PERSISTENCE FORCING: EXPLOITING FEATURE SPECIALIZATION IN PIXEL-SPACE DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Pixel-space diffusion Transformers (DiTs) directly operate on high-dimensional visual data, yet their hidden representations typically undergo uniform refinement across depth. Natural images, however, are inherently organized at different levels of granularity. Global structure can often be represented compactly, whereas local textures and fine details require richer representations. Motivated by this, we introduce heterogeneous refinement in pixel-space DiTs, assigning different feature groups distinct refinement budgets across depth. Consequently, an ordered feature specialization emerges: sparsely refined features predominantly encode global visual structure, whereas more frequently refined features increasingly specialize toward localized, high-frequency details. We refer to these two groups as persistent and active features, respectively. Building on this emergent specialization, we introduce *Persistence Forcing (PerF)*, which explicitly exploits this persistent–active feature organization for pixel-space image generation. This enables persistent features to continuously condition actively refined features, allowing stable global information to guide the ongoing refinement of finer visual details. During generative sampling, this interaction further induces a meaningful guidance direction that promotes coherent global structure and naturally complements classifier-free guidance. On ImageNet 256×256, PerF-L achieves FID of 1.91, approaching 1.86 of JiT-H with only half the parameters, while PerF-H further achieves FID of 1.63 and 1.76 on ImageNet 256×256 and 512×512, respectively.

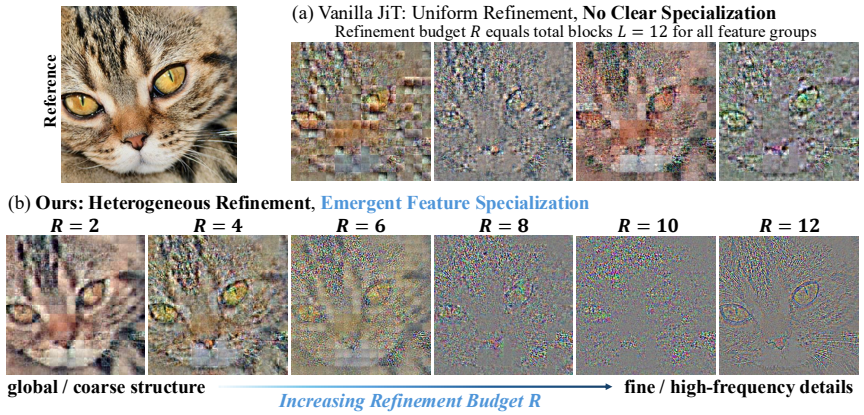

Figure 1: **Heterogeneous refinement induces ordered feature specialization.** We visualize the contributions of different feature groups to the final RGB prediction. Under uniform refinement, feature groups show no systematic ordering. Under heterogeneous refinement, increasing refinement budget R organizes their contributions from global structure toward fine, high-frequency details.

1 INTRODUCTION

Learning effective internal representations is a key bottleneck in training generative Diffusion Transformers (DiTs) (Peebles & Xie, 2023; Yu et al., 2025; Wang et al., 2026b). In particular, recent pixel-space diffusion models (Li & He, 2026; Chen et al., 2026b; Yu et al., 2026) directly operate on high-dimensional visual data without relying on a pretrained latent representation (Rombach et al.,

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

2022; Zheng et al., 2026) and therefore rely heavily on the model itself to organize useful representations throughout denoising. Despite rapid progress in architectures (Wang et al., 2026b), training objectives (Ma et al., 2026a), and sampling strategies (Zhou et al., 2026), DiTs largely retain a simple computational pattern: the full hidden representation is repeatedly transformed throughout network depth. As a result, hidden features typically undergo a uniform refinement schedule. Natural images, however, are inherently organized at different levels of granularity. Global structure can often be represented compactly, whereas local textures and fine details require richer representations (Burt & Adelson, 1983; Mallat, 1989; Schwartz & Simoncelli, 2001). This distinction is particularly relevant to pixel-space diffusion, where a single model must jointly organize global semantic structure and resolve fine-grained, high-frequency visual details directly in the raw pixel space (Chen et al., 2026b; Ma et al., 2026a; Yu et al., 2026). This motivates us to reconsider whether all hidden features should receive the same amount of refinement in pixel-space DiTs.

We therefore introduce heterogeneous refinement, assigning different feature groups distinct refinement budgets across network depth. Under this design, an ordered pattern of feature specialization emerges. Sparsely refined features predominantly encode coherent global visual structure, whereas more frequently refined features increasingly specialize toward localized, high-frequency details. We refer to these two groups as *persistent* and *active* features, respectively. Notably, persistent features already exhibit coherent global organization after limited refinement and remain stable while active features continue to evolve toward progressively finer details. Heterogeneous refinement therefore gives rise to a natural persistent–active organization of the hidden representation, as illustrated in Figure 1. This emergent feature specialization further opens the possibility of using the persistent features as stable context for features that remain actively refined.

To this end, we propose *Persistence Forcing (PerF)*, which explicitly exploits the persistent–active feature specialization induced by heterogeneous refinement. During denoising, persistent features continuously condition actively refined representations, allowing stable global information to guide the ongoing refinement of progressively finer visual details. This persistent-to-active interaction is related to the broader coarse-to-fine principle explored in recent diffusion architectures, where global representations condition high-frequency or pixel-level refinement (Wang et al., 2026b; Ma et al., 2026a; Yu et al., 2026). Unlike these explicitly designed hierarchies, however, PerF builds on the feature specialization that emerges naturally from heterogeneous refinement. We further leverage this interaction at sampling time through Persistence Guidance (PG). By contrasting predictions with and without persistent conditioning, PG constructs a guidance direction that strengthens the influence of the internally established structural context. This guidance complements the externally specified semantic signal provided by classifier-free guidance (CFG) (Ho & Salimans, 2022). On ImageNet 256×256 generation, PerF consistently improves the corresponding JiT baselines across multiple model scales, reducing FID from 3.66 to 2.81 for B/16, from 2.36 to 1.91 for L/16, and from 1.86 to 1.63 for H/16. The improvement further extends to ImageNet 512×512 generation, where PerF-H/32 reduces FID from 1.94 to 1.76 and achieves an Inception Score of 335.3.

