AHPA: Adaptive Hierarchical Prior Alignment for Diffusion Transformers

Ruibin Min[1,2], **Yexin Liu**[1,*], **Aimin Pan**[3], **Changsheng Lu**[1], **Jiafei Wu**[5,6], **Kelu Yao**[4], **Xiaogang Xu**[5,6], **Harry Yang**[1,†]

[1]The Hong Kong University of Science and Technology
[2]Sun Yat-sen University [3]VNET Group [4]Zhejiang Lab
[5]School of Software Technology, Zhejiang University
[6]Innovation and Management Center,
School of Software Technology (Ningbo), Zhejiang University

[*]Project Leader [†]Corresponding Author
minrb@mail2.sysu.edu.cn



Figure 1: **High-fidelity image generated by SiT-XL/2 with AHPA alignment.**

Abstract

Representation alignment has recently emerged as an effective paradigm for accelerating Diffusion Transformer training. Despite their success, existing alignment methods typically impose a fixed supervision target or a fixed alignment granularity throughout the entire denoising trajectory, whether the guidance is provided by external vision encoders, internal self-representations, or VAE-derived features. We argue that such timestep-agnostic alignment is suboptimal because the useful granularity of representation supervision changes systematically with the signal-to-noise ratio. In high-noise regimes, diffusion models benefit more from coarse semantic and layout-level anchoring, whereas in low-noise regimes, the training signal should emphasize spatially detailed and structurally faithful refinement. This non-stationary alignment behavior creates a representational mismatch for static single-level supervisors. To address this issue, we propose **Adaptive Hierarchical Prior Alignment (AHPA)**, a lightweight alignment framework that exploits the

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

hierarchical representations naturally embedded in the frozen VAE encoder. Instead of using only a single compressed latent as the alignment target, AHPA extracts multi-level VAE features that provide complementary priors ranging from local geometry and spatial topology to coarse semantic layout. A timestep-conditioned Dynamic Router adaptively selects and weights these hierarchical priors along the denoising trajectory, thereby synchronizing the alignment granularity with the model's evolving training needs. Extensive experiments show that AHPA improves convergence and generation quality over baselines and incurs no additional inference cost while avoiding external encoder supervision during training. Our code is available at https://github.com/UIOSN/AHPA.

1 Introduction

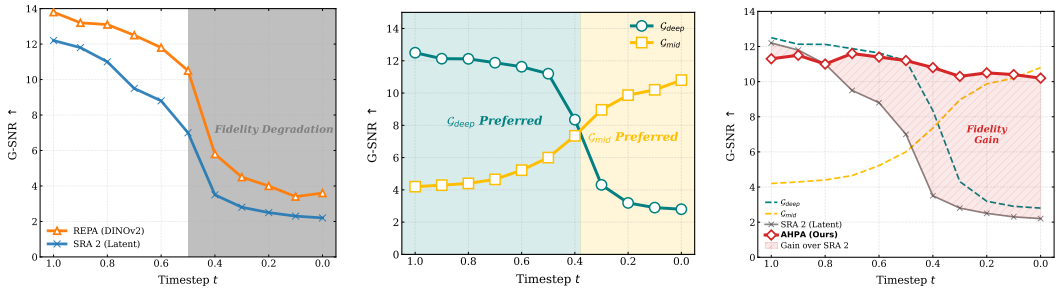
Recent advances in diffusion models [10, 30] have substantially improved high-fidelity image generation, with Diffusion Transformers [22, 18] emerging as a scalable backbone for generative modeling. However, training such models from scratch remains computationally demanding, often requiring long optimization schedules before the denoising network learns semantically meaningful and structurally coherent representations. Representation alignment has recently been introduced as an effective paradigm for accelerating this process: by aligning intermediate transformer features with informative visual representations, the denoising backbone can acquire stronger semantic and structural priors during training [36, 29, 33, 3]. For example, REPA-style methods transfer knowledge from external vision encoders such as DINOv2 [21], while recent variants further improve the alignment objective through structural constraints, adversarial matching, or improved representation regularization [29, 3].

Despite these advances, existing alignment strategies are still largely built upon a fixed-target formulation. Most methods differ in the source of supervision—external vision encoders, internal self-representations, or VAE-derived features—but typically impose a fixed type of alignment signal throughout the denoising trajectory. External-teacher-based methods provide strong semantic guidance but introduce additional encoder forward passes, memory usage, and training complexity. Some alternatives such as SRA [13] and DUPA [23] reduce the dependence on external teachers through self-distillation or dual-path consistency, yet may still incur extra training computation. Recent VAE-based methods such as SRA2 [32] further reduce the teacher cost by using VAE-derived supervision; however, relying on a single compressed latent representation provides limited access to the fine-grained intermediate structures that are important for effective alignment [35, 29].

A more intrinsic limitation lies in the timestep-agnostic nature of existing alignment objectives. Diffusion training is inherently non-stationary: as the signal-to-noise ratio evolves, the useful granularity of representation supervision changes accordingly. In high-noise regimes, the denoising network benefits from coarse semantic and layout-level anchoring, while in low-noise regimes, it requires spatially detailed and structurally faithful guidance. A fixed single-level supervisor therefore cannot remain equally suitable across all timesteps, resulting in a granularity mismatch between the alignment target and the denoising state. To analyze this, we introduce the Gradient Signal-to-Noise Ratio (G-SNR, Sec. 3) to measure the alignment health along the trajectory, which reveals that static methods suffer from declining directional consistency and gradient interference. This mismatch limits the acceleration potential of static representation alignment.

Motivated by this observation, we propose **Adaptive Hierarchical Prior Alignment (AHPA)**, which reformulates representation alignment from static target matching into timestep-adaptive hierarchical prior scheduling. Instead of treating the frozen VAE encoder [26] merely as a latent compression module, AHPA re-examines it as a multi-scale repository of hierarchical priors. Intermediate VAE features naturally encode complementary information at different levels, ranging from local geometry and spatial topology to coarse layout-level structures. To exploit these priors effectively, AHPA introduces a lightweight timestep-conditioned dynamic router that adaptively selects and weights hierarchical VAE features according to the current denoising stage. This design synchronizes the granularity of alignment supervision with the evolving needs of diffusion training, while avoiding heavyweight external teacher encoders and introducing no additional inference-time cost.

Our main contributions are summarized as follows:



(a) Diagnostic Probing (b) Complementarity Discovery (c) Proposal & Verification

Figure 2: **Quantifying the non-stationary alignment requirement.** (a) Diagnostic probing reveals that static baselines suffer from catastrophic fidelity degradation as $t \to 0$. (b) Hierarchical features ($\mathcal{G}_{deep}$ and $\mathcal{G}_{mid}$) exhibit significant phase complementarity. (c) Our AHPA adaptively bridges these gaps to maintain a high-level G-SNR envelope throughout the full trajectory.

- **Non-stationary Alignment Analysis.** We identify that the effective granularity of representation alignment varies across denoising timesteps, and show that fixed single-level supervision can lead to a representational mismatch during diffusion training.
- **Hierarchical VAE Prior Mining.** We reveal that intermediate layers of the frozen VAE encoder provide a natural multi-scale prior bank, offering complementary guidance from local structural details to coarse layout-level representations.
- **Adaptive and Efficient Alignment.** We propose AHPA, a timestep-aware alignment framework equipped with a lightweight dynamic router, achieving adaptive hierarchical supervision with zero additional inference cost and reduced dependence on heavyweight external teachers.

2 Related Work

Generative Models and Diffusion Transformers. Image generation has witnessed a transition from pixel-space denoising models, such as DDPM [10] and DDIM [30], to Latent Diffusion Models (LDMs) [26] that operate in compressed latent spaces. Architecturally, while early frameworks relied on U-Net backbones [10, 26], modern paradigms have shifted towards Transformer-based architectures like DiT [22] and SiT [18], which leverage self-attention mechanisms for superior spatial modeling [22]. Despite these advances, diffusion transformers typically require extensive training iterations to achieve convergence [29, 33].

Representation Alignment and Entanglement. A prominent direction to expedite training is leveraging discriminative priors from pretrained vision encoders. REPA [36] pioneered this by aligning intermediate diffusion features with representations from external teachers like DINOv2 [21]. Building on this, iREPA [29] and SARA [3] further enhanced alignment quality by focusing on spatial structures or introducing adversarial distribution matching. While effective, these methods are teacher-dependent, requiring external models that introduce significant memory overhead and inherent domain biases [29, 33]. Alternatively, REG [33] proposes entangling high-level class tokens directly into the diffusion process for semantic guidance, yet still relies on fixed category embeddings.

Self-Supervised Alignment and its Bottlenecks. To eliminate dependence on external teachers, a new frontier of autonomous alignment has emerged. SRA [13] and SRA 2 [32] propose self-guidance by distilling features from either an EMA-updated teacher model or the final output of a frozen VAE encoder. Similarly, DUPA [23] achieves self-alignment by enforcing consistency between dual noising paths of the same image. However, they rely on static guidance, utilizing a monolithic feature target throughout the entire process. Such "time-invariant" strategies ignore the non-stationary alignment requirement inherent in diffusion models. AHPA fills this gap by adaptively routing hierarchical VAE priors, ensuring the guidance granularity is synchronized with the model's shifting receptivity along the denoising trajectory.

3 Motivation: Identifying and Resolving Representational Mismatch

It is widely recognized that the generative trajectory of diffusion models is inherently multi-stage [10, 20, 4], typically transitioning from global structural anchoring in high-noise regimes to local texture refinement as $t \to 0$. This progressive nature raises a critical question: *can a static supervisor effectively guide a non-stationary denoising process?* We hypothesize that a static approach, by utilizing a fixed-granularity prior, inevitably creates a mismatch with the model's evolving representational needs. To rigorously investigate this mismatch and its impact on alignment health, we introduce the **Gradient Signal-to-Noise Ratio (G-SNR)**. Let $g_t = \nabla_\theta \mathcal{L}_{align}$ be the gradient elicited by the alignment objective at timestep t; we define:

$$\text{G-SNR}(t) = \frac{\|\mathbb{E}[g_t]\|_2^2}{\text{Tr}(\text{Var}(g_t))} \tag{1}$$

G-SNR measures signal purity: a high G-SNR indicates directionally consistent guidance across the mini-batch, whereas a collapse implies severe gradient interference. To provide a reliable estimate, we compute G-SNR(t) using the variance across per-sample gradients within each mini-batch. This formulation renders the metric dimensionless and invariant to absolute gradient magnitudes, allowing for a consistent assessment of alignment quality across different timesteps and layers. A formal discussion on the variance estimator and its theoretical grounding is provided in Appendix F.

Diagnostic Probing: The Collapse of Static Baselines. We first quantify the gradient properties of existing acceleration paradigms, such as REPA [36] and SRA 2 [32]. As illustrated in Fig. 2a, while these static methods perform well in early stages, their G-SNR undergoes a dramatic collapse as $t \to 0$, indicating that the guidance becomes directionally inconsistent and suffers from gradient interference. This degradation reveals a bottleneck: a single, fixed representational granularity cannot adapt to the non-stationary SNR environment throughout the denoising trajectory. Consequently, we posit that the key to sustained acceleration lies in a multi-scale feature library that can dynamically match the model's evolving requirements across different phases. This motivates us to explore the internal hierarchy of the VAE encoder, leveraging its multi-scale feature space as a natural source of adaptive 'blueprints' that correspond to different denoising stages.

