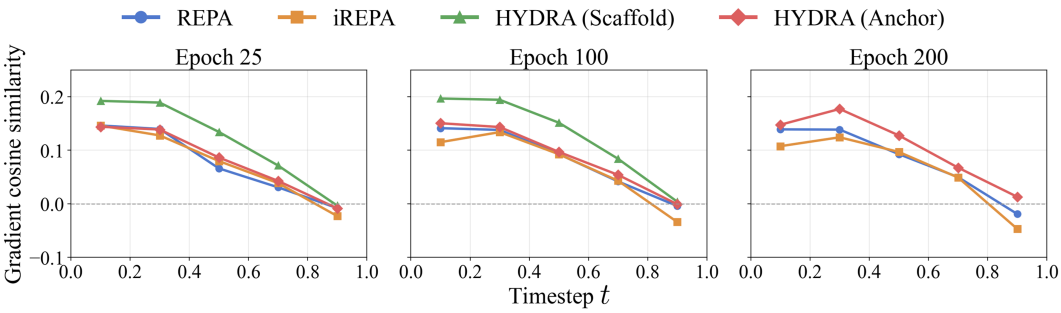


Figure 5: Cosine similarity between gradients of the Transformer block parameters induced by the generative objective and different representation alignment objectives on JiT-B/16.

Gradient compatibility of alignment targets. To further examine how Scaffold and Anchor alignment interact with the generative objective, we analyze the cosine similarity between the gradients induced by each alignment loss and the Flow Matching (FM) loss, with detailed measurement settings provided in App. C.4. As shown in Fig. 5, Scaffold alignment exhibits consistently high positive gradient similarity with the FM objective across most timesteps and training stages, indicating highly compatible optimization directions. In contrast, Anchor alignment maintains positive similarity over a broader range of timesteps and exhibits substantially weaker gradient conflict at large timesteps than REPA and iREPA, particularly in later training stages. These results further support their distinct roles: Scaffold provides compatible supervision that facilitates early optimization, while Anchor preserves semantic guidance with limited interference with the generative objective throughout training.

Effect of early stopping. We examine the early stopping schedules of Scaffold and Anchor alignment at 150 epochs in Tab. 4. Stopping Scaffold alignment early while maintaining Anchor alignment throughout training achieves the best FID of 3.15. Applying early stopping only to Anchor alignment gives an FID of 3.33, and stopping both alignment objectives gives 3.29. These results support the complementary scheduling used in HYDRA: Scaffold alignment serves as temporary supervision, while Anchor alignment is maintained throughout training to provide continuous category-level semantic guidance. Unlike prior findings in latent-space diffusion, where prolonged representation alignment can hinder later-stage training and is therefore terminated (Wang et al., 2025), our results suggest that sustained category-level alignment remains beneficial in pixel-space diffusion by providing semantic guidance that is otherwise difficult to learn directly in pixel space.

Effect of timestep conditioning. We investigate the effect of applying timestep conditioning to the alignment projectors in Tab. 5. Timestep conditioning consistently improves performance regardless of Scaffold early stopping, reducing FID by 0.10 without early stopping and by 0.11 with early stopping. Since the noise level and available visual information vary across diffusion timesteps, diffusion representations exhibit timestep-dependent characteristics. Conditioning the projector on t therefore enables timestep-adaptive alignment.

Appendix C provides additional ablations on feature centering, target composition, and loss weights, along with computational costs and DINOv3 results.

5 Conclusion

In this work, we investigated how the semantic granularity of pretrained visual representations affects representation alignment in pixel-space diffusion models. Our analysis shows that late-intermediate DINOv2 features provide component-level semantics that accelerate early-stage convergence, while final-block features offer category-level semantics that provide complementary supervision throughout training. Based on these observations, we proposed HYDRA, which decouples these complementary roles through Scaffold and Anchor alignment, together with a timestep-conditioned projector. HYDRA consistently improves generation quality and accelerates FID convergence across model scales on ImageNet. These results highlight the importance of considering both what representations to align and when to apply them in pixel-space generative modeling.

Reproducibility Statement

We provide detailed model architectures, training and sampling configurations, alignment settings, and hyperparameters in Appendix B. Additional ablation studies and analyses supporting the design choices of HYDRA are provided in Appendix C. These details are intended to facilitate reproduction of the experimental results reported in this work.

Ethics Statement

This work focuses on improving the training efficiency and generation quality of pixel-space diffusion models using the publicly available ImageNet-1K (Russakovsky et al., 2015) benchmark. We do not collect new human-subject data or use private personal information. As with other generative image models, the resulting models may inherit biases present in the training data and could potentially be used to generate misleading or inappropriate visual content. We encourage responsible use of generative models and careful consideration of dataset biases and potential downstream misuse.