We highlight the main contribution of this paper below:

- We introduce heterogeneous refinement in pixel-space diffusion Transformers and reveal an ordered form of feature specialization, where different refinement budgets naturally yield persistent and actively refined representations with complementary coarse-to-fine roles.
- We propose *Persistence Forcing*, which leverages persistent representations as structural context for active refinement and amplifies their influence through sampling-time guidance.
- PerF consistently improves JiT across model scales and resolutions, reducing FID from 3.66 to 2.81 for JiT-B on ImageNet 256×256, while PerF-H achieves FIDs of 1.63 and 1.76 at 256×256 and 512×512 resolutions, respectively.

2 Heterogeneous Refinement Induces Feature Specialization

2.1 Heterogeneous Refinement via Variable Width

Existing pixel-space diffusion Transformers (Li & He, 2026; Baade et al., 2026) largely inherit the standard DiT design (Peebles & Xie, 2023), in which the entire hidden representation is propagated by every Transformer block. Consequently, all hidden feature dimensions un-

dergo the same number of refinement steps throughout network depth. We question whether such uniform refinement is necessary for pixel-space generation, where a single end-to-end network must organize representations ranging from global visual structure to fine-grained image details. To investigate this question, we introduce heterogeneous refinement, where different feature groups receive distinct refinement budgets across network depth. We instantiate this study on JiT (Li & He, 2026), a simple uniform-width pixel-space DiT that provides a clean setting for isolating the effect of heterogeneous refinement. As illustrated in Figure 2, we maintain a D-dimensional hidden representation and allow the ℓ-th Transformer block to actively transform only the first $d_\ell \leq D$ feature dimensions. The remaining dimensions bypass the block unchanged and can re-enter computation when the width expands in later layers. We quantify how frequently each feature, indexed by i, is transformed by its refinement budget

$$R_i = \sum_{\ell=1}^{L} \mathbf{1}[i \leq d_\ell], \tag{1}$$

indicating that only first d_ℓ feature channels are active at layer ℓ. Under uniform refinement, as in vanilla JiT (Li & He, 2026), $R_i = L$ for all feature dimensions. By varying d_ℓ across depth, our construction instead creates a range of refinement budgets and thus heterogeneous refinement histories across the hidden representation. This layer-wise width variation has recently been explored in decoder-only language models as a means of nonuniform capacity allocation across network depth (Wu et al., 2026). Here, we use this architectural degree of freedom to induce heterogeneous refinement histories across feature dimensions in pixel-space diffusion.

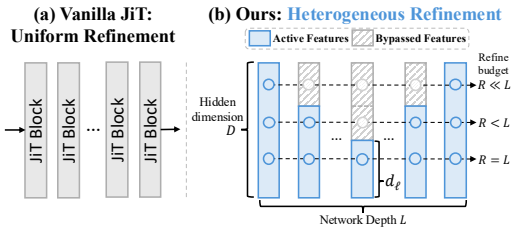

Figure 2: **From uniform to heterogeneous refinement.** (a) In vanilla JiT (Li & He, 2026), every hidden dimension is updated at every block, yielding the same refinement budget $R_i = L$. (b) Our design updates only the first d_ℓ dimensions at layer ℓ while bypassing the remaining. Varying d_ℓ across depth assigns different refinement budgets to different feature dimensions.

In this work, we utilize a contraction–expansion schedule for d_ℓ, gradually narrowing and then expanding the active width across depth to produce a broad range of refinement budgets while allowing bypassed features to re-enter later computation. To isolate the effect of heterogeneous refinement from changes in backbone capacity, we approximately match the attention and MLP parameter budget of the corresponding vanilla JiT by preserving $\sum_{\ell=1}^{L} d_\ell^2 = Ld^2$, where d denotes the hidden dimension of the corresponding JiT baseline.

2.2 Emergent Feature Specialization

Heterogeneous refinement assigns different feature groups distinct refinement budgets, without prescribing their representational roles. Despite this minimal intervention, an ordered feature specialization emerges.

To make this specialization visible, we decompose the final RGB prediction into the additive contributions of different feature groups through the model output projection. Figure 1 visualizes these contributions for groups with different refinement budgets. Under heterogeneous refinement, they exhibit an ordered coarse-to-fine progression. Groups receiving only limited refinement already preserve coherent object appearance and global image structure, while increasing refinement generally shifts their contributions toward more localized and higher-frequency content, including textures, boundaries, and residual details. In contrast, feature partitions under uniform refinement all receive the same refinement budget and show no comparable ordering. Therefore, heterogeneous refinement does more than expose different feature groups to different amounts of computation. Moreover, it organizes their representational roles according to their refinement histories.

This organization reveals a natural feature specialization. We refer to sparsely refined features that preserve predominantly global visual structure as *persistent features*, and to features that remain repeatedly transformed throughout network depth as *actively refined features*. Notably, these roles are not explicitly assigned. The architecture determines only the refinement history of each feature group, while its representational role emerges through denoising.

2.3 Heterogeneous Refinement Alone Is Not Enough

However, the feature specialization induced by heterogeneous refinement above does not directly translate into better generation performance. Despite exhibiting a clear coarse-to-fine feature specialization, a parameter-matched model with heterogeneous refinement does not outperform the corresponding vanilla JiT and can even produce worse FID, as shown in Table 1. This gap suggests that heterogeneous refinement provides an effective mechanism for revealing distinct representational roles, but is not sufficient by itself to improve generation quality.

Model	Params.	100 Epoch		200 Epoch	
		FID ↓	IS ↑	FID ↓	IS ↑
Uni.	131M	**42.1**	**37.6**	**33.1**	**49.8**
Hetero.	131M	42.7	36.2	33.8	47.1

Table 1: **Heterogeneous refinement alone does not readily achieve better generation.**

The remaining opportunity lies in how these roles interact. In the basic heterogeneous-refinement construction, persistent features are carried forward while actively refined features continue through the Transformer blocks, but the preserved information is not explicitly used to support the ongoing computation. This motivates us to move beyond inducing heterogeneous refinement and explicitly exploit the persistent representations it creates.

3 Persistence Forcing

3.1 Persistent-to-Active Conditioning

Section 2.2 reveals a natural feature specialization under heterogeneous refinement. Motivated by their specialization, we explicitly construct the interaction between persistent and actively refined features by using the persistent representation as a structural context for active refinement. Rather than repeatedly transforming information that has already established coherent global organization, we preserve it and allow it to continuously inform the features that remain under active computation. This turns persistence from passive feature preservation into an explicit interaction between the two representational roles.