The Complementarity Discovery: Phase-wise Synergies. We perform a full-spectrum probe of the VAE encoder's hierarchical features, Our analysis reveals that while the shallow layers consistently yield low G-SNR due to excessive pixel-level noise and a lack of semantic coherence, a significant phase complementarity emerges between the mid-level ($\mathcal{G}_{mid}$) and deep-level ($\mathcal{G}_{deep}$) features. As illustrated in Fig. 2b, $\mathcal{G}_{deep}$ maintains higher directional consistency during the high-noise regimes ($t > 0.4$), whereas $\mathcal{G}_{mid}$ takes over as the more reliable guide for structural details as $t \to 0$. This crossover confirms that while no single layer can dominate the entire trajectory, their synergy offers a potential path to sustaining high-purity guidance.

Proposal & Verification: AHPA Trajectory Optimization. Driven by these insights, we propose AHPA, which utilizes a dynamic router to switch or fuse these complementary features in real-time based on t. Experimental verification (Fig. 2c) confirms that this adaptive scheduling successfully maintains a high-level G-SNR envelope across the full trajectory. This global enhancement in gradient quality directly translates into faster manifold convergence and superior image generation quality, effectively capturing the fidelity gain previously lost to static methods.

4 Method

4.1 Adaptive Hierarchical Prior Alignment (AHPA)

The core of AHPA is that the diffusion process is inherently non-stationary, requiring a dynamic shift in guidance granularity as the signal-to-noise ratio (SNR) evolves. We propose to align the internal representations of the Transformer with a multi-scale features extracted from a frozen VAE encoder, ensuring the guidance remains synchronized with the model's time-varying representational needs. The overall pipeline of AHPA is visualized in Fig. 3.

4.2 Mining Hierarchical Representation Blueprints

We leverage the frozen VAE encoder $\mathcal{E}$ as a hierarchical feature extractor to provide the diffusion backbone with a set of multi-scale "blueprints." We partition the VAE encoder layers into four groups

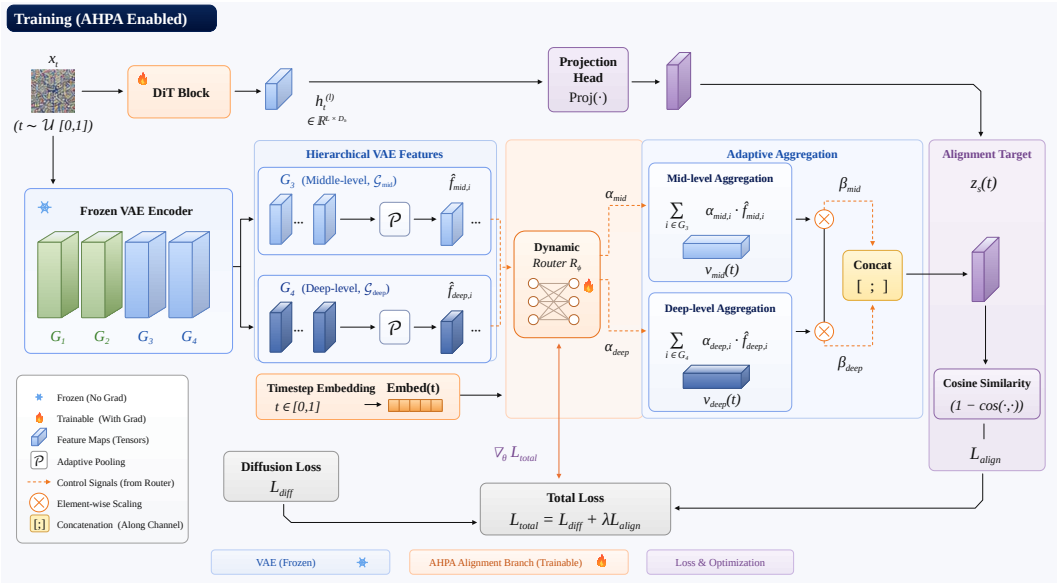


Figure 3: **Overview of AHPA.** AHPA extracts multi-scale hierarchical priors from a frozen VAE encoder. A timestep-conditioned dynamic router $\mathcal{R}_\phi$ adaptively schedules these priors (α, β) to align with the DiT backbone's evolving needs, incurring zero inference overhead.



Figure 4: **Group-averaged Feature Visualization via PCA.** From top to bottom: Original images, Middle-level blueprints ($\mathcal{G}_{mid}$), and Deep-level blueprints ($\mathcal{G}_{deep}$). The visualization demonstrates the consistent structural-to-semantic abstraction across diverse categories.

$(G_1 \dots G_4)$ based on their channel dimensions and spatial resolutions (detailed in App. A). While earlier stages (G_1, G_2) primarily encapsulate low-level pixel intensities, they exhibit lower alignment purity in our diagnostics (as shown in Sec. 3), likely due to the prevalence of stochastic noise in early denoising phases. In contrast, the selected groups G_3 and G_4 provide more stable and semantically meaningful guidance for the transformer backbone, further evidenced in Sec.5.2.2. As visualized in Figure 4, the selected VAE stages exhibit a distinct functional stratification across diverse samples:

Middle-level Group ($\mathcal{G}_{mid}$): Derived from G_3, this group represents a transitional stage of abstraction, primarily capturing spatial topologies and part-level relationships that provide a structural scaffold for the denoising process.

Deep-level Group ($\mathcal{G}_{deep}$): Derived from G_4, this group resides at the peak of the encoder hierarchy, encapsulating broader compositions and global semantic abstractions.

Let $\mathbf{f}_{g,i}$ denote the feature map of the i-th layer within group $g \in \{mid, deep\}$. To resolve the spatial discrepancy between VAE stages and the Transformer's patch grid, we apply an adaptive pooling operator $\mathcal{P}(\cdot)$ to unify all features to a fixed latent grid $S \times S$ (e.g., 16×16): $\hat{\mathbf{f}}_{g,i} = \mathcal{P}(\mathbf{f}_{g,i}) \in$

$\mathbb{R}^{C_i \times S \times S}$. By flattening $\hat{\mathbf{f}}_{g,i}$ into a sequence of length $L = S^2$, each pixel is precisely aligned with the corresponding Transformer token.

4.3 Collaborative Dynamic Routing

Design Choice: Timestep Conditioning. We introduce a Dynamic Router $\mathcal{R}_\phi$. While content-adaptive gating is intuitive, we prioritize the timestep $t \in [0, 1]$ as the primary control variable. This is informed by our finding that the shift in demand for hierarchical priors is a trajectory-wide phenomenon dictated by global SNR evolution.

Adaptive Weighting and Aggregation. We employ a linear projection $\text{Embed}(t)$ to predict intra-group aggregation logits $\hat{\boldsymbol{\alpha}}(t)$ and inter-group coordination logits $\hat{\boldsymbol{\beta}}(t)$:

$$\left\{\hat{\boldsymbol{\alpha}}_{mid}(t), \hat{\boldsymbol{\alpha}}_{deep}(t), \hat{\boldsymbol{\beta}}(t)\right\} = \mathcal{R}_\phi(\text{Embed}(t)) \tag{2}$$

The aggregated feature $\mathbf{v}_g(t)$ for each group is computed via softmax-normalized summation:

$$\mathbf{v}_g(t) = \sum_{i=1}^{K_g} \frac{\exp(\hat{\alpha}_{g,i}(t))}{\sum_{j=1}^{K_g} \exp(\hat{\alpha}_{g,j}(t))} \cdot \hat{\mathbf{f}}_{g,i} \tag{3}$$

To balance middle vs. deep-level contributions, we obtain scaling factors $\boldsymbol{\beta}(t) = \text{Sigmoid}(\hat{\boldsymbol{\beta}}(t))$. The final alignment target $\mathbf{z}_s(t)$ is constructed as:

$$\mathbf{z}_s(t) = \text{Concat}\left[\beta_{mid}(t) \cdot \mathbf{v}_{mid}(t),\ \beta_{deep}(t) \cdot \mathbf{v}_{deep}(t)\right] \tag{4}$$

Concatenation preserves representational integrity in distinct channel subspaces, allowing the backbone to adaptively resolve inter-group interactions through the subsequent projection layer.

4.4 Joint Optimization Objective

The Transformer θ and router ϕ are optimized jointly. We align the hidden states $h^{(l)}$ from an early block with $\mathbf{z}_s(t)$ using a convolutional projection head $\text{Proj}(\cdot)$. We employ a Cosine Similarity Loss to ensure invariance to activation scales:

$$\mathcal{L}_{align}(\theta, \phi) = \mathbb{E}_{t,x}\left[1 - \frac{\text{Proj}(h^{(l)}; \theta) \cdot \mathbf{z}_s(t; \phi)}{\left\|\text{Proj}(h^{(l)}; \theta)\right\|_2 \cdot \left\|\mathbf{z}_s(t; \phi)\right\|_2}\right] \tag{5}$$

The total objective is $\mathcal{L}_{total} = \mathcal{L}_{diff} + \lambda \mathcal{L}_{align}$. At inference, the alignment branch is discarded, yielding performance gains with **zero extra cost**.

5 Experiments and Analysis

5.1 Experimental Setup

We evaluate AHPA across SiT-B/2, L/2, and XL/2 scales. Following the hierarchical partitioning established in Sec. 4.2, the dynamic router $\mathcal{R}_\phi$ is implemented as a 4-layer MLP that adaptively modulates the contributions of $\mathcal{G}_{mid}$ and $\mathcal{G}_{deep}$. The projection head $\text{Proj}(\cdot)$ consists of a single 3×3 convolutional layer to preserve spatial inductive biases. All models are trained for 400k iterations on the ImageNet 256×256 dataset using NVIDIA A100 GPUs. Detailed architectural configurations, hyperparameter schedules, and evaluation protocols are documented in Appendix A.

5.2 Ablation Studies

Experiments in this section are conducted on SiT-B/2. Extended ablations on routing strategies, positional optimality, and component-wise configurations are detailed in Appendix E.

5.2.1 Determining the Optimal Alignment Stage

Table 1: **Ablation on alignment block l.**

l	2	**3**	4	8	Multi
FID ↓	31.4	**30.7**	31.6	32.2	32.9
IS ↑	46.5	**48.7**	47.9	44.2	47.2

We evaluate the SiT-B/2 backbone's receptivity across different stages using G4 as guidance source. As shown in Table 1, performance follows a U-shaped trend, with block $l = 3$ emerging as the optimal entry point. We observe that early-stage alignment ($l = 2$) lacks sufficient abstraction, while late-stage integration ($l \geq 8$) interferes with the model's autonomous denoising. Notably, multi-stage guidance (simultaneously at blocks $\{3, 4, 8\}$) yields inferior results (32.9 FID), suggesting that redundant constraints introduce optimization conflicts. Consequently, we fix alignment at the 25th percentile of model depth to consistently target the most flexible representational phase.

5.2.2 Identifying the Optimal Guidance Source

Table 2: **VAE groups ablation** (Static, $l = 3$).

Source	IS ↑	FID ↓	sFID ↓
Vanilla SiT-B/2	43.7	33.0	6.46
Group 1	41.1	34.5	6.69
Group 2	41.8	35.1	6.38
Group 3 ($\mathcal{G}_{mid}$)	48.9	32.6	6.45
Group 4 ($\mathcal{G}_{deep}$)	48.7	30.7	6.26
G3 ∪ G4	**51.6**	**29.7**	**6.04**

We investigate the effectiveness of VAE hierarchical groups at $l = 3$. Table 2 reveals a correlation between semantic depth and generative quality. Aligning with low-level groups (G1, G2) introduces high-frequency noise that distracts optimization. Conversely, mid-to-deep groups (G3, G4) act as effective blueprints, where their combination yields a significant synergy in both structural consistency and semantic accuracy. Based on these findings, we utilize G3 and G4 as our primary alignment sources for AHPA.