AI Use Statement

In this work, we used generative AI tools for language polishing, literature retrieval and discovery, and code debugging. All AI-assisted text, retrieved information, and code were reviewed and verified by the authors. We take full responsibility for the final content of this work.

References

Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. Pixelflow: Pixel-space generative models with flow. *arXiv preprint arXiv:2504.07963*, 2025.

Zhennan Chen, Junwei Zhu, Xu Chen, Jiangning Zhang, Xiaobin Hu, Hanzhen Zhao, Chengjie Wang, Jian Yang, and Ying Tai. Dip: Taming diffusion models in pixel space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 36136–36146, June 2026.

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

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

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR, 21–27 Jul 2024.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Yasaman Haghighi, Bastien Van Delft, Mariam Hassan, and Alexandre Alahi. Layersync: Self-aligning intermediate layers. In *The Fourteenth International Conference on Learning Representations*, 2026.

Ernst Hairer, Syvert P. Nørsett, and Gerhard Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*, volume 8 of *Springer Series in Computational Mathematics*. Springer, 2nd edition, 1993. ISBN 978-3-540-56670-0.

Ju He, Shuo Yang, Shaokang Yang, Adam Kortylewski, Xiaoding Yuan, Jie-Neng Chen, Shuai Liu, Cheng Yang, Qihang Yu, and Alan Yuille. Partimagenet: A large, high-quality dataset of parts. In *European Conference on Computer Vision*, pp. 128–145, 2022.

Karl Heun. Neue methoden zur approximativen integration der differentialgleichungen einer unabhängigen veränderlichen. *Z. Math. Phys*, 45:23–38, 1900.

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*, volume 30, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 13213–13232. PMLR, 23–29 Jul 2023.

Emiel Hoogeboom, Thomas Mensink, Jonathan Heek, Kay Lamerigts, Ruiqi Gao, and Tim Salimans. Simpler diffusion: 1.5 fid on imagenet512 with pixel-space diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18062–18071, June 2025.

Allan Jabri, David J. Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 14569–14589. PMLR, 23–29 Jul 2023.

Dengyang Jiang, Ruoyi Du, Zhennan Chen, Dongyang Liu, Zanyi Wang, Mingzhe Zheng, Xiangpeng Yang, Huanqia Cai, Aiming Hao, Yuming Jiang, et al. An empirical study of training pixel-space text-to-image diffusion models. *arXiv preprint arXiv:2608.16887*, 2026a.

Dengyang Jiang, Mengmeng Wang, Liuzhuozheng Li, Lei Zhang, Haoyu Wang, Wei Wei, Guang Dai, Yanning Zhang, and Jingdong Wang. Representation alignment for diffusion transformers without external components. In *The Fourteenth International Conference on Learning Representations*, 2026b.

Diederik Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 65484–65516. Curran Associates, Inc., 2023.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Felix Krause, Timy Phan, Ming Gui, Stefan Andreas Baumann, Vincent Tao Hu, and Björn Ommer. Tread: Token routing for efficient architecture-agnostic diffusion training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15703–15713, October 2025.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 122458–122483. Curran Associates, Inc., 2024.

Jiachen Lei, Keli Liu, Julius Berner, Haiming Yu, Hongkai Zheng, Jiahong Wu, and Xiangxiang Chu. There is no VAE: End-to-end pixel-space generative modeling via self-supervised pre-training. In *The Fourteenth International Conference on Learning Representations*, 2026.

Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning of latent diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18262–18272, October 2025.

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

Tianhong Li, Dina Katabi, and Kaiming He. Return of unconditional generation: A self-supervised representation generation method. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Tianhong Li, Qinyi Sun, Lijie Fan, and Kaiming He. Fractal generative models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.

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

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 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 *Computer Vision – ECCV 2024*, pp. 23–40. Springer Nature Switzerland, 2024.

Zehong Ma, Longhui Wei, Shuai Wang, Shiliang Zhang, and Qi Tian. Deco: Frequency-decoupled pixel diffusion for end-to-end image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 43600–43610, June 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

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

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Jaeyo Shin, Jiwook Kim, and Hyunjung Shim. Representation alignment for just image transformers is not easier than you think. In *European Conference on Computer Vision*, pp. 603–620. Springer, 2026.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

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 *The Fourteenth International Conference on Learning Representations*, 2026.

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.

Jiayan Teng, Wendi Zheng, Ming Ding, Wenyi Hong, Jianqiao Wangni, Zhuoyi Yang, and Jie Tang. Relay diffusion: Unifying diffusion process across resolutions for image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024.