At the ℓ-th Transformer block, we denote the actively transformed features by $\mathbf{h}_\ell^a$ and the persistent features bypassing the block by $\mathbf{h}_\ell^p$. While $\mathbf{h}_\ell^a$ is updated by the Transformer block, $\mathbf{h}_\ell^p$ is withheld from transformation at this block and instead serves as an additional modulation condition for the active features. Specifically, we augment the standard timestep and class modulation with a token-wise modulation (Wang et al., 2026b) derived from the persistent representation,

$$\mathbf{m}_\ell = \mathrm{AdaLN}_\ell(t, y) + \mathcal{P}_\ell\left(\mathrm{Norm}(\mathbf{h}_\ell^p)\right). \tag{2}$$

The first term denotes the standard timestep and class modulation, which is shared across image tokens, while the second provides token-wise modulation derived from the persistent representation. Specifically, the resulting persistent modulation remains spatially aligned with the active representation, allowing the established structure of the current sample to directly inform its subsequent refinement.

The predominantly coarse and globally organized information carried by persistent features also suggests that this conditioning can be represented compactly. We therefore implement $\mathcal{P}_\ell$ with a low-rank bottleneck of dimension r, providing a lightweight structural condition for active refinement. Persistent information can thus influence every active refinement step without requiring the persistent representation itself to be repeatedly transformed. This interaction follows a coarse-to-fine principle related to recent decoupled diffusion architectures (Wang et al., 2026b; Ma et al., 2026a; Yu et al., 2026), while the two roles here emerge within a single backbone from heterogeneous refinement rather than being explicitly assigned.

Importantly, making the persistent influence explicit also turns it into a controllable internal conditioning variable. In the next section, we exploit this property to construct Persistence Guidance.

3.2 Persistence Guidance

Guidance can be interpreted as extrapolating along a meaningful contrast between two denoising predictions. Classifier-free guidance (CFG) (Ho & Salimans, 2022) forms this contrast by removing

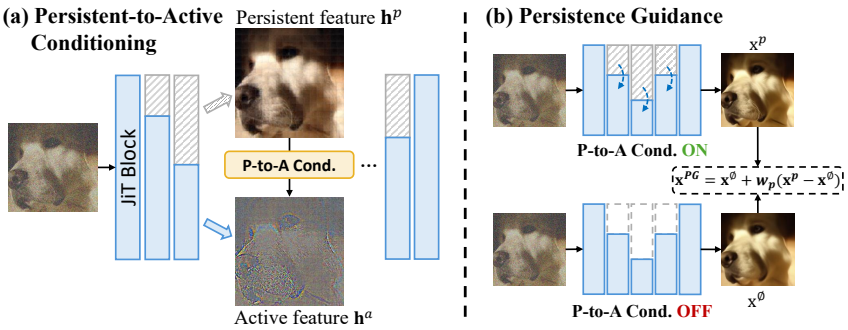

Figure 3: **Overview of Persistence Forcing.** (a) Persistent-to-active conditioning explicitly uses preserved persistent features as structural context for actively refined features. (b) The resulting controllable interaction is further exploited to construct Persistence Guidance.

an external semantic condition, while recent self-guidance approaches construct it from intermediate predictions of the same model (Zhou et al., 2026). These methods highlight that the effectiveness of guidance depends on what information the prediction contrast isolates.

Persistent-to-active conditioning not only provides stable context for active refinement, but also exposes how persistent representations influence the denoising prediction. This conditioning provides a different and directly interpretable contrast which can be further utilized as meaningful guidance during the generative sampling. We therefore stochastically disable the persistent-conditioning pathway during training, enabling the same model to predict both with and without access to this internal context. This strategy give rise to a meaningful sampling guidance called Persistence Guidance (PG). For a noisy input $\mathbf{x}_t$, we denote by $\mathbf{x}_c^p$ the class-conditional prediction with P-to-A conditioning, by $\mathbf{x}_u^p$ its class-unconditional counterpart, and by $\mathbf{x}_c^\varnothing$ the class-conditional prediction with P-to-A conditioning disabled. CFG and PG can then be written symmetrically as

$$\text{CFG} : \mathbf{x}^{\text{CFG}} = \mathbf{x}_u^p + w_c(\mathbf{x}_c^p - \mathbf{x}_u^p), \qquad \text{PG} : \mathbf{x}^{\text{PG}} = \mathbf{x}_c^\varnothing + w_p(\mathbf{x}_c^p - \mathbf{x}_c^\varnothing), \tag{3}$$

where w_c and w_p control the strengths of CFG and PG, respectively. In PG, since two predictions $\mathbf{x}_c^p$ and $\mathbf{x}_c^\varnothing$ differ only in the availability of persistent conditioning, their contrast $(\mathbf{x}_c^p - \mathbf{x}_c^\varnothing)$ captures how the sample-specific structural context redirects the ongoing refinement. Given that persistent features predominantly preserve established global visual structure, PG provides a direction for reinforcing consistency with the structure already formed within the model. Setting $w_p = 1$ recovers the standard persistent-conditioned prediction, whereas $w_p > 1$ strengthens the influence of the persistent structural context on subsequent refinement.

PG and CFG can be naturally combined during to further improve the generation quality as

$$\mathbf{x}^{\text{CFG+PG}} = \mathbf{x}_u^p + w_c(\mathbf{x}_c^p - \mathbf{x}_u^p) + (w_p - 1)(\mathbf{x}_c^p - \mathbf{x}_c^\varnothing). \tag{4}$$

Intuitively, PG strengthens the influence of the global structural context carried by persistent features during denoising. Unlike CFG, which relies on external class conditioning, PG derives its guidance direction from internally formed persistent representation, making it a form of internally evolved structural guidance. Because the two guidance signals arise from different sources of conditioning, they capture distinct yet related aspects of generation and can be naturally complementary. As illustrated in Fig. 6, CFG primarily strengthens class-consistent semantic appearance, whereas PG promotes coherent global organization. Combining them allows semantic refinement to proceed while remaining anchored to the structural context already established within the model.