5.3 System-level Comparisons

5.3.1 Performance and Efficiency Analysis

We evaluate AHPA's scalability across multiple model scales (Table 3) and its competitive standing among current state-of-the-art methods (Table 4). Experiments on ImageNet 512×512 and additional qualitative comparisons are detailed in Appendix G and I. Our analysis reveals three core insights:

Preserving Acceleration Paradigm. Consistent with the alignment paradigm pioneered by REPA [36], AHPA successfully recovers the nearly **17**× training reduction. By matching the 7M-iteration vanilla baseline at only 400K steps (8.1 vs. 8.3 FID), our results demonstrate that intrinsic VAE hierarchies provide sufficient guidance to maintain the high convergence speeds characteristic of representation alignment, even in the absence of external teachers.

Superiority of Adaptive Routing. AHPA outperforms the latent-based baseline SRA 2 across all scales, with the performance gap widening as model capacity increases. This validates our core hypothesis: while static alignment provides a coarse prior, it fails to accommodate the non-stationary representational needs of larger transformers. By adaptively scheduling hierarchical priors, AHPA resolves the representational mismatch and information collapse that bottleneck previous methods.

Competitiveness against Heavyweight Teachers. AHPA demonstrates strong competitiveness against methods supervised by massive external encoders. Specifically, at 400K iterations, it outperforms REPA in image realism (8.1 vs. 8.5 FID) and achieves results closely approaching REG at 800 epochs (1.40 vs. 1.36 FID). Notably, AHPA maintains superior structural fidelity (4.12 vs. 4.70 sFID against REPA). These results, achieved with negligible overhead (+5% GFLOPs), prove that adaptively routed intrinsic priors can effectively substitute for external vision learners, eliminating the trade-off between efficiency and synthesis quality.

5.3.2 Text-to-Image Generation on MS-COCO

Table 5: **T2I on MS-COCO.** Evaluated using MMDiT (SDE, NFE=250).

Method	FID ↓	PS ↑
MMDiT	5.30	20.54
+ REPA	4.74	20.90
+ SRA 2	4.91	20.76
+ AHPA	**4.52**	**21.33**

To verify the generalization of AHPA, we conduct text-to-image (T2I) generation experiments on MS-COCO using the MMDiT backbone. As shown in Table 5, under SDE sampling (NFE=250), AHPA achieves a FID of **4.52** and a PickScore (PS) of **21.33**. Despite relying solely on intrinsic VAE features, AHPA not only outperforms previous self-alignment

Table 3: **Comprehensive Performance Comparison on ImageNet** 256×256. We report a holistic evaluation across three model scales (Base, Large, and Extra Large). **w/o Ext.** indicates whether the method is standalone and does *not* rely on external pre-trained encoders. **Iter.** denotes the number of training iterations. Bold indicates the best performance within each model scale.

Scale	Method	w/o Ext.	Iter.	FID ↓	sFID ↓	IS ↑	Prec. ↑	Rec. ↑
SiT-B/2	Vanilla SiT	✓	400K	33.0	6.46	43.7	0.53	0.63
	SRA 2 [32]	✓	400K	31.3	6.42	47.0	0.55	0.63
	REPA [36] (DINOv2-B)	✗	400K	27.2	6.80	**54.9**	0.57	**0.64**
	AHPA (Ours)	✓	400K	**25.5**	**5.78**	52.3	**0.59**	0.63
SiT-L/2	Vanilla SiT	✓	400K	18.8	5.29	72.0	0.64	**0.64**
	SRA 2 [32]	✓	400K	16.2	5.11	77.4	0.65	0.62
	REPA [36] (DINOv2-B)	✗	400K	12.6	5.25	**95.0**	0.67	**0.64**
	AHPA (Ours)	✓	400K	**10.5**	**4.97**	92.1	**0.69**	0.62
SiT-XL/2	Vanilla SiT	✓	400K	17.2	5.95	83.1	0.64	0.63
	Vanilla SiT	✓	7M	8.3	6.32	**131.7**	0.68	**0.67**
	SRA 2 [32]	✓	400K	13.8	5.10	94.3	0.65	0.64
	REPA [36] (DINOv2-B)	✗	400K	8.5	5.14	110.6	0.69	0.64
	AHPA (Ours)	✓	400K	**8.1**	**4.90**	112.0	**0.71**	0.62

Table 4: **State-of-the-Art Comparison on ImageNet** 256×256 **with CFG.** Bold indicates the best results among methods without external dependencies.

Method	Epochs	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑	w/o Ext.
Latent diffusion, Transformer							
DiT-XL/2 [22]	1400	2.27	4.60	278.2	**0.83**	0.57	✓
+ iSSD [19]	1400+0.8	2.02	4.22	250.0	0.81	0.60	✗
MaskDiT [39]	1600	2.28	5.67	276.6	0.80	0.61	✗
SD-DiT [42]	480	3.23	-	-	-	-	✗
SiT-XL/2 [18]	1400	2.06	4.50	270.3	0.82	0.59	✓
+ REPA [36]	800	1.42	4.70	311.4	0.80	0.63	✗
+ REG [33]	800	1.36	4.25	299.4	0.77	0.66	✗
+ SRA [13]	800	1.58	4.65	305.7	0.80	0.63	✓
+ SRA 2 [32]	800	1.52	4.63	316.2	0.82	0.62	✓
+ AHPA (Ours)	100	2.15	4.38	265.4	0.81	0.58	✓
+ AHPA (Ours)	200	1.78	4.26	291.8	0.82	0.61	✓
+ AHPA (Ours)	400	1.56	4.15	308.2	0.81	0.63	✓
+ AHPA (Ours)	800	**1.40**	**4.12**	**317.5**	0.81	**0.64**	✓

baselines (SRA 2) but also surpasses methods utilizing heavy-weight external vision learners (REPA) in both image realism and text-alignment. These results confirm that adaptively scheduled hierarchical priors provide robust guidance for T2I task. Detailed experimental setups and ODE sampler results are provided in Appendix D.

5.3.3 Training Computational Efficiency

We evaluate the training overhead by pre-computing VAE hierarchical features offline, eliminating real-time encoding latency. As shown in Table 6, AHPA demonstrates superior efficiency over teacher-based methods. While REPA incurs substantial overhead (up to +73% GFLOPs) due to heavy external encoders, AHPA is *external-free*. By employing a lightweight MLP router and a single-layer projection head, AHPA adds only a negligible **5%** to total GFLOPs and 2.5M parameters. Our method maintains a high throughput of 7.4 Iter/s, nearly on par with the baseline, providing a frugal yet effective path for representation alignment. Detailed comparison is presented in Appendix J.

Table 6: **Cost comparison on ImageNet** 256^2. TS: Training Speed (batch size 256, A100 GPUs).

Method	#EFP (M)	TS ↑	GFLOPs ↓
Baseline (SiT-XL/2)	0	8.1	114.5
+ REPA [36]	94	6.3	138.5
+ SRA [13]	483	5.1	197.6
+ SRA 2 [32]	18	7.2	118.6
+ AHPA (Ours)	**2.5**	**7.4**	**119.9**

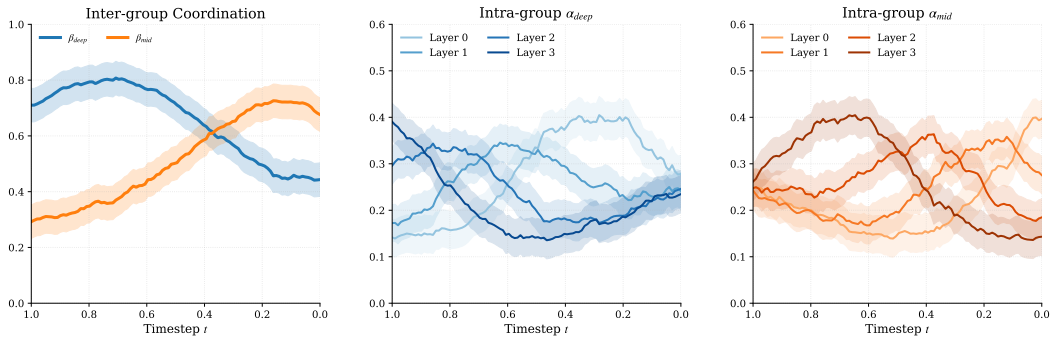


Figure 5: **Mechanistic analysis of AHPA's dynamic routing policy.** (Left) The inter-group weights β exhibit a clear transition from semantic anchoring to structural refinement as $t \to 0$. (Middle, Right) The intra-group weights α show the microscopic selection of hierarchical layers within each functional group. Results are obtained from SiT-XL/2 after 400k training iterations on ImageNet.

5.4 Mechanistic Insights into Non-stationary Alignment

In this section, we provide empirical evidence for how AHPA resolves the representational mismatch.

5.4.1 Emergent Routing Policy and Physical Consistency

We analyze the trajectories learned by the Dynamic Router $\mathcal{R}_\phi$ to uncover the model's emergent alignment preferences. Crucially, the router is optimized solely via the generative objective ($\mathcal{L}_{diff} + \lambda\mathcal{L}_{align}$), with no explicit knowledge of the G-SNR diagnostic conducted in Sec. 3.

Autonomous Handover Policy. As visualized in Fig. 5 (Left), the macroscopic weights reveal an autonomous handover policy that mirrors the underlying SNR evolution. The router prioritizes β_{deep} during high-noise phases to anchor the generative manifold, while spontaneously shifting focus toward β_{mid} as denoising progresses. This self-adaptive transition provides a powerful cross-validation: the fact that the backbone "chooses" to reconfigure its guidance source to minimize loss confirms that the non-stationary demand aligns with the inherent dynamics of the diffusion process.

Microscopic Calibration. Within each group, the router executes a fine-grained granularity shifting mechanism. As shown in Fig. 5 (Middle, Right), layer-wise focus consistently transitions from deep bottleneck features to shallower ones as $t \to 0$. This synchronized evolution ensures that the hierarchical priors remain closely matched to the transformer's shrinking receptive field and its increasing need for granular structural cues during final synthesis.

5.4.2 Fidelity Synergy: AHPA Resolves Representational Mismatch

We analyze AHPA's resolution of representational mismatch through the directional consistency of its alignment gradients (G-SNR). As shown in Table 7, static methods suffer from fidelity collapse in late denoising stages, where fixed priors misalign with the model's evolving receptivity and introduce destructive interference. In contrast, AHPA maintains a healthy guidance spectrum throughout the trajectory, achieving the highest averaged G-SNR (11.6). By dynamically routing mid-level structural priors as the backbone's focus shifts, AHPA keeps the alignment signal constructive during critical refinement phases, mitigating the gradient impurity that bottlenecks static strategies. The sustained high-fidelity gradients offer a mechanistic explanation for the improved convergence and structural integrity in our main results, confirming that resolving this non-stationary mismatch is key to unlocking the full potential of intrinsic hierarchical guidance.

Table 7: Analysis of alignment fidelity (G-SNR). Avg. denotes the mean of 100 t-samples.