Yuchuan Tian, Hanting Chen, Mengyu Zheng, Yuchen Liang, Chao Xu, and Yunhe Wang. U-REPA: Aligning diffusion u-nets to vits. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Michael Tschannen, André Susano Pinto, and Alexander Kolesnikov. Jetformer: An autoregressive generative model of raw images and text. In *International Conference on Learning Representations*, volume 2025, pp. 85995–86016, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. Pixnerd: Pixel neural field diffusion. In *International Conference on Learning Representations*, volume 2026, pp. 43559–43580, 2026a.

Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. Ddt: Decoupled diffusion transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 40633–40642, June 2026b.

Ziqiao Wang, Wangbo Zhao, Yuhao Zhou, Zekai Li, Zhiyuan Liang, Mingjia Shi, Xuanlei Zhao, Pengfei Zhou, Kaipeng Zhang, Zhangyang Wang, Kai Wang, and Yang You. REPA works until it doesn't: Early-stopped, holistic alignment supercharges diffusion training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15703–15712, June 2025.

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

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. Pixeldit: Pixel diffusion transformers for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14273–14282, June 2026.

Shuangfei Zhai, Ruixiang Zhang, Preetum Nakkiran, David Berthelot, Jiatao Gu, Huangjie Zheng, Tianrong Chen, Miguel Ángel Bautista, Navdeep Jaitly, and Joshua M. Susskind. Normalizing flows are capable generative models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 74348–74369. PMLR, 13–19 Jul 2025.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11941–11952. IEEE, 2023.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *International Conference on Learning Representations*, volume 2026, pp. 35791–35820, 2026a.

Guangting Zheng, Qinyu Zhao, Tao Yang, Fei Xiao, Zhijie Lin, Jie Wu, Jiajun Deng, Yanyong Zhang, and Rui Zhu. Farmer: Flow autoregressive transformer over pixels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 25730–25741, June 2026b.

Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Rui Zhu, Yingwei Pan, Yehao Li, Ting Yao, Zhenglong Sun, Tao Mei, and Chang Wen Chen. Sd-dit: Unleashing the power of self-supervised discrimination in diffusion transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8435–8445, June 2024.

Appendix

Table of Contents

A Related Work **16**

A.1 Pixel-space Diffusion 16

A.2 Representation Learning in Diffusion Models 16

B Additional Implementation Details **17**

C Additional Analysis **18**

C.1 Sensitivity to Alignment Loss Weights 18

C.2 Effect of Position-wise Feature Centering 18

C.3 Effect of Alignment Target Composition 19

C.4 Gradient Compatibility of Alignment Targets 20

C.5 Gradient Relationship between Scaffold and Anchor Alignment . . . 20

C.6 Computational Overhead 20

C.7 Generalization to DINOv3 Representations 21

D Qualitative Results **21**

E Limitations and Future Work **21**

A Related Work

A.1 Pixel-space Diffusion

To alleviate the computational burden of high-resolution pixel-space generation, early pixel-space diffusion studies adopted multi-stage or multi-resolution generation strategies. CDM (Ho et al., 2022) first generates low-resolution images and then sequentially applies multiple super-resolution diffusion models, while RDM (Teng et al., 2024) proposes relay-based generation that continuously extends the diffusion process across resolutions by transforming low-resolution diffusion states into their corresponding high-resolution states. Subsequent studies have moved toward more direct and efficient pixel-space generation by improving training strategies and architectural designs. Simple Diffusion (Hoogeboom et al., 2023) demonstrates that competitive performance on high-resolution image generation can be achieved with single-stage pixel diffusion by refining the noise schedule and architecture design. PixelFlow (Chen et al., 2025) proposes an end-to-end pixel-space flow model that efficiently performs generation across multiple resolutions through cascade flow modeling. PixNerd (Wang et al., 2026a) introduces a patch-wise neural field to model fine-grained details within patches while retaining the efficiency of large-patch Transformers. DiP (Chen et al., 2026) efficiently models global structure using a large-patch DiT and restores fine-grained local details with a lightweight Patch Detailer Head. DeCo (Ma et al., 2026) decouples the modeling of low-frequency semantics and high-frequency details, enabling more efficient learning of both global structure and fine-grained details. In addition, EPG (Lei et al., 2026) combines self-supervised representation pretraining with end-to-end generative fine-tuning to improve the training efficiency and performance of pixel-space generative modeling. PixelDiT (Yu et al., 2026) proposes a dual-level architecture in which a patch-level DiT captures global semantics, while a pixel-level DiT models local texture details. JiT (Li & He, 2026) demonstrates that high-dimensional pixel-space generation can be achieved with a simple Transformer architecture by directly applying a large-patch Transformer to raw pixels and performing clean-image prediction. Meanwhile, TarFlow (Zhai et al., 2025), FARMER (Zheng et al., 2026b), JetFormer (Tschannen et al., 2025), and Fractal Generative Models (Li et al., 2025) present alternative generative paradigms that directly model raw pixels, including normalizing-flow-based and autoregressive approaches.