4 Experiments

4.1 Implementation Details

We conduct class-conditional generation on ImageNet (Deng et al., 2009) 256×256 and 512×512 resolutions, and build PerF upon the JiT architectures (Li & He, 2026). Unless otherwise stated, we follow the JiT optimization and noise-sampling setup. We replace the uniform hidden width with a contraction–expansion schedule, while approximately matching the attention and MLP parameter budget of the corresponding vanilla JiT by preserving $\sum_{\ell=1}^{L} d_\ell^2 = Ld^2$. Persistent-to-active

Model	Params.	Epochs	FID ↓	IS ↑	Prec. ↑	Rec. ↑
Latent-space Diffusion Models						
DiT-XL/2 (Peebles & Xie, 2023)	675M	1400	2.27	278.2	**0.83**	0.57
SiT-XL/2 (Ma et al., 2024)	675M	1400	2.06	277.5	**0.83**	0.59
REPA-XL/2 (Yu et al., 2025)	675M	800	1.42	305.7	0.80	0.64
DDT-XL/2 (Wang et al., 2026b)	675M	400	1.26	**310.6**	0.79	0.65
RAE-XL/2 (Zheng et al., 2026)	839M	800	1.13	262.6	0.78	**0.67**
Internal Guidance (Zhou et al., 2026)	678M	680	**1.07**	274.1	0.79	0.66
Pixel-space Diffusion Models						
ADM-G (Dhariwal & Nichol, 2021)	554M	400	4.59	186.7	**0.82**	0.52
RIN (Jabri et al., 2022)	410M	480	3.42	182.0	–	–
SiD (Hoogeboom et al., 2023)	2B	–	2.44	256.3	–	–
VDM++ (Kingma & Gao, 2023)	2B	–	2.12	267.7	–	–
PixelFlow-XL/4 (Chen et al., 2025)	677M	320	1.98	282.1	<u>0.81</u>	0.60
PixNerd-XL/16 (Wang et al., 2026a)	700M	320	1.93	297.0	0.79	0.59
EPG-G/16 (Lei et al., 2026)	1391M	600	1.75	275.1	0.80	0.62
PixelDiT-XL (Yu et al., 2026)	797M	320	**1.61**	292.7	0.78	<u>0.64</u>
DiP-XL/16 (Chen et al., 2026b)	631M	600	1.79	281.9	0.80	0.62
DeCo-XL/16 (Ma et al., 2026a)	682M	600	1.69	304.0	0.79	0.63
PixelU-H/16 (Guo et al., 2026)	1168M	600	<u>1.63</u>	305.9	0.79	<u>0.64</u>
PixelGen-XL/16 (Ma et al., 2026b)	676M	160	1.83	293.6	0.79	0.63
PixelREPA-H/16 (Shin et al., 2026)	953M	600	1.81	<u>317.2</u>	0.79	0.63
JiT-B/16 (Li & He, 2026)	131M	600	3.66	275.1	<u>0.81</u>	0.52
JiT-L/16 (Li & He, 2026)	459M	600	2.36	298.5	0.79	0.60
JiT-H/16 (Li & He, 2026)	953M	600	1.86	303.4	0.78	0.62
PerF-B/16	137M	600	2.81	288.3	<u>0.81</u>	0.56
PerF-L/16	471M	600	1.91	311.2	0.77	<u>0.64</u>
PerF-H/16	987M	600	<u>1.63</u>	**324.5**	0.77	**0.65**

Table 2: Comparison with latent-space and pixel-space diffusion models on ImageNet 256×256.

Model	Params.	Epochs	FID ↓	IS ↑	Prec. ↑	Rec. ↑
Latent-space Diffusion Models						
SiT-XL/2 (Ma et al., 2024)	675M	600	2.62	252.2	**0.84**	0.57
REPA-XL/2 (Yu et al., 2025)	675M	800	2.08	**274.6**	0.83	0.58
RAE-XL/2 (Zheng et al., 2026)	839M	800	**1.13**	259.6	0.80	**0.63**
Pixel-space Diffusion Models						
ADM-G (Dhariwal & Nichol, 2021)	554M	400	7.72	172.7	**0.84**	0.53
PixNerd-XL/16 (Wang et al., 2026a)	700M	340	2.84	245.6	0.80	0.59
EPG-L/32 (Lei et al., 2026)	540M	800	2.35	295.4	0.82	0.57
DeCo-XL/16 (Ma et al., 2026a)	682M	340	2.22	290.0	0.80	0.60
PixelDiT-XL (Yu et al., 2026)	797M	850	1.81	278.6	0.78	0.67
PixelU-H/32 (Guo et al., 2026)	1152M	600	1.92	322.1	0.80	0.58
JiT-H/32 (Li & He, 2026)	956M	600	1.94	309.1	0.80	0.61
PerF-H/32	992M	600	**1.76**	**335.3**	0.79	**0.64**

Table 3: Comparison with latent-space and pixel-space diffusion models on ImageNet 512×512.

conditioning is implemented with a low-rank bottleneck of rank 128 for PerF-B/L and 192 for PerF-H, keeping the additional parameter overhead below 5% across model scales. Class conditioning and persistent conditioning are independently dropped with probability 0.1. We employ REPA (Yu et al., 2025; Singh et al., 2026) on the persistent features with a weight of 0.1 only for the main experiments. All ablation and representation analyses are conducted without REPA unless otherwise stated. More detailed implementation configurations are provided in the Appendix C.



Figure 4: **Qualitative results** on ImageNet 512×512 and 256×256 using PerF-H.

4.2 Main Results

ImageNet 256×256**.** Table 2 compares PerF with representative latent-space and pixel-space generative models. Across all model scales, PerF consistently improves the corresponding JiT baseline while introducing less than 5% additional parameters. PerF-B reduces FID from 3.66 to 2.81, PerF-L from 2.36 to 1.91, and PerF-H from 1.86 to 1.63, corresponding to relative reductions of approximately 23%, 19%, and 12%, respectively. The gains therefore persist as the backbone scales from Base to Huge, despite the increasingly strong JiT baseline. PerF also consistently improves Inception Score, with PerF-H increasing IS from 303.4 to 324.5. Notably, the improvement is accompanied by higher recall across all three model scales, while precision remains comparable to the corresponding JiT models. This suggests that Persistence Forcing improves generative coverage without simply trading diversity for sample fidelity. Overall, PerF-H achieves an FID of 1.63 while remaining competitive with the strongest recent pixel-space models.

ImageNet 512×512**.** The same behavior extends to higher-resolution generation in Table 3. PerF-H/32 improves JiT-H/32 from FID of 1.94 to 1.76 and from IS of 309.1 to 335.3, while recall increases from 0.61 to 0.64. These gains are obtained without changing the underlying pixel-space generation framework, showing that the benefit of the persistent–active organization transfers across both model scale and image resolution. Together, the results indicate that Persistence Forcing provides a consistent improvement over its JiT counterpart rather than relying on a particular backbone size or resolution.