Method	$t = 0.9$	$t = 0.3$	Avg.
REPA (DINOv2-B)	13.9	4.2	8.5
SRA 2 (Latent)	12.0	3.1	7.2
Uniform $\mathcal{G}_{deep}$	12.3	4.1	7.3
Uniform $\mathcal{G}_{mid}$	4.1	9.2	7.0
AHPA (Ours)	11.9	**10.8**	**11.6**

6 Conclusion

In this study, we introduced AHPA, an adaptive hierarchical alignment framework that resolves the representational mismatch inherent in static guidance paradigms. By identifying the non-stationary alignment requirement along the diffusion trajectory, we demonstrated that the model's needs shift from global semantic anchoring to fine-grained structural refinement. AHPA effectively addresses this by mining hierarchical priors from the intrinsic VAE encoder and adaptively scheduling them via a Dynamic Router. Extensive experiments show that AHPA achieves competitive convergence and structural fidelity, matching or exceeding the performance of heavy-weight external teachers. As a frugal and self-contained solution, AHPA offers a principled path for accelerating large-scale generative models without the burden of external dependencies or redundant computational overhead.

Acknowledgements and Disclosures

Funding. The authors received no third-party funding or support for this work during the 36 months preceding submission.

Competing Interests. The authors declare no competing interests.

References

[1] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models, 2023.

[2] Fan Bao, Shen Nie, Kaiwen Xue, Chongxuan Li, Shi Pu, Yaole Wang, Gang Yue, Yue Cao, Hang Su, and Jun Zhu. One transformer fits all distributions in multi-modal diffusion at scale, 2023.

[3] Hesen Chen, Junyan Wang, Zhiyu Tan, and Hao Li. Sara: Structural and adversarial representation alignment for training-efficient diffusion models, 2025.

[4] Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models, 2022.

[5] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021.

[6] 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, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024.

[7] Wan-Cyuan Fan, Yen-Chun Chen, Dongdong Chen, Yu Cheng, Lu Yuan, and Yu-Chiang Frank Wang. Frido: Feature pyramid diffusion for complex scene image synthesis, 2022.

[8] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis, 2022.

[9] Ali Hatamizadeh, Jiaming Song, Guilin Liu, Jan Kautz, and Arash Vahdat. Diffit: Diffusion vision transformers for image generation, 2024.

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[11] Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation, 2021.

[12] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. Simple diffusion: End-to-end diffusion for high resolution images, 2023.

[13] Dengyang Jiang, Mengmeng Wang, Liuzhuozheng Li, Lei Zhang, Haoyu Wang, Wei Wei, Guang Dai, Yanning Zhang, and Jingdong Wang. No other representation component is needed: Diffusion transformers can provide representation guidance by themselves, 2026.

[14] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion, 2024.

[15] Diederik P. Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation, 2023.

[16] Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models, 2023.

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[18] 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 *ECCV (77)*, volume 15135 of *Lecture Notes in Computer Science*, pages 23–40. Springer, 2024.

[19] Xinyin Ma, Runpeng Yu, Songhua Liu, Gongfan Fang, and Xinchao Wang. Diffusion model is effectively its own teacher. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12901–12911, 2025.

[20] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models, 2021.

[21] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024.

[22] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4172–4182. IEEE, 2023.

[23] Changhao Peng, Yuqi Ye, Shuangjun Du, Wenxu Gao, and Wei Gao. Dual-path condition alignment for diffusion transformers. In *The Fourteenth International Conference on Learning Representations*, 2026.

[24] 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, 2023.

[25] 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, 2021.

[26] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10674–10685. IEEE, 2022.

[27] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[28] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, 2014.

[29] 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? *arXiv preprint arXiv:2512.10794*, 2025.

[30] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*. OpenReview.net, 2021.

[31] Ming Tao, Hao Tang, Fei Wu, Xiao-Yuan Jing, Bing-Kun Bao, and Changsheng Xu. Df-gan: A simple and effective baseline for text-to-image synthesis, 2022.

[32] Mengmeng Wang, Dengyang Jiang, Liuzhuozheng Li, Yucheng Lin, Guojiang Shen, Xiangjie Kong, Yong Liu, Guang Dai, and Jingdong Wang. Sra 2: Variational autoencoder self-representation alignment for efficient diffusion training, 2026.

[33] Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, Jian Yang, et al. Representation entanglement for generation: Training diffusion transformers is much easier than you think. *arXiv preprint arXiv:2507.01467*, 2025.

[34] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks, 2017.

[35] Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models, 2025.

[36] 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 *ICLR*. OpenReview.net, 2025.

[37] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning, 2020.

[38] Han Zhang, Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. Cross-modal contrastive learning for text-to-image generation, 2022.

[39] Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *TMLR*, 2024.

[40] Yufan Zhou, Ruiyi Zhang, Changyou Chen, Chunyuan Li, Chris Tensmeyer, Tong Yu, Jiuxiang Gu, Jinhui Xu, and Tong Sun. Lafite: Towards language-free training for text-to-image generation, 2022.

[41] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis, 2019.

[42] 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, 2024.

A Detailed Experimental Settings

Model Configurations. We evaluate AHPA across three scales: SiT-B/2, L/2, and XL/2. The pre-trained VAE encoder is frozen and utilized as a hierarchical feature extractor. Following the observation that deep convolutional networks exhibit a progressive transition from local textures to global abstractions, We partition the 15 VAE encoder layers into four functional groups ($G_1 \dots G_4$) based on their channel dimensions and spatial resolutions. This partitioning is conducted based on the standard SD-VAE architecture [26], as detailed in Table A.8.

This partitioning reflects the inherent representational hierarchy of the encoder: G_1 and G_2 reside in the early stages, primarily capturing local pixel intensities and geometric primitives. As the network deepens, the representations undergo a gradual abstraction process: G_3 aligns with a transitional stage that preserves spatial topologies and structural relationships, while G_4 resides at the deepest level, encapsulating broader semantic compositions. By designating G_3 and G_4 as $\mathcal{G}_{mid}$ and $\mathcal{G}_{deep}$ respectively, our Dynamic Router can adaptively bridge the gap between these varying levels of abstraction and the diffusion process throughout the denoising trajectory.

Table A.8: **VAE Encoder Layer Partitioning.** Layers are grouped by channel dimensions (C), native spatial resolutions, and their representational focus based on the SD-VAE architecture. Groups G_3 and G_4 are utilized as $\mathcal{G}_{mid}$ and $\mathcal{G}_{deep}$ respectively for adaptive guidance.

Group ID	VAE Encoder Layers	C	Native Res.	Representational Focus
G_1	`conv_in, down_0_block_0, down_0_block_1, down_0_downsample`	128	$256^2 \to 128^2$	Local pixel textures
G_2	`down_1_block_0, down_1_block_1, down_1_downsample`	256	$128^2 \to 64^2$	Geometric primitives
G_3 ($\mathcal{G}_{mid}$)	`down_2_block_0, down_2_block_1, down_2_downsample, down_3_block_0`	512	$64^2 \to 32^2$	Structural topologies
G_4 ($\mathcal{G}_{deep}$)	`down_3_block_1, mid_block_1, mid_attn_1, mid_block_2`	512	32^2	Semantic compositions

Alignment Modules. To align multi-resolution VAE features with the transformer patch grid, all selected features are spatially resampled to 16×16 via bilinear interpolation. The dynamic router $\mathcal{R}_\phi$ is implemented as a 4-layer MLP that takes the timestep t as input to predict hierarchical weights. For feature projection, we employ a single-layer 3×3 convolution as the projection head. This choice preserves spatial inductive biases and smooths potential resampling artifacts while ensuring a lightweight enhancement.

Training Protocol. All models are trained for 400k iterations. We utilize the AdamW optimizer with a constant batch size and learning rate as specified in Table A.9. To isolate the intrinsic contribution of hierarchical priors, we perform alignment training *without* Classifier-Free Guidance (CFG). All experiments are conducted on NVIDIA A100 GPUs.

Evaluation Protocol. To assess generative quality, we employ FID for realism, sFID for spatial coherence, and IS for diversity, alongside Precision and Recall. All metrics are computed using 50,000 samples against the ImageNet training set. During inference, we use the deterministic ODE solver with 250 steps to ensure consistency across all scales and baselines.

B Generalization Across VAE Architectures

To verify whether the learned routing policy ϕ is over-fitted to the specific characteristics of SD-VAE, we conduct a cross-architecture study using SDXL-VAE [24] and Consistency VAE [14]. We adopt a resolution-centric grouping strategy: layers with the lowest spatial resolution (e.g., 32×32 for a 256^2 input) are assigned to $\mathcal{G}_{deep}$, while layers from the preceding stage (e.g., 64×64) form $\mathcal{G}_{mid}$.

Zero-shot Alignment Health. We first evaluate the policy's ability to coordinate "unseen" feature distributions. As shown in Table B.10, we evaluate the AHPA-trained router on a pre-trained vanilla SiT-B/2 by swapping the supervisor VAE encoder without any further fine-tuning. AHPA maintains a robust average G-SNR (exceeding 10.0), significantly outperforming the static uniform baselines.

Table A.9: **Hyperparameter settings across different model scales for AHPA.** We detail the architectural configurations, optimization schedules, and alignment parameters used in our experiments.

Configuration	**SiT-B/2**	**SiT-L/2**	**SiT-XL/2**
Architecture			
# Parameters	132M	460M	677M
Input Latent	$32 \times 32 \times 4$	$32 \times 32 \times 4$	$32 \times 32 \times 4$
Layers	12	24	28
Hidden dim.	768	1,024	1,152
Num. heads	12	16	16
AHPA Settings (Dynamic Router)			
Alignment Weight λ	1.0	1.0	1.0
Alignment Depth	3	6	7
Router MLP Layers	4	4	4
Router Time Embed Dim.	128	128	128
Router Hidden Dim.	256	256	256
Projection Head	3×3 Conv	3×3 Conv	3×3 Conv
Encoder $\mathcal{E}$ (Frozen)	VAE-Enc	VAE-Enc	VAE-Enc
Optimization			
Batch size	256	256	256
Optimizer	AdamW	AdamW	AdamW
Learning rate	2×10^{-4}	2×10^{-4}	2×10^{-4}
(β_1, β_2)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)
Weight decay	0.0	0.0	0.0
Max Grad Norm	2.0	2.0	2.0
EMA decay	0.9999	0.9999	0.9999
CFG Dropout Prob. (Training)	0.1	0.1	0.1
Interpolants & Sampling			
α_t	$1 - t$	$1 - t$	$1 - t$
σ_t	t	t	t
Training objective	v-prediction	v-prediction	v-prediction
Sampler	Euler	Euler	Euler
Sampling steps (NFE)	250	250	250

This high zero-shot alignment health confirms that AHPA captures a universal representational gradient inherent in convolutional hierarchies, proving that our dynamic scheduling logic is an intrinsic property of the diffusion trajectory rather than an artifact of a specific encoder.

Generative Performance Gain. To further assess whether this alignment health translates into superior image quality, we train SiT-B/2 from scratch for 400k iterations with each VAE type. As summarized in Table B.11, AHPA consistently yields significant improvements in both FID and sFID across all architectures. Notably, even with the high-performance Consistency VAE, AHPA achieves a notable FID reduction compared to its vanilla counterpart. These results demonstrate that AHPA effectively mitigates representational bottlenecks regardless of the underlying latent manifold, showcasing its robustness as a general-purpose guidance framework.

Table B.10: **Zero-shot Alignment Diagnostics.** All metrics are Trajectory Averages (G-SNR) computed over 100 t-samples on a frozen vanilla SiT-B/2 backbone. AHPA maintains high alignment health despite supervisor swapping.

Backbone (Frozen)	Supervisor VAE Encoder	Avg. G-SNR ↑
SiT-B/2 (Vanilla)	—	—
AHPA (Ours)	SD-VAE [27] (Default)	**11.6**
	SDXL-VAE [24]	10.4
	Consistency VAE [14]	10.1
Uniform $\mathcal{G}_{deep}$	SD-VAE	7.3
Uniform $\mathcal{G}_{mid}$	SD-VAE	7.0

Table B.11: **Performance Comparison across VAE Architectures.** All models are SiT-B/2 trained from scratch for 400k iterations on ImageNet 256 × 256. AHPA consistently enhances generative quality across different latent manifolds.