A.2 Representation Learning in Diffusion Models

Recent studies have explored various ways to incorporate representation learning into diffusion model training to improve both training efficiency and generation performance. MaskDiT (Zheng et al., 2024) and SD-DiT (Zhu et al., 2024) promote representation learning by integrating masked modeling and self-supervised learning objectives into diffusion training, while TREAD (Krause et al., 2025) accelerates diffusion transformer training by jointly employing a representation learning objective and token routing. In contrast, SRA (Jiang et al., 2026b) and LayerSync (Haghighi et al., 2026) improve inter-layer representation learning by using representations formed within the diffusion model itself as guidance signals, rather than introducing an additional representation learning task. Another line of work leverages external representations obtained from pretrained vision models during diffusion model training. RCG (Li et al., 2024) uses semantic representations extracted from a self-supervised visual encoder as conditions for the generation process. REPA (Yu et al., 2025) introduces a representation alignment objective that aligns intermediate features of a diffusion Transformer with clean-image representations extracted from a pretrained vision encoder, thereby accelerating training convergence and improving generation performance. U-REPA (Tian et al., 2025) extends representation alignment to U-Net architectures, while REPA-E (Leng et al., 2025) enables end-to-end training of the VAE and latent diffusion model by backpropagating the alignment objective through the VAE. HASTE (Wang et al., 2025) shows that although representation alignment substantially accelerates early-stage training, it can constrain generative learning in later stages, and accordingly proposes a stage-wise strategy that terminates alignment after a certain point in training. iREPA (Singh et al., 2026) analyzes various pretrained vision encoders and demonstrates that patch-level spatial structure is more closely correlated with generation performance than the global semantic performance of the target representation, leading to an alignment strategy that strengthens spatial information transfer. PixelREPA (Shin et al., 2026) identifies an information mismatch that arises when applying REPA to pixel-space diffusion models and proposes a more suitable representation alignment approach based on a Masked Transformer Adapter. Our work focuses on the semantic granularity of pretrained representations and their distinct roles during

Table 6: Experimental configurations and implementation settings.

	JiT-B+HYDRA	JiT-L+HYDRA	JiT-H+HYDRA
Architecture			
Depth	12	24	32
Hidden dim	768	1024	1280
Heads	12	16	16
Image size		256×256	
Patch size		16×16	
Bottleneck	128	128	256
Dropout	0	0	0.2
In-context class tokens		32	
In-context start block	4	8	10
Alignment depth		(in-context start block) $- 1$	
Projector hidden dim.	768	1024	1280
Projector output dim.		768	
Training			
Epochs		100, 200 (ablation), 600	
Warmup epochs		5	
Optimizer		Adam, $(\beta_1, \beta_2) = (0.9, 0.95)$	
Batch size		1024	
Learning rate		2×10^{-4}	
Learning rate schedule		constant	
Weight decay		0	
EMA decay		$\{0.9996, 0.9998, 0.9999\}$	
Time sampler		$\text{logit}(t) \sim \mathcal{N}(\mu, \sigma^2)$, $\mu = -0.8$, $\sigma = 0.8$	
Noise scale		1.0	
Clip of $(1 - t)$ in division		0.05	
Class token drop (for CFG)		0.1	
Sampling			
ODE solver		Heun	
ODE steps		50	
Time steps		linear in $[0.0, 1.0]$	
CFG scale sweep range		$[1.0, 4.0]$	
CFG interval		$[0.1, 1.0]$	

diffusion training, motivating HYDRA to decouple the representation hierarchy into complementary alignment targets.

B Additional Implementation Details

Table 6 summarizes the architecture, training, and sampling configurations used in our experiments. Unless otherwise specified, we follow the original JiT configuration (Li & He, 2026) for the backbone architecture and training setup. All models are trained on NVIDIA RTX PRO 6000 Blackwell GPUs using bfloat16 mixed precision, with a global batch size of 1024. We use the Adam optimizer (Kingma & Ba, 2015) with $(\beta_1, \beta_2) = (0.9, 0.95)$ and apply a 5-epoch learning-rate warmup (Goyal et al., 2017). Training timesteps are sampled from a logit-normal distribution (Esser et al., 2024) with $\mu = -0.8$ and $\sigma = 0.8$. For computational efficiency, we replace the explicit attention implementation in the original JiT codebase with PyTorch's `F.scaled_dot_product_attention`, and use this implementation consistently across all experiments.