4.3 Ablation Study

We conduct ablation studies on ImageNet 256×256 using PerF-B. Unless otherwise stated, each model variant is trained for 100 epochs without REPA, while sampling-only guidance sweeps reuse the trained PerF-B model from the main experiments.

Refinement schedule. We study the sensitivity of PerF to the precise contraction–expansion profile by shifting the bottleneck across network depth. As shown in Figure 5, all four schedules achieve similar performance and outperform the uniform JiT baseline at the same training stage, despite substantially different refinement histories. In particular, FID varies only from 40.35 to 41.35 across the tested schedules, compared with 42.1 for JiT. This suggests that the effectiveness of PerF does not depend on a narrowly tuned width profile, but is robust to different realizations of heterogeneous refinement. Among them, schedules with a later bottleneck perform slightly better, with the Late and Very Late variants achieving FIDs of 40.35 and 40.44, respectively. We therefore adopt the Late Bottleneck schedule as our default configuration.

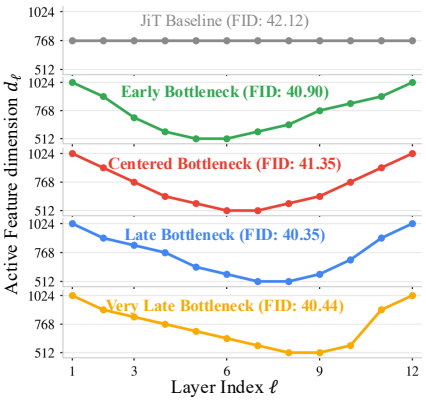

Figure 5: **Ablation on PerF Schedules.**

Component analysis. As established in Table 1, heterogeneous refinement alone does not improve generation quality, showing that inducing distinct refinement histories is not sufficient by itself. Table 4 therefore starts from the heterogeneous backbone and studies how its emergent fea-

Model	Params.	w/o CFG		with CFG	
		FID ↓	IS ↑	FID ↓	IS ↑
PerF (hetero. only)	131M	42.71	36.2	8.02	165.6
+ P-to-A Condition	137M	40.35	38.2	7.19	173.8
+ Persistence Guidance	137M	**24.92**	**62.1**	**5.81**	**195.8**

Table 4: **Progressive construction of PerF.**

P-to-A low-rank r	Params.	PG=1.0		PG=4.0	
		FID ↓	IS ↑	FID ↓	IS ↑
32	1.5M	41.66	36.9	28.89	52.9
64	2.9M	42.24	36.6	26.02	59.1
128 (default)	5.7M	40.35	38.2	24.92	62.1
256	11.5M	40.06	38.4	22.16	67.2

Table 5: **Ablation on P-to-A low-rank r.**

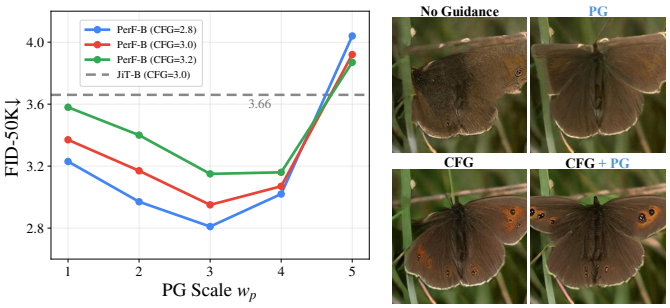

Figure 6: **PG plays a complementary role to CFG.** PG primarily reinforces the global structural organization, whereas CFG strengthens semantic appearance. Combining the two preserves both effects.

PG Interval	w/o CFG		with CFG	
	FID ↓	IS ↑	FID ↓	IS ↑
w/o PG	21.05	75.0	3.37	308.3
[0,1]	**16.03**	**90.3**	**3.12**	**311.0**
$[0.1, 1]$	17.40	85.6	3.16	307.4
$[0, 0.9]$	16.17	89.3	3.23	309.6
$[0.1, 0.9]$	17.60	84.7	3.19	306.2

Table 6: **Analysis of PG interval.** We set CFG=3.0 and PG=2.0 when guidance is enabled.

ture organization is progressively exploited. Introducing P-to-A conditioning improves FID from 42.71 to 40.35 without CFG and from 8.02 to 7.19 with CFG, demonstrating the benefit of explicitly coupling persistent and actively refined features. The resulting controllable conditioning pathway further enables Persistence Guidance, which provides the largest additional improvement, reducing FID to 24.92 and 5.81, respectively. Importantly, PG is not an independent guidance module: its guidance direction is defined by the prediction contrast obtained by enabling and disabling the P-to-A pathway. The three stages therefore play different roles where heterogeneous refinement induces the feature organization, P-to-A conditioning exposes this organization as an explicit interaction, and PG further exploits its effect during sampling.

P-to-A bottleneck rank. Table 5 studies the capacity of the low-rank projector used for persistent-to-active conditioning. To isolate the effects of P-to-A conditioning and Persistence Guidance, we disable CFG in this experiment (CFG $= 1$). Increasing the bottleneck rank generally improves generation quality, especially when PG is applied. Without additional PG (PG $= 1$), increasing r from 128 to 256 yields only a marginal FID improvement from 40.35 to 40.06, while doubling the additional parameters from 5.7M to 11.5M. With stronger PG (PG $= 4$), the benefit of a larger bottleneck becomes more pronounced, suggesting that a higher-capacity conditioning pathway provides a more informative direction for Persistence Guidance. We use $r = 128$ by default as a favorable trade-off between generation quality and parameter overhead.

Interaction between PG and CFG. For this sampling-only analysis, we reuse the PerF-B model from the main experiment and vary only the guidance configuration. Figure 6 studies whether the benefit of Persistence Guidance can be explained by simply changing the strength of CFG. Across all tested CFG scales, a moderate amount of PG consistently improves FID, with the best region appearing around $w_p = 3$. The persistence of this gain after CFG is separately tuned suggests that PG does not merely reproduce stronger semantic guidance. Instead, it provides an additional direction that remains useful over a broad range of CFG strengths. This behavior is consistent with the different sources of the two guidance signals. CFG emphasizes externally specified semantic information, while PG reinforces the internally evolved structural context of the current sample. Performance degrades again when w_p becomes too large, indicating that the persistent context is most effective when it guides active refinement without over-constraining it. Table 6 further studies the denoising interval over which PG is applied. Applying PG throughout the full interval $[0, 1]$ gives the best overall performance both with and without CFG. These results suggest that the persistent guidance

signal remains useful throughout much of the denoising trajectory, rather than being confined to a narrow stage of generation.