Supervisor VAE	Method	FID ↓	sFID ↓
SD-VAE (Default)	Vanilla SiT	33.0	6.46
	AHPA (Ours)	**25.5**	**5.78**
SDXL-VAE	Vanilla SiT	28.4	6.12
	AHPA (Ours)	**22.1**	**5.45**
Consistency VAE	Vanilla SiT	24.5	5.90
	AHPA (Ours)	**21.7**	**5.48**

C Generalization Across Feature Sources and Diffusion Formulations

C.1 DINOv2 Feature-Source Generalization

We further test whether timestep-adaptive hierarchical alignment extends beyond VAE representations. In this diagnostic variant, we replace the VAE alignment features with intermediate features from DINOv2-B [21]. The DINOv2 blocks are partitioned into four depth-wise groups with channel-compatible features within each group, and the latter two groups are routed using the same inter-group and intra-group mechanisms as the default AHPA configuration. This variant is used only to evaluate prior-source generality; the default AHPA model continues to use intrinsic VAE features.

Table C.12 shows that AHPA-DINO improves both FID and trajectory-average G-SNR over REPA at all three model scales. This result extends adaptive hierarchical alignment to a representation hierarchy distinct from the native VAE.

Table C.12: **Generalization to DINOv2 priors across model scales.** All models are trained for 400K iterations on ImageNet 256 × 256. The DINOv2 experiment is a diagnostic variant for evaluating prior-source generality.

Scale	Method	Prior Source	FID ↓	Avg. G-SNR ↑
SiT-B/2	REPA	DINOv2-B	27.2	7.7
	AHPA-DINO	DINOv2-B	**21.6**	**12.4**
SiT-L/2	REPA	DINOv2-B	12.6	7.9
	AHPA-DINO	DINOv2-B	**9.1**	**13.3**
SiT-XL/2	REPA	DINOv2-B	8.5	8.5
	AHPA-DINO	DINOv2-B	**6.8**	**12.9**

C.2 DDPM-based DiT Generalization

To evaluate whether AHPA depends on the flow-matching formulation used by SiT, we apply it to the original DDPM-based DiT-L/2 [22]. We retain the ImageNet 256 × 256 dataset, 400K training iterations, and the AHPA alignment configuration wherever applicable, while following the native DDPM training and sampling formulation of DiT. All methods in this comparison use identical DiT training and sampling settings. As shown in Table C.13, AHPA achieves 14.4 FID, outperforming both REPA and SRA 2. This result demonstrates that AHPA transfers from flow-matching-based SiT to the original diffusion formulation.

Table C.13: **Generalization to DDPM-based DiT.** All methods use DiT-L/2, are trained for 400K iterations on ImageNet 256 × 256, and are evaluated using identical DDPM sampling settings.

Method	FID ↓
Vanilla DiT-L/2	23.3
+ REPA	15.6
+ SRA 2	17.1
+ AHPA (Ours)	**14.4**

Table D.14: **Quantitative comparison on text-to-image generation (MS-COCO)**. Following the established experimental protocol, we use classifier-free guidance with $w = 2.0$ for all evaluations. AHPA achieves competitive performance against methods relying on external representation learners, despite using only intrinsic VAE features.

Method	Type	FID ↓
AttnGAN [34]	GAN	35.49
DM-GAN [41]	GAN	32.64
VQ-Diffusion [8]	Discrete Diffusion	19.75
DF-GAN [31]	GAN	19.32
XMC-GAN [38]	GAN	9.33
Frido [7]	Diffusion	8.97
LAFITE [40]	GAN	8.12
U-Net [2]	Diffusion	7.32
U-ViT-S/2 [1]	Diffusion	5.95
U-ViT-S/2 (Deep) [1]	Diffusion	5.48
MMDiT [6] (ODE; NFE=50)	Diffusion	6.05
MMDiT+REPA [36] (ODE; NFE=50)	Diffusion	5.37
MMDiT+SRA 2 [32] (ODE; NFE=50)	Diffusion	5.65
MMDiT+AHPA (ODE; NFE=50)	Diffusion	**5.02**
MMDiT [6] (SDE; NFE=250)	Diffusion	5.30
MMDiT+REPA [36] (SDE; NFE=250)	Diffusion	4.74
MMDiT+SRA 2 [32] (SDE; NFE=250)	Diffusion	4.91
MMDiT+AHPA (SDE; NFE=250)	Diffusion	**4.52**

D Text-to-Image Generation Experiment

To verify the generalization of AHPA beyond class-conditional synthesis, we conduct text-to-image (T2I) generation experiments on the MS-COCO dataset [17]. This task requires the model to capture complex cross-modal alignments between free-form text and visual structures. Following the established protocol [36, 1], we adopt MMDiT [6] as the diffusion backbone, which jointly processes image patches and text embeddings.

Experimental Setup. We train the MMDiT model from scratch on the MS-COCO training split for 150K iterations with a total batch size of 256. The model configuration consists of a hidden dimension of 768 and a depth of 24 layers. Text prompts are derived using a pre-trained CLIP text encoder [25]. During inference, we evaluate performance on the validation split with a classifier-free guidance scale of 2.0, reporting FID under both ODE (NFE=50) and SDE (NFE=250) samplers.

Performance Analysis. Quantitative results are summarized in Table D.14. AHPA achieves a superior FID of 4.52 (SDE) and 5.02 (ODE), outperforming the previous self-alignment SOTA, SRA 2, by a significant margin. Remarkably, despite relying solely on intrinsic VAE features, AHPA matches or even slightly surpasses REPA, which utilizes heavy-weight external vision learners (DINOv2). These results indicate that the hierarchical priors recovered by AHPA, when adaptively scheduled, provide sufficient semantic-spatial guidance for complex text-conditioned tasks. The consistency of these gains across different samplers further underscores the robustness of our dynamic routing mechanism in resolving representational mismatch.

E Additional Ablation Studies and Technical Details

E.1 Comprehensive Component Ablation

To identify the optimal configuration for AHPA, we conduct an extensive ablation study on the projection head architecture, the alignment loss function, and the supervision weight λ.

Analysis of Results. As summarized in Table E.15, our findings indicate several key design principles:

- **Loss Function: Cosine Similarity** outperforms L1-based losses, suggesting that maintaining directional consistency in the diffusion manifold is more vital than regressing absolute magnitudes.
- **Weighting Strategy:** $\lambda = 1.0$ provides the optimal balance; higher weights (e.g., $\lambda = 2.0$) slightly enhance structural metrics but risk distorting the generative manifold and compromising image realism.
- **Convolutional Advantage:** Single-layer 3×3 Conv significantly outperforms MLP and Linear heads. By leveraging spatial inductive bias, the convolutional head effectively captures local structural dependencies, smoothing resampling artifacts for superior alignment quality.
- **Minimalism vs. Depth:** Stacking deeper blocks (e.g., Conv + GN + SiLU) leads to performance degradation. This indicates that over-processing hierarchical priors via redundant non-linearities distorts their intrinsic distribution, whereas a single-layer conv preserves the highest guidance fidelity.

Table E.15: **Component-wise ablation study of AHPA.** All experiments are conducted on SiT-B/2 for 400K iterations. We evaluate the impact of projection head architecture, loss functions, and alignment weight λ. The configuration (l) represents the default setting used in AHPA.

Variant	Proj. Head	Loss	λ	FID ↓	sFID ↓
(a) Vanilla SiT	—	—	—	33.0	6.46
Loss & Weight					
(b)	1-layer 3×3 Conv	L1	1.0	27.2	6.12
(c)	1-layer 3×3 Conv	Smooth-L1	1.0	26.9	5.95
(d)	1-layer 3×3 Conv	Cosine	0.5	27.8	6.30
(e)	1-layer 3×3 Conv	Cosine	2.0	26.4	5.82
Architecture Depth					
(f)	1-layer Linear	Cosine	1.0	28.2	6.45
(g)	3-layer MLP	Cosine	1.0	26.8	6.10
(h)	5-layer MLP	Cosine	1.0	27.5	6.35
(i)	2-layer Conv Block	Cosine	1.0	26.2	6.01
(j)	3-layer Conv Block	Cosine	1.0	27.4	6.22
(k)	5-layer Conv Block	Cosine	1.0	28.1	6.41
(l)	**1-layer 3×3 Conv**	**Cosine**	**1.0**	**25.5**	**5.78**

E.2 Sensitivity to Hierarchical Partitioning Boundary

To address the potential ambiguity in partitioning VAE layers with identical channel dimensions ($C = 512$), we evaluate the model's sensitivity to the boundary between $\mathcal{G}_{mid}$ and $\mathcal{G}_{deep}$. Specifically, while keeping the total alignment depth constant at 8 layers, we vary the number of layers assigned to each group.

As shown in Table E.16, the balanced configuration ($4 + 4$) achieves the optimal FID, though the performance remains relatively stable across different ratios. This stability suggests that AHPA's effectiveness stems from the functional gradient inherent in the VAE hierarchy rather than a hyper-tuned split point. The slight degradation in the $6 + 2$ setting indicates that insufficient semantic supervision may weaken global composition, while the $2 + 6$ setting slightly hampers structural precision. Consequently, we adopt the balanced split as a robust and parameter-free default.

E.3 Extended Analysis of the Non-stationary Alignment Requirement

The efficacy of AHPA stems from its ability to synchronize guidance granularity with the model's evolving representational needs. To isolate the contribution of the routing mechanism, we contrast our Dynamic Router against two baseline strategies in Table E.17:

Representational vs. Scheduling Gains. The transition from Vanilla SiT-B to Static Uniform alignment provides a 3.3 FID improvement (33.0 → 29.7), establishing hierarchical VAE features as

Table E.16: **Sensitivity analysis of the partitioning boundary.** We vary the layer distribution between $\mathcal{G}_{mid}$ and $\mathcal{G}_{deep}$ while maintaining a total budget of 8 layers. Evaluated on SiT-B/2 at 400K iterations.

$\mathcal{G}_{mid}$ Layers	$\mathcal{G}_{deep}$ Layers	FID ↓	sFID ↓	IS ↑
2	6	26.3	5.95	50.8
4 (Default)	**4 (Default)**	**25.5**	**5.78**	**52.3**
6	2	27.1	6.12	50.5

a robust foundation. However, a fixed linear Heuristic schedule yields only a marginal 1.1 FID gain (29.7 → 28.6). This limited improvement suggests that a simplistic linear transition is insufficient to capture the complex, non-stationary dynamics of the diffusion trajectory.

Autonomy vs. Manual Heuristics. In contrast, our learned Dynamic Router achieves a substantial 4.2 FID gain over the static baseline (29.7 → 25.5), outperforming the linear heuristic by a significant 3.1 FID margin. This gap confirms that the granularity-noise tradeoff requires precise, non-linear coordination that emerges naturally through our adaptive mechanism.

Crucially, while one might attempt to design a hand-tuned non-linear schedule (e.g., a Sigmoid function) based on the diagnostic crossovers in Sec 5.4, such closed-form solutions are inherently fragile. Our observations indicate that the optimal crossover point is architecture-dependent; for instance, the $t \approx 0.4$ crossover identified for SiT-XL does not necessarily generalize to SiT-B. The primary advantage of our router is its autonomy—it eliminates the need for expensive per-model diagnostic probing and manual parameter tuning, providing a generalized solution that automatically discovers the optimal scheduling curvature for any given architecture.