For representation alignment, we use DINOv2 ViT-B/14 (Oquab et al., 2024) as the pretrained visual encoder and keep its parameters frozen throughout training. Scaffold and Anchor alignment employ separate three-layer MLP projectors with adaLN-based timestep conditioning. The projector hidden

Table 7: Sensitivity analysis of the Scaffold and Anchor loss weights on JiT-B/16. FID is evaluated at 200 epochs.

λ_{S}	λ_{A}	**FID** $\downarrow$
0.05	0.05	3.79
0.06	0.04	3.71
0.07	0.03	**3.65**
0.08	0.02	3.68

Table 8: Ablation study of Position-wise Feature Centering for HYDRA with JiT-B/16, evaluated by FID (↓).

Epoch	**FID** $\downarrow$	
	w/o Centering	**w/ Centering**
50	13.90	**12.77**
200	3.75	**3.65**

dimension follows the corresponding JiT backbone dimension, i.e., 768, 1024, and 1280 for JiT-B/16, JiT-L/16, and JiT-H/16, respectively, while the output dimension is fixed to 768 to match the feature dimension of DINOv2 ViT-B/14. The alignment is applied to the Transformer block immediately preceding the in-context class-token insertion block.

We train the models for up to 600 epochs, while the ablation studies are conducted at 100 and 200 epochs to reduce computational cost. For sampling, we use the Heun method (Heun, 1900) with 50 linearly spaced steps over $[0, 1]$. We use classifier-free guidance (CFG) (Ho & Salimans, 2021) and apply guidance only over the interval $[0.1, 1.0]$, following limited-interval guidance (Kynkäänniemi et al., 2024). The CFG scale is selected by sweeping values in the range $[1.0, 4.0]$ with a step size of 0.1.

For the experiments reported in Tabs. 1 and 9, we set the alignment loss weight to 0.1 and use a three-layer MLP projector without adaLN conditioning. For the REPA and iREPA experiments, we likewise set the alignment loss weight to 0.1, following PixelREPA, to adapt the alignment supervision to the pixel-space diffusion setting. For iREPA, we use an MLP projector instead of the original convolutional projector to isolate the effect of the alignment features.

C Additional Analysis

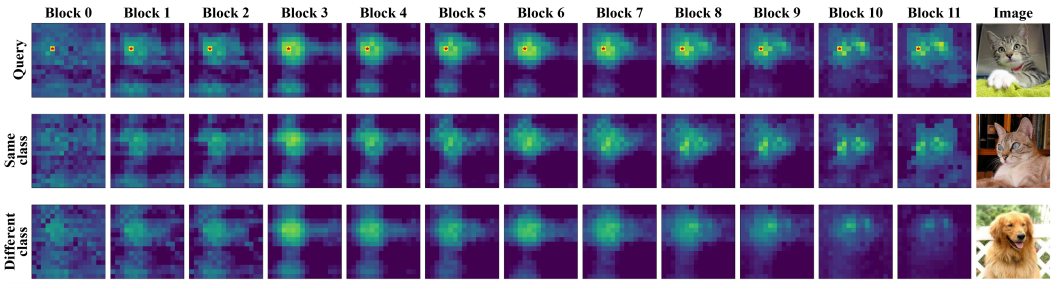
C.1 Sensitivity to Alignment Loss Weights

We further investigate the sensitivity of HYDRA to the relative weighting between Scaffold and Anchor alignment. As shown in Tab. 7, we vary λ_{S} and λ_{A} while keeping their sum fixed at 0.1. Among the evaluated settings, $\lambda_{\mathrm{S}} = 0.07$ and $\lambda_{\mathrm{A}} = 0.03$ achieves the best FID of 3.65 at 200 epochs. Nearby configurations also yield comparable performance, with FIDs of 3.71 and 3.68 for $(\lambda_{\mathrm{S}}, \lambda_{\mathrm{A}}) = (0.06, 0.04)$ and $(0.08, 0.02)$, respectively.

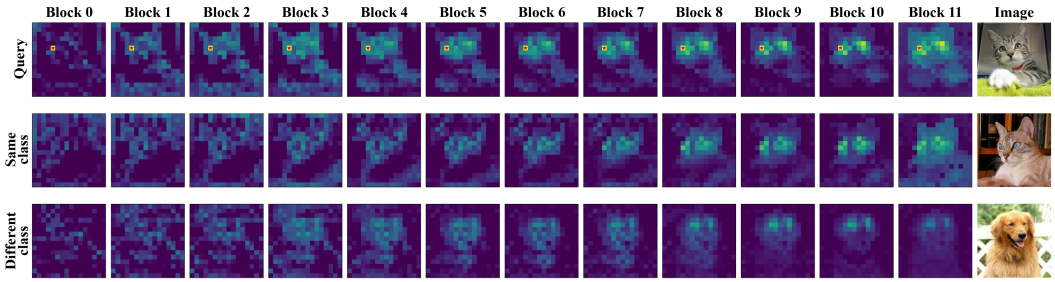
C.2 Effect of Position-wise Feature Centering

We further examine the effect of Position-wise Feature Centering on representation alignment. As shown in Tab. 8, removing feature centering consistently degrades generation performance, yielding FIDs of 13.90 and 3.75 at 50 and 200 epochs, respectively. In contrast, applying Position-wise Feature Centering improves the FID to 12.77 at 50 epochs and 3.65 at 200 epochs, demonstrating that suppressing position-dependent components in the alignment representations leads to more effective alignment.