5 Related Work

5.1 Learning Representations in Pixel-Space Diffusion Transformers

Recent pixel-space diffusion models have highlighted the importance of how visual representations are formed and organized within the denoising backbone. JiT (Li & He, 2026) shows that a simple Transformer operating on large image patches can directly model pixel-space generation without a pretrained tokenizer. PixelDiT (Yu et al., 2026) further separates patch-level global modeling from pixel-level local refinement, while DeCo (Ma et al., 2026a) organizes generation along the frequency dimension. These works highlight the benefit of assigning different forms of visual information to distinct computational roles. Another line of work directly improves intermediate diffusion representations. REPA (Yu et al., 2025; Singh et al., 2026) aligns latent diffusion features with pretrained visual representations, and subsequent methods further adapt representation alignment to pixel-space diffusion (Shin et al., 2026). Together, these studies show that both the quality and organization of intermediate representations can substantially influence generative learning. These perspectives motivate us to study how distinct representational roles can emerge from heterogeneous computation itself. We find that heterogeneous refinement naturally induces a persistent–active feature specialization, which Persistence Forcing exploits to support ongoing active refinement with preserved persistent representations.

5.2 Guidance for Diffusion Models

Sampling guidance is widely used to improve the quality and controllability of diffusion models. Classifier-free guidance (CFG) (Ho & Salimans, 2022) amplifies the contrast between conditional and unconditional predictions, while AutoGuidance (Karras et al., 2024) generalizes this principle by contrasting models of different quality. A growing line of training-free methods constructs guidance signals by perturbing internal attention or computation paths to obtain alternative predictions for contrastive extrapolation (Hong et al., 2023; Hong, 2024; Ahn et al., 2024; Hyung et al., 2025; Chen et al., 2026a). Recent Internal Guidance (IG) (Zhou et al., 2026) further explores guidance signals arising directly from the internal representations of diffusion Transformers, showing that the internal representational hierarchy can itself provide useful sampling signals. Our work derives its signal from another form of internal organization induced by heterogeneous refinement. Persistent-to-active conditioning exposes a controllable contrast between predictions with and without persistent structural context, which PG amplifies during sampling. The resulting signal reinforces internally evolved structural information and complements the external semantic conditioning provided by CFG.

6 Conclusion

In this work, we revisit the common assumption of uniform refinement in pixel-space diffusion Transformers and show that heterogeneous refinement gives rise to an ordered form of feature specialization. Features receiving limited refinement predominantly preserve coherent global visual structure, while more frequently refined features increasingly focus on finer, high-frequency details, revealing a natural persistent–active organization within the denoising backbone. Building on this observation, we introduce *Persistence Forcing (PerF)*, which explicitly leverages persistent representations as structural context for active refinement and further amplifies their influence through sampling-time Persistence Guidance. Across model scales and image resolutions, PerF consistently improves the corresponding JiT baselines, achieving FIDs of 1.63 and 1.76 with the Huge models on ImageNet 256×256 and 512×512, respectively. These results suggest that organizing computation around emergent feature roles, rather than refining all hidden representations uniformly, provides a promising direction for pixel-space generative modeling and motivates further study of heterogeneous refinement in broader diffusion architectures and applications.

AI USE STATEMENT

In this work, we did not use generative AI tools for tasks requiring mandatory disclosure. We did not use generative AI tools to develop theoretical models, formulate mathematical claims, design experiments, implement methods, or interpret experimental results, and the remaining mandatory-disclosure tasks are not applicable to this work. We used generative AI tools for code review, debugging, and minor code editing, as well as to proofread the manuscript for grammar, clarity, and readability. All AI-assisted work was reviewed by the authors. AI-assisted code modifications were tested for correctness, and suggested manuscript revisions were verified to preserve the technical meaning and accuracy of the paper. The authors take full responsibility for the final content of this work, including all text and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the model architecture, training, sampling, and evaluation protocols throughout the main paper and appendix. The appendix further includes complete implementation details and experimental configurations for reproducing the reported results.

REFERENCES

Donghoon Ahn, Hyoungwon Cho, Jaewon Min, Wooseok Jang, Jungwoo Kim, SeonHwa Kim, Hyun Hee Park, Kyong Hwan Jin, and Seungryong Kim. Self-rectifying diffusion sampling with perturbed-attention guidance. In *European Conference on Computer Vision*, pp. 1–17. Springer, 2024.

Alan Baade, Eric Ryan Chan, Kyle Sargent, Changan Chen, Justin Johnson, Ehsan Adeli, and Li Fei-Fei. Latent forcing: Reordering the diffusion trajectory for pixel-space image generation. *arXiv preprint arXiv:2602.11401*, 2026.

P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983. doi: 10.1109/TCOM.1983.1095851.

Shuning Chang, Pichao WANG, Jiasheng Tang, Fan Wang, and Yi Yang. Sparsedit: Token sparsification for efficient diffusion transformer. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=jTBxyQempF.

Chubin Chen, Jiashu Zhu, Xiaokun Feng, Nisha Huang, Chen Zhu, Meiqi Wu, Fangyuan Mao, Jiahong Wu, Xiangxiang Chu, and Xiu Li. Stochastic self-guidance for training-free enhancement of diffusion models. In *International Conference on Learning Representations*, volume 2026, pp. 134462–134490, 2026a.

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

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

Quan Dao and Dimitris Metaxas. Multi-patch global-to-local transformer architecture for efficient flow matching and diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33000–33011, 2026.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Weichen Fan, Haiwen Diao, Penghao Wu, and Ziwei Liu. Show the signal, hide the noise: Spectral forcing for pixel-space diffusion. *arXiv preprint arXiv:2606.15236*, 2026.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

Zipeng Guo, Lichen Ma, Yu He, Xiaolong Fu, Jingling Fu, Junshi Huang, and Yan Li. Pixelu: A u-shaped transformer for efficient end-to-end pixel diffusion. *arXiv preprint arXiv:2606.27760*, 2026.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Susung Hong. Smoothed energy guidance: Guiding diffusion models with reduced energy curvature of attention. *Advances in Neural Information Processing Systems*, 37:66743–66772, 2024.

Susung Hong, Gyuseong Lee, Wooseok Jang, and Seungryong Kim. Improving sample quality of diffusion models using self-attention guidance. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7428–7437. IEEE, 2023.