Table E.17: **Necessity of Learned Dynamic Routing.** Evaluated on SiT-B/2. We contrast our learned router against fixed weighting and a manually designed linear scheduling baseline (Heuristic). Our adaptive mechanism consistently maximizes guidance efficiency.

Feature Source	Strategy	IS ↑	FID ↓	Prec. ↑
Vanilla SiT-B	–	43.7	33.0	0.53
$G_3 \cup G_4$	Static Uniform	51.6	29.7	0.58
	Heuristic (Linear)	50.2	28.6	0.59
	Dynamic Routing (Ours)	**52.3**	**25.5**	**0.63**

E.4 Positional Optimality and Proportional Scaling

A critical design decision in AHPA is the choice of the alignment block l. In our preliminary study using static guidance, we identified block $l = 3$ as the optimal entry point for SiT-B. To ensure this optimality persists under the dynamic regime, we evaluate AHPA at various alternative blocks.

As shown in Table E.18, while the Dynamic Router improves performance across all tested blocks compared to static baselines, the peak consistently remains at $l = 3$. This reinforces the observation that the early-to-mid stages of the Transformer backbone are uniquely receptive to hierarchical priors. At these depths, the model has begun to aggregate global context but has not yet committed to the fine-grained pixel details that characterize the final blocks.

Consequently, we adopt a proportional scaling strategy across model sizes. By placing the alignment at approximately the 25th percentile of the total depth (block 3 for Base, 6 for Large, and 7 for XL), we ensure that AHPA consistently targets the most "flexible" representational phase of the backbone, facilitating efficient knowledge transfer from the VAE's hierarchical blueprint.

E.5 Hierarchical Conflict and Optimization Stability

Further analysis of the "Full Union (G1–G4)" experiment in Table 2 provides insight into why localized guidance (G3 ∪ G4) is superior. Shallow VAE features (G1, G2) are tightly coupled with the input pixel space. When forced as alignment targets, they compel the Transformer's early

Table E.18: **Verification of positional optimality under AHPA**. We evaluate our dynamic mechanism at different blocks to confirm if the "golden entry point" identified in the static setting ($l = 3$) remains optimal even with adaptive scheduling.

Alignment Block l	**3**	4	8
Strategy	**Dynamic**	Dynamic	Dynamic
FID ↓	**25.5**	27.2	31.5
IS ↑	**52.3**	50.8	46.4

layers to reconstruct low-level details prematurely. This creates an optimization bottleneck, as the model's capacity is diverted from learning the generative manifold toward simple pixel replication. By excluding these shallow layers, AHPA focuses the alignment signal on the mid-to-deep features that capture the semantic and structural feature of the image, leading to faster convergence and higher fidelity.

E.6 Diversified exploration of Dynamic Router Design

To provide a thorough understanding of the Dynamic Router $\mathcal{R}_\phi$, we ablate its design across four critical dimensions. These experiments justify why a minimalist, timestep-conditioned MLP is the most effective architecture for coordinating hierarchical priors. All models are SiT-B/2 scales trained for 400K iterations.

1. Timestep Encoding Strategy. We evaluate how the representation of scalar $t \in [0, 1]$ affects the quality of manifold alignment. Formally, we compare three encoding schemes:

- **Linear Projection (Ours):** Defined as $e_{linear}(t) = \mathbf{W}t + \mathbf{b}$, where $\mathbf{W} \in \mathbb{R}^{D\times 1}$ and $\mathbf{b} \in \mathbb{R}^D$ are learnable. The constant Jacobian $\frac{\partial e}{\partial t} = \mathbf{W}$ ensures that the router's input changes at a uniform rate, preserving the monotonic nature of the diffusion trajectory and facilitating a seamless transition between different stages.
- **Sinusoidal Embedding:** Following [10], $e_{sin}(t) = [\dots, \sin(\omega_i t), \cos(\omega_i t), \dots]^\top$ with $\omega_i = 10000^{-2i/D}$. While standard for feature conditioning, its high-frequency harmonic components introduce periodic local fluctuations in the weights $\alpha(t)$, which destabilizes the alignment objective (27.0 FID).
- **Fourier Features (RFF):** Maps t via $e_{RFF}(t) = [\dots, \cos(2\pi\mathbf{b}_i t), \sin(2\pi\mathbf{b}_i t), \dots]^\top$ where $\mathbf{b}_i \sim \mathcal{N}(0, \sigma^2)$. This leads to "weight aliasing," where the router over-fits to micro-variations in noise levels, yielding a degraded FID of 26.8.

2. Informational Source: Global Sync vs. Local Adaptation. We evaluate whether the routing policy ϕ should depend on the global denoising schedule or adapt to localized feature representations. We contrast our timestep-only design with a content-adaptive variant.

Implementation of Content Feature: To capture the specific state of the model at each alignment step, the Content-adaptive router extracts the intermediate feature map $h^{(l)} \in \mathbb{R}^{C\times H\times W}$ directly from the target Transformer layer being aligned. We apply global average pooling to collapse the spatial dimensions, resulting in a compact representational vector $\bar{h} \in \mathbb{R}^C$. This vector is then projected via a linear layer and fed into the router MLP. For a fair comparison, the Content-adaptive variant also receives the timestep embedding $e(t)$ to ensure it has access to the full temporal context.

Failure Analysis: As shown in Table E.19, the Content-adaptive approach significantly degrades performance (29.6 vs. 25.5 FID). We attribute this to the stochastic noise dominance in early denoising stages (high t). Attempting to derive routing policies from these noise-heavy features introduces high-frequency "jitter" into the hierarchical weights $\alpha(t)$, preventing the backbone from establishing a consistent representational gradient. This confirms that the demand for VAE priors is a trajectory-wide property governed by global SNR evolution, rather than a local instance-specific one.

3. Router Architecture and Complexity: From Discreteness to Continuity. We contrast our MLP-based $\mathcal{R}_\phi$ with two alternative structural paradigms to verify the necessity of continuous non-linear mapping for inter-group (β) and intra-group (α) scheduling.

- **Lookup Table (LUT):** *Implementation:* We discretize the continuous timestep $t \in [0, 1]$ into $N = 10$ uniform bins. The router is parameterized by two learnable matrices, $\mathbf{M}_\beta \in \mathbb{R}^{N \times G}$ and $\mathbf{M}_\alpha \in \mathbb{R}^{N \times (G \times L)}$. For any input t, the index $idx = \lfloor t \cdot N \rfloor$ is used to retrieve the corresponding weighting vectors:

$$\begin{bmatrix} \beta(t) \\ \alpha(t) \end{bmatrix} = \text{Lookup}(\mathbf{M}, idx)$$

Failure Analysis: This formulation creates a piecewise constant function. As t transitions across a bin boundary (e.g., from 0.899 to 0.901), the routing weights undergo an instantaneous step change. Such discontinuities act as high-frequency "gradient shocks" to the backbone, forcing the transformer to abruptly adapt to new alignment scales. This disrupts the smooth evolution of the latent manifold, resulting in structural artifacts and a degraded FID of 27.2.

- **Attention-based Router:** *Implementation:* We implement a dynamic retrieval mechanism using a Query-Key-Value (QKV) framework. We define a set of $K = 16$ learnable *hierarchical prototypes* $\{\mathbf{k}_i, \mathbf{v}_i\}_{i=1}^K$, where each $\mathbf{v}_i$ represents a potential (α, β) configuration. Given the timestep embedding $e(t)$, the weights are generated via:

$$w_i = \frac{\exp(e(t) \cdot \mathbf{k}_i / \sqrt{D})}{\sum_{j=1}^K \exp(e(t) \cdot \mathbf{k}_j / \sqrt{D})}, \quad \begin{bmatrix} \beta(t) \\ \alpha(t) \end{bmatrix} = \sum_{i=1}^K w_i \cdot \mathbf{v}_i$$

Failure Analysis: While highly expressive, the Attention-based router is over-engineered for the alignment task. The redundant complexity of the attention bank introduces unnecessary optimization noise and potential over-fitting to specific timesteps, yielding a sub-optimal FID of **26.9**.

Conclusion: Our 4-layer MLP provides the ideal inductive bias for this task. Unlike LUT, it ensures $\mathcal{C}^\infty$ continuity across the entire trajectory; unlike Attention, it focuses on global functional mapping rather than discrete pattern retrieval. This leads to the most stable alignment signal and the best generative performance (25.5 FID).

4. Depth of the MLP. We sweep the number of layers $L \in \{3, 4, 8\}$ for our MLP router. A 3-layer MLP lacks the non-linearity required for the sharp semantic-to-structural transition (26.1 FID). We find that $L = 4$ provides the optimal balance; further increasing the depth to $L = 8$ results in inferior performance (26.3 FID), suggesting that excessive complexity may hinder the optimization of the temporal routing policy.

Table E.19: **Ablation Study on Router Configurations.** We systematically evaluate the impact of encoding strategies, input sources, and architectural depths on SiT-B/2 for 400K iterations. Each row represents a specific experimental configuration. **Bold** indicates our final optimized setting.

Exp.	t Encoding	Input Source	Architecture	MLP Depth	FID ↓
1	Sinusoidal	Timestep-only	MLP	4	27.0
2	Fourier (RFF)	Timestep-only	MLP	4	26.8
3	Linear	Content-adaptive	MLP	4	29.6
4	Linear	Timestep-only	Lookup Table	—	27.2
5	Linear	Timestep-only	Attention-based	—	26.9
6	Linear	Timestep-only	MLP	3	26.1
7	**Linear**	**Timestep-only**	**MLP**	**4**	**25.5**
8	Linear	Timestep-only	MLP	8	26.3

F Theoretical Grounding and Validity of G-SNR

To provide a rigorous justification for the Gradient Signal-to-Noise Ratio (G-SNR), we situate this metric within the dual framework of multi-task optimization and information propagation. We demonstrate that G-SNR is a principled measure of *gradient stationarity* and *representational receptivity*.

F.1 G-SNR: Quantifying Gradient Purity and Task Conflict

The AHPA objective, $\mathcal{L}_{total} = \mathcal{L}_{diff} + \lambda \mathcal{L}_{align}$, constitutes a multi-task optimization problem. According to the theory of *Gradient Surgery* [37], the efficacy of joint optimization is governed by the directional alignment of task-specific gradients. If $\cos(g_{diff}, g_{align}) < 0$, the objectives exhibit gradient conflict, where alignment updates counteract the manifold convergence of the denoising task.

Statistical Estimation via Per-sample Gradients. To provide a robust diagnostic independent of optimization hyperparameters, we estimate G-SNR using the statistical properties of the gradient vector field within each mini-batch. Let $\mathcal{B} = \{x_1, \dots, x_B\}$ be a mini-batch of size B, and $g_t^{(i)} = \nabla_\theta \mathcal{L}_{align}(x_i, t)$ be the alignment gradient elicited by the i-th individual sample. We define the empirical G-SNR estimator as:

$$\widehat{\text{G-SNR}}(t) = \frac{\left\| \frac{1}{B} \sum_{i=1}^{B} g_t^{(i)} \right\|_2^2}{\frac{1}{B-1} \sum_{i=1}^{B} \left\| g_t^{(i)} - \bar{g}_t \right\|_2^2} \tag{6}$$

where $\bar{g}_t = \frac{1}{B} \sum_{i=1}^{B} g_t^{(i)}$ is the mini-batch average gradient.

Physical Interpretation and Robustness. Physically, G-SNR measures signal purity: a high G-SNR signifies that the hierarchical prior provides a *stationary* signal that resides within the constructive optimization subspace of the denoising manifold. Conversely, a low G-SNR indicates that g_{align} is dominated by stochastic variance, effectively "poisoning" the trajectory with impurity noise.