We additionally provide a qualitative comparison of DINOv2 patch similarities before and after Position-wise Feature Centering in Fig. 6. Without centering, similarity maps exhibit recurring spa-



(a) Before Position-wise Feature Centering



(b) After Position-wise Feature Centering

Figure 6: Qualitative effect of Position-wise Feature Centering on DINOv2 patch representations across Transformer blocks. Each heatmap visualizes the cosine similarity between the query patch (red star) and patch representations from images of the same and different classes.

Table 9: FID comparison across different alignment targets using JiT-B/16 under the same experimental setting as Tab. 1. † denotes features after Position-wise Feature Centering.

Epoch	Alignment Target		
	Block $8^{\dagger}$,$9^{\dagger}$,$10^{\dagger}$	Block $9^{\dagger}$,$10^{\dagger}$	Block $9^{\dagger}$,$10^{\dagger}$,$11^{\dagger}$
50	11.64	**11.54**	11.95
100	4.63	**4.62**	4.74

tial patterns across different images, including those from different classes, indicating that patch similarity can be influenced by position-dependent components rather than visual content alone. After centering, these recurring spatial patterns are substantially suppressed, and the similarity maps more clearly reflect content-dependent correspondences between patches. Together with the quantitative results in Tab. 8, these observations support the use of Position-wise Feature Centering for constructing more semantically meaningful alignment targets.

C.3 Effect of Alignment Target Composition

We further analyze how the composition of the alignment target affects generation performance. Under the same experimental setting as Tab. 1, we compare combinations of neighboring DINOv2 blocks after applying Position-wise Feature Centering. As shown in Tab. 9, using Blocks 9, 10 consistently achieves the best performance, with FIDs of 11.54 and 4.62 at 50 and 100 epochs, respectively. Extending the target to Blocks 8, 9, 10 provides no additional improvement, while incorporating the final Block 11 degrades the FID to 11.95 and 4.74.

These results indicate that concentrating the alignment target on the late-intermediate representations of Blocks 9 and 10 is more effective than incorporating either an earlier block or the final block. Together with our semantic granularity analysis, this trend is consistent with Blocks 9 and 10 providing component-level representations suitable for Scaffold alignment.

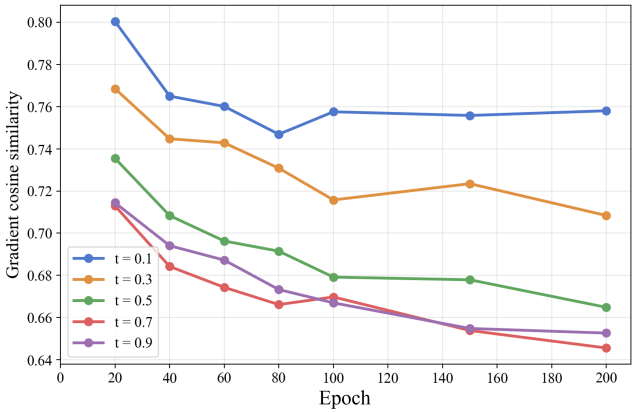


Figure 7: Gradient cosine similarity between Scaffold and Anchor alignment across training epochs at different timesteps.

C.4 Gradient Compatibility of Alignment Targets

We analyze the interaction between representation alignment objectives and the Flow Matching (FM) objective (Lipman et al., 2023; Liu et al., 2023) using gradient cosine similarity. For each alignment objective, we compute gradients with respect to the parameters of the Transformer block immediately preceding the corresponding alignment point and measure their cosine similarity with the gradients induced by the FM loss. This local measurement directly characterizes how the alignment supervision affects the optimization of the nearby backbone parameters.

A positive cosine similarity indicates directionally compatible updates between the alignment and FM objectives, whereas a negative value indicates gradient conflict. We evaluate REPA, iREPA, Scaffold alignment, and Anchor alignment across different timesteps and training stages. The resulting comparison is presented and discussed in Sec. 4.3.