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pp. 13213–13232. PMLR, 2023.

Junha Hyung, Kinam Kim, Susung Hong, Min-Jung Kim, and Jaegul Choo. Spatiotemporal skip guidance for enhanced video diffusion sampling. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11006–11015. IEEE, 2025.

Allan Jabri, David Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. *arXiv preprint arXiv:2212.11972*, 2022.

Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. *Advances in Neural Information Processing Systems*, 37:52996–53021, 2024.

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

Jiachen Lei, Keli Liu, Julius Berner, Y HoiM, Hongkai Zheng, Jiahong Wu, and Xiangxiang Chu. There is no VAE: End-to-end pixel-space generative modeling via self-supervised pre-training. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=HbUoKPIZmp`.

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

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

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

Zehong Ma, Ruihan Xu, and Shiliang Zhang. Pixelgen: Pixel diffusion beats latent diffusion with perceptual loss. *arXiv preprint arXiv:2602.02493*, 2026b.

Stephane G Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693, 1989.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Odelia Schwartz and Eero P Simoncelli. Natural signal statistics and sensory gain control. *Nature neuroscience*, 4(8):819–825, 2001.

Jaeyo Shin, Jiwook Kim, and Hyunjung Shim. Representation alignment for just image transformers is not easier than you think. *arXiv preprint arXiv:2603.14366*, 2026.

Jaskirat Singh, Xingjian Leng, Zongze Wu, Liang Zheng, Richard Zhang, Eli Shechtman, and Saining Xie. What matters for representation alignment: Global information or spatial structure? In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=y0UxFtXqXf.

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

Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. Ddt: Decoupled diffusion transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 40633–40642, 2026b.

Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. In *European Conference on Computer Vision*, pp. 420–436. Springer, 2018.

Ziqiao Wang, Wangbo Zhao, Yuhao Zhou, Zekai Li, Zhiyuan Liang, Mingjia Shi, Xuanlei Zhao, Pengfei Zhou, Kaipeng Zhang, Zhangyang Wang, Kai Wang, and Yang You. REPA works until it doesn't: Early-stopped, holistic alignment supercharges diffusion training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=HK96GI5s7G.

Zhaofeng Wu, Oliver Sieberling, Shawn Tan, Rameswar Panda, Yury Polyanskiy, and Yoon Kim. Variable-width transformers. *arXiv preprint arXiv:2606.18246*, 2026.

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

Jiahui Yu and Thomas S Huang. Universally slimmable networks and improved training techniques. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1803–1811, 2019.

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

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

Wangbo Zhao, Yizeng Han, Jiasheng Tang, Kai Wang, Yibing Song, Gao Huang, Fan Wang, and Yang You. Dynamic diffusion transformer. In *International Conference on Learning Representations*, volume 2025, pp. 65520–65552, 2025.

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

Xingyu Zhou, Qifan Li, Xiaobin Hu, Hai Chen, and Shuhang Gu. Guiding a diffusion transformer with the internal dynamics of itself. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11536–11545, 2026.

A Extended Related Work

Pixel-Space Diffusion Models Pixel-space diffusion models directly model images without relying on a pretrained tokenizer (Rombach et al., 2022; Yao et al., 2025; Zheng et al., 2026). Early works demonstrate the feasibility of diffusion directly in pixel space (Dhariwal & Nichol, 2021; Jabri et al., 2022; Hoogeboom et al., 2023; Kingma & Gao, 2023), while recent approaches increasingly adopt Transformer-based architectures. PixelFlow (Chen et al., 2025) and JiT (Li & He, 2026) show that scalable Transformer backbones can effectively operate on raw pixels, with JiT further demonstrating that large image patches provide a simple and strong design for pixel-space generation. Subsequent works improve pixel modeling through different forms of architectural organization. PixNerd (Wang et al., 2026a) models intra-patch details with neural fields, while DiP (Chen et al., 2026b) and PixelDiT (Yu et al., 2026) separate global structure modeling from local detail refinement. DeCo (Ma et al., 2026a) instead decouples generation along the frequency dimension, and PixelU (Guo et al., 2026) introduces a U-shaped Transformer to combine semantic abstraction with fine-grained spatial information. Other works improve pixel-space generation through self-supervised pretraining or perceptual objectives (Lei et al., 2026; Ma et al., 2026b). In contrast to these explicitly designed spatial, frequency, or objective decompositions, our work studies how representational roles can emerge from heterogeneous computation itself. By assigning different feature groups heterogeneous refinement budgets, PerF induces a persistent–active organization within a single Transformer backbone and subsequently exploits this emergent specialization for generation.

Nonuniform Computation in Transformers Nonuniform computation has been explored extensively as a way to allocate model capacity according to computational demand. Early approaches vary network depth or width through adaptive execution, layer skipping, or slimmable architectures (Graves, 2016; Wang et al., 2018; Yu & Huang, 2019). In Transformers, related methods distribute computation nonuniformly across layers or tokens through structured layer dropping, token pruning, and dynamic routing (Rao et al., 2021; Raposo et al., 2024). More recently, Variable-Width Transformers (Wu et al., 2026) explicitly vary hidden width across network depth, demonstrating that nonuniform layer-wise capacity allocation can alter computational efficiency in decoder-only large language models. Nonuniform computation has also been introduced into diffusion Transformers. DyDiT (Zhao et al., 2025) dynamically adapts model width across diffusion timesteps and selectively processes spatial tokens, while subsequent methods vary token density or computation across layers, spatial regions, and denoising stages (Chang et al., 2025). Rather than using nonuniform computation primarily for efficiency or capacity allocation, PerF uses heterogeneous refinement as a mechanism for inducing and exploiting structured specialization within the hidden representation.