Notably, this formulation renders the metric dimensionless and invariant to absolute gradient magnitudes. Since both the numerator (squared expectation) and the denominator (trace of the covariance matrix) scale quadratically with the gradient magnitude, G-SNR reflects directional consistency rather than the absolute energy of the update. This scale-invariance is critical for assessing alignment health across the non-stationary diffusion trajectory where gradient scales vary significantly.

F.2 G-SNR as a Proxy for Representational Receptivity

Beyond stability, G-SNR implicitly measures the backbone's *receptivity* to specific priors. As established in *Signal Propagation* theory [28], a model's ability to extract information from a supervisor is limited by its instantaneous receptive field and abstraction level. A diminishing G-SNR (e.g., for Mid-level priors at $t > 0.8$) signifies a state of representational "blindness," where the backbone lacks the geometric maturity to resolve the prior, causing the gradient to dissipate into unconstructive variance. By maximizing G-SNR, AHPA ensures that guidance is injected only when the model has the capacity to internalize it.

F.3 Alignment with Diffusion SNR Physics

Crucially, G-SNR is coupled with the non-stationary physics of diffusion. *Variational Diffusion Models* [16] show that the focus transitions from global anchoring to local refinement as input SNR increases. G-SNR acts as the representation-space counterpart to this schedule. The crossover behaviors identified in our diagnostic provide mechanistic evidence that the demand for prior "purity" is synchronized with the noise schedule, justifying our use of a Dynamic Router.

F.4 Empirical Validity and Interpretability

We validate G-SNR as a diagnostic tool by monitoring its evolution across methods. As shown in Table F.20, there is a high degree of consistency between G-SNR and generative performance. Methods that suffer from low trajectory-average G-SNR (e.g., Uniform $\mathcal{G}_{deep}$ and SRA 2) consistently exhibit inferior structural fidelity. Notably, AHPA achieves the highest Avg. G-SNR, which mechanistically aligns with its superior sFID and FID scores. This suggests that G-SNR is a *reasonable and robust*

diagnostic metric for assessing alignment health, offering insights into signal purity that standard loss curves fail to capture.

Table F.20: **Comparison of diagnostic health and generative performance.** Evaluated on SiT-XL/2 at 400K iterations (ImageNet-256). Trajectory Avg. G-SNR serves as a mechanism-consistent indicator of the alignment process's health, aligning with the improvements observed in structural fidelity (sFID).

Method	FID ↓	sFID ↓	**Avg. G-SNR ↑**
Uniform $\mathcal{G}_{deep}$	11.2	5.36	7.3
Uniform $\mathcal{G}_{mid}$	14.1	6.25	7.0
SRA 2 (Latent)	13.8	5.10	7.2
REPA (DINOv2-B)	8.5	5.14	8.5
AHPA (Ours)	**8.1**	**4.90**	**11.6**

G 512×512 ImageNet Generation

To evaluate the scalability and high-resolution synthesis capabilities of our method, we conduct experiments on the ImageNet 512×512 dataset using the SiT-XL/2 backbone.

As shown in Table G.21, AHPA maintains its efficiency lead on 512×512 resolution, where it achieves an FID of 2.28 in only 100 epochs—already outperforming the vanilla SiT-XL/2 baseline trained for the full 600 epochs. Its substantial lead in sFID (4.16) confirms that AHPA preserves semantic structures more effectively than the vanilla SiT and static alignment strategies.

Table G.21: Performance comparison on ImageNet 512×512 with CFG.

Model	Epochs	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
Pixel diffusion						
VDM++ [15]	-	2.65	-	278.1	-	-
ADM-G, ADM-U [5]	400	2.85	5.86	221.7	0.84	0.53
Simple diffusion (U-Net) [2]	800	4.28	-	171.0	-	-
Simple diffusion (U-ViT, L) [1]	800	4.53	-	205.3	-	-
Latent diffusion, Transformer						
MaskDiT [39]	800	2.50	5.10	256.3	0.83	0.56
DiT-XL/2 [22]	600	3.04	5.02	240.8	0.84	0.54
SiT-XL/2 [18]	600	2.62	4.18	252.2	**0.84**	**0.57**
+ SRA 2 [32]	100	2.45	4.28	249.1	0.83	0.55
+ REPA [36]	100	2.37	4.20	253.7	0.83	0.56
+ AHPA (ours)	100	**2.28**	**4.16**	**261.4**	**0.84**	0.56

H Detailed Comparison with SOTA Methods

Detailed comparisons with SOTA methods are provided in Table H.22.

I Qualitative Comparison across Different Methods

To qualitatively evaluate the acceleration efficiency, we visualize the generated samples at various training milestones (100K, 200K, and 400K) in Figure I.6. The results highlight the distinct advantages of AHPA in terms of both pixel-level fidelity (FID) and structural coherence (sFID). While static alignment baselines like SRA 2 and REPA often struggle with granularity mismatch—resulting in blurred textures or fragmented geometries in mid-training stages—AHPA successfully maintains directional consistency across the generative manifold. Specifically, AHPA produces sharper object boundaries and more faithful semantic patterns (e.g., animal fur and complex structures) much earlier

Table H.22: **State-of-the-Art Comparison on ImageNet** 256×256 **with CFG.** Performance metrics are annotated with ↑ (higher is better) and ↓ (lower is better). **w/o Ext.** indicates whether the method is standalone and does *not* rely on external dependencies (e.g., extra encoders, diffusion teachers, or diffusion decoders). Bold indicates the best results among methods without external dependencies.

Method	Epochs	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑	w/o Ext.
Pixel diffusion							
ADM-U [5]	400	3.94	6.14	186.7	0.82	0.52	✓
VDM++ [15]	560	2.40	-	225.3	-	-	✓
Simple diffusion [12]	800	2.77	-	211.8	-	-	✓
CDM [11]	2160	4.88	-	158.7	-	-	✓
Latent diffusion, U-Net							
LDM-4 [26]	200	3.60	-	247.7	0.87	0.48	✓
Latent diffusion, Transformer + U-Net hybrid							
U-ViT-H/2 [1]	240	2.29	5.68	263.9	0.82	0.57	✓
DiffiT [9]	-	1.73	-	276.5	0.80	0.62	✓
Latent diffusion, Transformer							
DiT-XL/2 [22]	1400	2.27	4.60	278.2	**0.83**	0.57	✓
+ iSSD [19]	1400+0.8	2.02	4.22	250.0	0.81	0.60	✗
MaskDiT [39]	1600	2.28	5.67	276.6	0.80	0.61	✗
SD-DiT [42]	480	3.23	-	-	-	-	✗
SiT-XL/2 [18]	1400	2.06	4.50	270.3	0.82	0.59	✓
+ REPA [36]	800	1.42	4.70	311.4	0.80	0.63	✗
+ REG [33]	800	1.36	4.25	299.4	0.77	0.66	✗
+ SRA [13]	800	1.58	4.65	305.7	0.80	0.63	✓
+ SRA 2 [32]	800	1.52	4.63	316.2	0.82	0.62	✓
+ AHPA (Ours)	100	2.15	4.38	265.4	0.81	0.58	✓
+ AHPA (Ours)	200	1.78	4.26	291.8	0.82	0.61	✓
+ AHPA (Ours)	400	1.56	4.15	308.2	0.81	0.63	✓
+ AHPA (Ours)	800	**1.40**	**4.12**	**317.5**	0.81	**0.64**	✓

than the baselines. This visual evidence confirms that our adaptive hierarchical prior effectively guides the model to transition from global anchoring to local refinement, leading to superior generative performance under limited computational budgets.

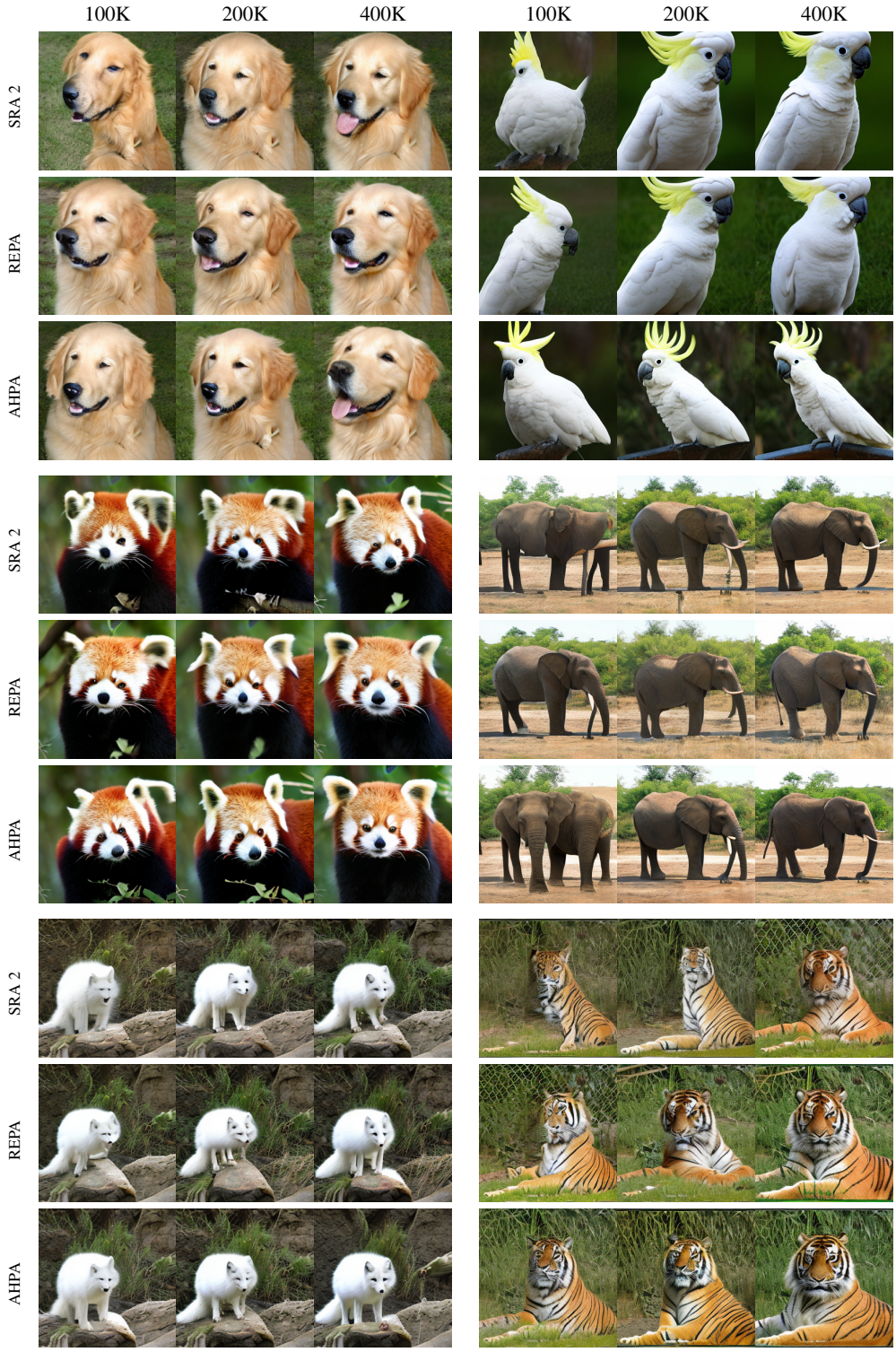


Figure I.6: **Qualitative comparison of generative quality on ImageNet** 256×256**.** We visualize the evolution of samples across three methods—SRA 2, REPA, and our AHPA—all utilizing the SiT-XL backbone. To ensure a rigorous controlled comparison, all samples are generated using a fixed classifier-free guidance scale (CFG=4.0) with identical initial noise latents, random seeds, and class labels.