C.5 Gradient Relationship between Scaffold and Anchor Alignment

To further examine the relationship between Scaffold and Anchor alignment, we measure the cosine similarity between the gradients induced by the two objectives throughout training at different timesteps. As shown in Fig. 7, their gradients remain positively aligned across all evaluated epochs and timesteps, where $t \in [0, 1]$ denotes the flow-matching timestep, indicating that the two objectives do not introduce conflicting optimization directions.

Nevertheless, the gradient similarity generally decreases as training progresses. This trend indicates that Scaffold and Anchor induce increasingly differentiated update directions over the course of training, despite remaining broadly compatible.

Together with the diminishing benefit of Scaffold alignment at later stages and the long-horizon ablation showing improved FID when Scaffold alignment is terminated early, this gradient behavior provides additional evidence that Scaffold and Anchor serve complementary temporal roles during training. Specifically, Scaffold provides useful structural supervision primarily in the early stage, while Anchor remains beneficial as training progresses.

C.6 Computational Overhead

We further analyze the computational overhead introduced by HYDRA. HYDRA employs two separate projectors for Scaffold and Anchor alignment, each containing 3.74M parameters, resulting in 7.48M additional parameters in total. Despite this dual-projector design, HYDRA requires 0.68 seconds per training iteration, compared with 0.64 seconds for REPA and 0.74 seconds for PixelREPA. Moreover, its projector parameter count remains substantially lower than that of PixelREPA (24.03M). These results show that HYDRA maintains a lightweight dual-projector design while introducing only modest additional training cost.

Table 10: Comparison of projector parameters and training time for representation alignment methods with JiT-B/16.

Method	Projector Params.	Training Time (sec/iter)
JiT	-	0.47
REPA	1.77M	0.64
PixelREPA	24.03M	0.74
HYDRA	3.74M $\times$ 2	0.68

Table 11: FID comparison of pretrained visual encoders for HYDRA with JiT-B/16 after 200 training epochs.

Visual Encoder	Epoch	FID (↓)
DINOv2-B	200	3.65
DINOv3-B	200	3.76

C.7 Generalization to DINOv3 Representations

We further investigate whether HYDRA can benefit from representations provided by a different pretrained visual encoder. Specifically, we replace DINOv2-B with DINOv3-B (Siméoni et al., 2025) while keeping the JiT-B/16 backbone and all remaining training configurations unchanged. The same block-selection strategy and Scaffold/Anchor configuration used in HYDRA are retained for DINOv3-B. Table 11 reports the FID after 200 training epochs. Although DINOv3-B yields slightly lower performance than DINOv2-B, it still provides effective alignment supervision and achieves competitive generation quality. This result suggests that HYDRA is not tied to a particular pretrained encoder and can be readily applied to representations from different visual backbones.

D Qualitative Results

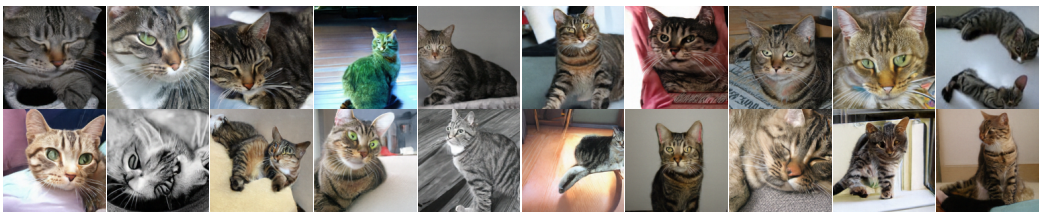
We provide additional uncurated samples on ImageNet 256×256 to qualitatively evaluate HYDRA across different model scales. Figures 8–9, 10–11, and 12–13 show samples generated by JiT-B/16 + HYDRA, JiT-L/16 + HYDRA, and JiT-H/16 + HYDRA, respectively. For each model, we use the same CFG setting as that used to obtain the corresponding FID reported in Tab. 2, ensuring that the qualitative results reflect the same generation setting as the quantitative evaluation.

E Limitations and Future Work

A limitation of our work is that HYDRA is evaluated primarily with DINOv2 as the pretrained visual encoder and on ImageNet-1K at 256×256 resolution. While our results suggest that different levels of the DINOv2 representation hierarchy provide complementary alignment targets, it remains unclear whether the same semantic granularity and training dynamics arise with other encoders or datasets. In particular, we terminate Scaffold alignment at a fixed point of 150 epochs. The stage at which its supervision becomes less useful may depend on the encoder's representation hierarchy, the dataset, and the training configuration; moreover, an epoch count corresponds to different numbers of optimization steps across datasets. Consequently, the optimal stopping point may need to be adjusted for each setting. Future work will investigate HYDRA across diverse pretrained visual encoders and datasets, examine how their representation hierarchies affect the roles of Scaffold and Anchor alignment, and develop a stopping criterion based on observed training dynamics rather than a fixed epoch.