J Detailed Training Cost Comparison

Here we analyze the training computational overhead of AHPA. To ensure a fair and efficient comparison, we pre-compute and store the VAE hierarchical features offline, thereby eliminating real-time encoding latency during training.

Offline storage requirement. We store all intermediate feature maps from G_3 and G_4 in float16 using sharded Arrow files. For the full ImageNet-1K training set at 256×256 resolution, the serialized cache occupies approximately 3.8 TB (about 3 MB per image on disk). Constructing this cache requires a single frozen-VAE preprocessing pass and serialization, taking approximately 1.5 hours on four NVIDIA A100 GPUs. This is a one-time cost and fits within the local scratch capacity of a standard multi-node GPU cluster.

Training-time I/O overhead. During training, the Arrow shards are memory-mapped and the required features are loaded per batch. The materialized G_3 and G_4 feature tensors amount to approximately 10 MB per image before serialization and storage-layout effects. In practice, this access does not become the training bottleneck: AHPA achieves 7.4 Iter/s compared with 8.1 Iter/s for vanilla SiT-XL/2, corresponding to a 9% reduction. This modest throughput change is consistent with the 5% GFLOPs increase from the alignment head and routing modules; a substantial I/O bottleneck would produce a considerably larger throughput degradation.

As summarized in Table J.23, AHPA demonstrates superior computational efficiency compared to state-of-the-art alignment-based methods. While methods like SRA [13] or REPA [36] incur significant overhead (up to +73% GFLOPs and +71% latency) by relying on heavy external teacher models (e.g., DINOv2 or CLIP), AHPA is external-free. By utilizing a streamlined single-layer 3×3 convolution for alignment and a lightweight 5-layer MLP for dynamic routing, AHPA adds only a negligible **5%** to the total GFLOPs.

Even when accounting for the channel projection layers and the routing mechanism, AHPA maintains a minimal parametric footprint of only 2.5M additional parameters. Our method achieves a high training throughput of 7.4 Iter/s, which is nearly on par with the vanilla SiT-XL/2 baseline (8.1 Iter/s) and significantly faster than external-guided counterparts. These results underscore that AHPA provides a highly frugal yet effective path for feature alignment.

Table J.23: **Detailed Training Computational Cost Comparison on ImageNet** 256×256. We evaluate AHPA against various alignment baselines using SiT-XL/2 as the transformer backbone. #EFP is decomposed into *External Model Params* + *Alignment Head Params*. All benchmarks are conducted on NVIDIA A100 GPUs with a fixed batch size of 256.

Method	#EFP (M)	TS (Iter/s) ↑	GFLOPs ↓	Latency (ms) ↓
SiT-XL/2 (Baseline)	$0 + 0$	8.1	114.46	18.05
+ REPA [36]	$86 + 8$	6.3 (-22%)	138.50 (+21%)	22.81 (+26%)
+ SRA [13]	$481 + 2$	5.1 (-37%)	197.57 (+73%)	30.90 (+71%)
+ SRA 2 [32]	$0 + 18$	7.2 (-11%)	118.55 (+4%)	19.20 (+6%)
+ AHPA (Ours)	**0 + 2.5**	**7.4** (-9%)	**119.85** (+5%)	**19.80** (+10%)

K Limitations

While AHPA demonstrates significant improvements in training efficiency and generative quality, we acknowledge several limitations that provide avenues for future work:

- **Dependence on Pre-trained VAEs:** Our method relies on the hierarchical feature space of a frozen VAE (e.g., SD-VAE). While these features are robust for general image synthesis, the effectiveness of AHPA might be constrained if the pre-trained encoder fails to capture domain-specific nuances in highly specialized datasets (e.g., medical imaging or high-resolution remote sensing).

- **Heuristic Layer Partitioning as a Principled Baseline:** While our group-wise PCA and diagnostic analysis (Sec. 3) establish a robust empirical foundation for partitioning VAE layers into functional groups (G1–G4), these boundaries are currently predefined to maintain architectural simplicity and transparency. By fixing these groups, we demonstrate that AHPA can achieve significant gains using minimalist inductive biases without the need for complex, end-to-end grouping protocols. Nevertheless, developing a fully automated, learnable partitioning mechanism represents an exciting future direction that could further enhance AHPA's adaptability to heterogeneous encoder architectures.
- **Focus on Class-Conditional Synthesis:** This study primarily focuses on class-conditional ImageNet synthesis and MS-COCO text-to-image tasks. The generalization of adaptive hierarchical alignment to other modalities, such as video generation or 3D synthesis, requires further investigation to account for temporal and volumetric prior requirements.
- **Fixed Number of Groups:** In our experiments, we utilize four functional groups. While this granularity is sufficient for existing DiT/SiT scales, extremely large models (e.g., $>$10B parameters) might benefit from an even finer-grained hierarchical schedule which we have not yet explored.

L Broader Impacts

Our work on Adaptive Hierarchical Prior Alignment (AHPA) has several societal implications, ranging from environmental benefits to ethical considerations in generative AI.

Positive Societal Impacts. The most direct positive impact is the significant reduction in the environmental footprint of training large-scale generative models. By achieving a 17$\times$ acceleration in convergence, AHPA substantially lowers the total energy consumption and associated carbon emissions required to reach state-of-the-art performance. Furthermore, our "frugal" framework—which avoids reliance on massive external teacher models—promotes the democratization of AI research. It enables researchers with limited computational budgets to develop high-quality generative systems without needing the prohibitive hardware resources typically required for traditional alignment methods.

Negative Societal Impacts and Mitigations. As AHPA enhances the visual fidelity and structural consistency of synthetic images, it inherently increases the risk of misuse for creating deepfakes or disinformation. High-quality synthetic content can be weaponized to deceive the public or impersonate individuals. To mitigate these risks, we advocate for the integration of digital watermarking (e.g., StegaStamp) and the development of robust forgery detection algorithms alongside the release of such models. Additionally, since AHPA aligns with VAE features trained on large-scale datasets (e.g., ImageNet), it may inadvertently inherit or amplify societal biases present in the training data. We encourage users of our framework to perform bias audits and apply post-generation filtering when deploying these models in sensitive applications.

Ethical Statement. This research does not involve human subjects, personal data, or harmful biological/chemical agents. We have conducted our experiments in accordance with established ethical guidelines for machine learning research.

M Computational Resources and Compute Estimates

In accordance with the NeurIPS 2026 guidelines, we provide a detailed disclosure of the computational resources and total compute investment utilized for this research.

M.1 Hardware and Environment

All experiments, including preliminary explorations and final evaluations, were performed on an internal high-performance computing cluster. The specific hardware configuration per node is as follows:

- **Compute Workers:** 8 $\times$ NVIDIA A100 GPUs (80GB SXM5).

- **CPU:** Dual AMD EPYC 7742 64-Core Processors.
- **Memory:** 1 TB DDR4 system RAM.
- **Storage:** 4 TB NVMe local scratch space.
- **Software:** CUDA 12.1, PyTorch 2.2, and NCCL 2.18.

M.2 Individual Run Compute Requirements

We report the estimated wall-clock time and GPU hours for single successful training runs (400K iterations) on ImageNet 256×256. The specific costs are:

- **SiT-B/2 + AHPA:** $\sim$44 hours on 8 GPUs ($\approx$ 352 GPU hours).
- **SiT-L/2 + AHPA:** $\sim$78 hours on 8 GPUs ($\approx$ 624 GPU hours).
- **SiT-XL/2 + AHPA:** $\sim$112 hours on 8 GPUs ($\approx$ 896 GPU hours).
- **Vanilla Baselines (Re-trained):** Totaling $\sim$1,800 GPU hours to ensure fair comparison under identical hardware.
- **MS-COCO Text-to-Image:** $\sim$65 hours on 16 GPUs ($\approx$ 1,040 GPU hours).

M.3 Total Project Compute Disclosure

The full research project required significantly more compute than the final experiments reported in the main paper to account for the iterative development process:

- **Preliminary Explorations:** Initial diagnostic probing of VAE hierarchical groups and verification of the non-stationary requirement ($\sim$500 GPU hours).
- **Hyperparameter Tuning:** Searching for the optimal Dynamic Router architecture and alignment weight λ ($\sim$1,200 GPU hours).
- **Failed Experiments:** Early attempts at content-adaptive routing and alignment at late-stage blocks that did not yield performance gains ($\sim$800 GPU hours).

Total Estimated Compute: We estimate the total computational investment for the entire research cycle to be approximately 7,200 NVIDIA A100 GPU hours.

N More Related Works

Generative Models and Diffusion Transformers. Image generation has witnessed a transition from pixel-space denoising models, such as DDPM [10] and DDIM [30], to Latent Diffusion Models (LDMs) [26] that operate in compressed latent spaces. Architecturally, while early frameworks relied on U-Net backbones [10, 26], modern paradigms have shifted towards Transformer-based architectures like DiT [22] and SiT [18], which leverage self-attention mechanisms for superior spatial modeling [22]. Despite these advances, diffusion transformers typically require extensive training iterations to achieve convergence [29, 33]. While acceleration can be achieved through structural modifications like masked training [39], these methods often necessitate changes to the training paradigm.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly state our primary contributions: identifying non-stationary alignment requirements, proposing the AHPA framework, and leveraging hierarchical VAE priors. These claims are directly supported by the diagnostic analysis in Section 3 and the extensive experiments in Section 5.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitations of AHPA in Appendix K, specifically noting its dependence on the representational richness of the pre-trained VAE and potential performance variability in domain-specific tasks outside of general image synthesis.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: This paper is primarily an empirical study. We do not present new formal mathematical theorems or proofs. Instead, we provide conceptual motivations and rationales for our proposed diagnostic metrics (G-SNR), grounded in established optimization and signal propagation theories, as detailed in Section 3 and further discussed in Appendix F.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: We provide detailed descriptions of the VAE layer partitioning (Table A.8), Dynamic Router architecture and all training hyperparameters (Appendix A) necessary for replication.

Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The implementation, including router code and training scripts, is publicly available at https://github.com/UIOSN/AHPA. All datasets used (ImageNet and MS-COCO) are publicly available under their respective access terms.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Comprehensive details regarding data preprocessing, optimizer settings, and model-specific hyperparameters are provided in Appendix A.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: Consistent with large-scale generative model training practices (e.g., DiT-XL/SiT-XL), single full-training runs are reported due to extreme computational costs.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Detailed hardware configuration and total compute estimates are provided in Appendix M.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Our research focuses on improving the optimization efficiency of generative models and conforms to all ethical standards.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: Broader impacts, including the reduced environmental footprint of efficient training and potential risks of high-fidelity synthesis (e.g., deepfakes), are discussed in Appendix L.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [N/A]

Justification: This paper proposes a training methodology rather than a new generative model or dataset for public release.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: We properly cite and credit all utilized assets. These include ImageNet (custom research license), MS-COCO (CC-BY 4.0), and the foundational SiT/DiT codebases (Apache-2.0). All terms of use and licensing requirements for these assets have been strictly respected throughout our research.

Guidelines:

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

13. **New assets**

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

    Answer: [N/A]

    Justification: No new datasets or large-scale pre-trained models are introduced.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: Our research did not involve human subjects or crowdsourcing.

    Guidelines:

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

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

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

    Answer: [N/A]

Justification: No human subject research was conducted.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLMs were only used for standard editing and formatting assistance.

Guidelines:

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