class 012: house finch, linnet, Carpodacus mexicanus



class 281: tabby, tabby cat



class 207: golden retriever



class 033: loggerhead, loggerhead turtle, Caretta caretta



class 130: flamingo



class 985: daisy

Figure 8: Uncurated ImageNet 256×256 samples from JiT-B/16 + HYDRA for the specified classes, generated with CFG=3.0 (reported FID: 3.15).



class 817: sports car, sport car



class 779: school bus



class 671: mountain bike, all-terrain bike, off-roader



class 402: acoustic guitar



class 409: analog clock

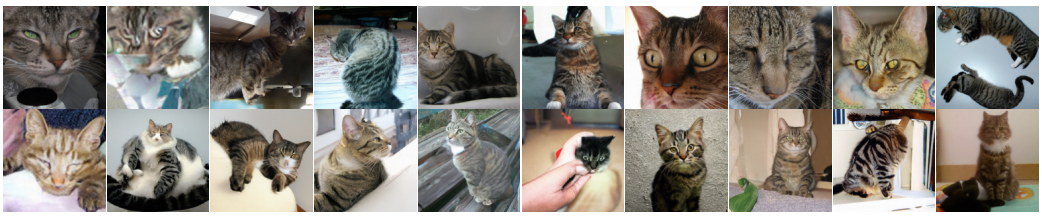


class 765: rocking chair, rocker

Figure 9: Uncurated ImageNet 256×256 samples from JiT-B/16 + HYDRA for the specified classes, generated with CFG=3.0 (reported FID: 3.15).



class 012: house finch, linnet, Carpodacus mexicanus



class 281: tabby, tabby cat



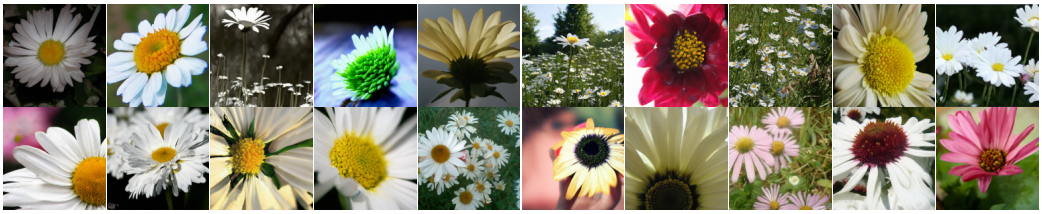
class 207: golden retriever



class 033: loggerhead, loggerhead turtle, Caretta caretta



class 130: flamingo



class 985: daisy

Figure 10: Uncurated ImageNet 256×256 samples from JiT-L/16 + HYDRA for the specified classes, generated with CFG=2.4 (reported FID: 2.06).



class 817: sports car, sport car



class 779: school bus



class 671: mountain bike, all-terrain bike, off-roader



class 402: acoustic guitar



class 409: analog clock



class 765: rocking chair, rocker

Figure 11: Uncurated ImageNet 256×256 samples from JiT-L/16 + HYDRA for the specified classes, generated with CFG=2.4 (reported FID: 2.06).



class 012: house finch, linnet, Carpodacus mexicanus



class 281: tabby, tabby cat



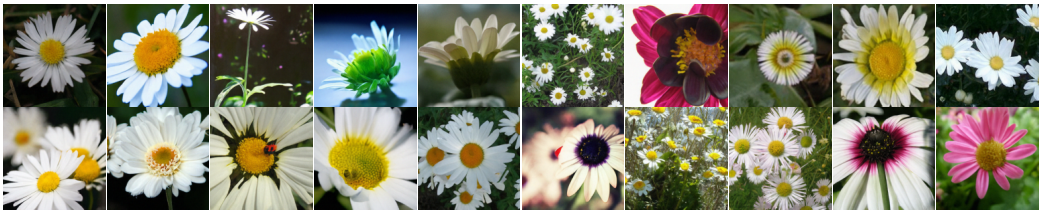
class 207: golden retriever



class 033: loggerhead, loggerhead turtle, Caretta caretta



class 130: flamingo



class 985: daisy

Figure 12: Uncurated ImageNet 256×256 samples from JiT-H/16 + HYDRA for the specified classes, generated with CFG=2.2 (reported FID: 1.74).



class 817: sports car, sport car



class 779: school bus



class 671: mountain bike, all-terrain bike, off-roader



class 402: acoustic guitar



class 409: analog clock



class 765: rocking chair, rocker

Figure 13: Uncurated ImageNet 256×256 samples from JiT-H/16 + HYDRA for the specified classes, generated with CFG=2.2 (reported FID: 1.74).