Hierarchical Organization in Pixel-Space Diffusion Hierarchical organization has emerged as a recurring principle in generative modeling, where global or low-frequency structure is established before finer visual details are resolved. Recent pixel-space models explicitly instantiate this principle through architectural decomposition. DiP (Chen et al., 2026b) separates global structure construction from local detail restoration, while PixelDiT (Yu et al., 2026) employs patch-level and pixel-level Transformers for global semantic modeling and local texture refinement, respectively. Related hierarchical designs progressively vary spatial resolution or patch granularity across network depth (Dao & Metaxas, 2026). Other methods impose coarse-to-fine organization along the frequency or denoising trajectory. DeCo (Ma et al., 2026a) separates low-frequency semantic modeling from high-frequency detail generation, while Latent Forcing (Baade et al., 2026) encourages semantic latent representations to mature before high-frequency pixel information. Spectral Forcing (Fan et al., 2026) instead makes the time-dependent spectral structure of pixel-space diffusion

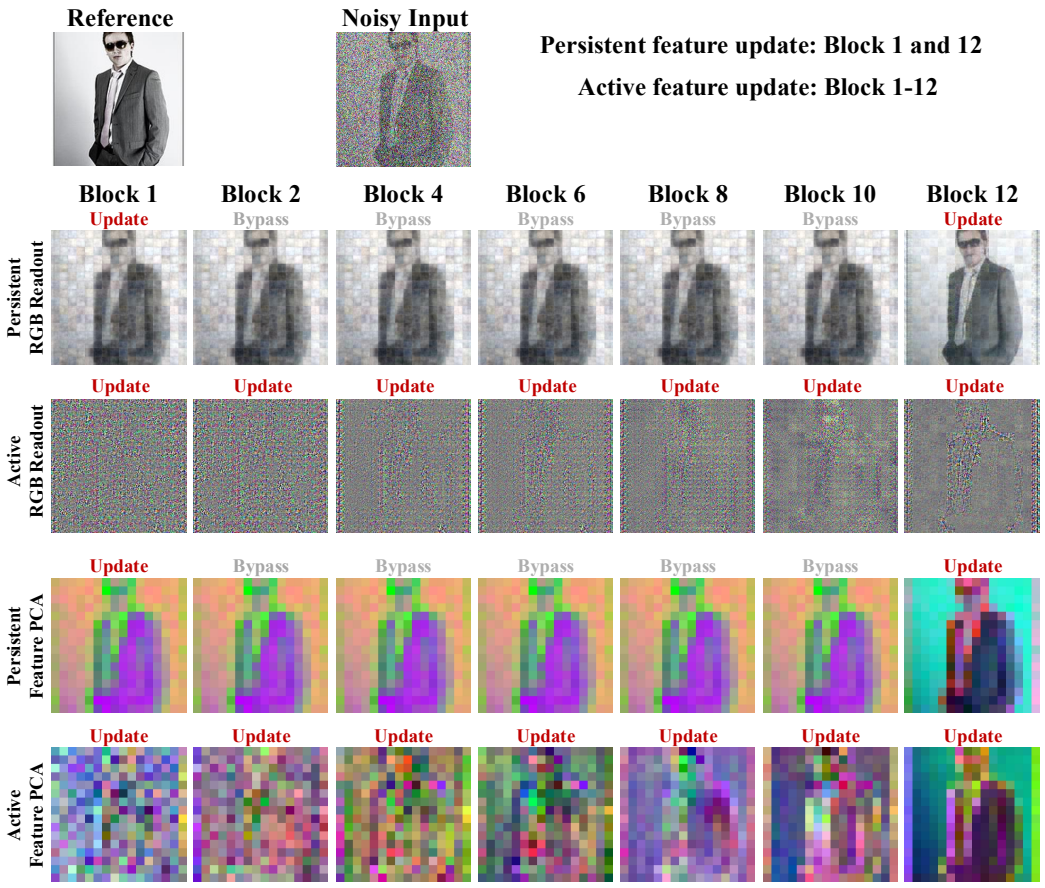

Figure 7: **Depth-wise evolution of persistent and active features.** We visualize the intermediate RGB readouts and feature-space PCA representations of a persistent feature group and the corresponding actively refined features. The persistent features are updated at Blocks 1 and 12 and bypass the intermediate blocks, whereas the active features are updated at every block. After only the first update, persistent features already exhibit coherent global object structure, which remains stable throughout the bypass trajectory before being further refined upon re-entry at Block 12. In contrast, active features continue to evolve across depth and predominantly exhibit residual and higher-frequency structures at intermediate blocks.

explicit by exposing signal-bearing frequency components while suppressing noise-dominated frequencies. These approaches explicitly prescribe a spatial, frequency, or temporal ordering for generation. In contrast, PerF does not assign coarse and fine roles a priori: heterogeneous refinement causes such an ordering to emerge across feature groups with different refinement histories.

B Analysis on Feature Specialization

B.1 Depth-wise Evolution of Persistent and Active Features

Our previous analyses characterize the different roles of persistent and active features through their contributions to the final prediction. We further examine how these representations evolve *within* the network, particularly whether persistent features already contain structured information before their final re-entry. Figure 7 visualizes the depth-wise trajectories of a representative persistent feature group and the corresponding actively refined features. In this example, the persistent group is updated only at Blocks 1 and 12, while bypassing the intermediate blocks, whereas the active features are updated throughout the network. For a controlled comparison, we select 128-dimensional persistent and active feature groups, ensuring that both visualizations use the same number of channels.

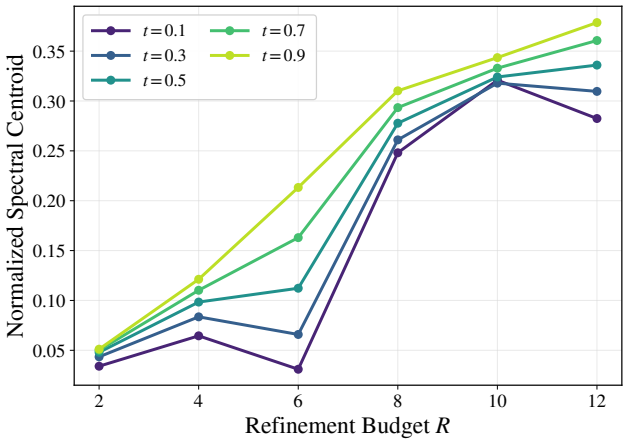

Figure 8: **Frequency specialization across refinement budgets.** We measure the normalized spectral centroid of the RGB contribution from feature groups with different refinement budgets R, using equal-sized 64-channel subsets for a controlled comparison. Larger refinement budgets generally correspond to higher spectral centroids across diffusion timesteps, quantitatively confirming the coarse-to-fine feature organization observed in Figure 1.

Remarkably, after only the first update, the persistent features already exhibit a coherent representation of the global object structure. Both their RGB readout and feature-space PCA visualization remain highly stable throughout the subsequent bypass trajectory from Blocks 2 to 10. This observation is important because the structural organization is already present *before* the persistent features re-enter computation at the final block. It therefore cannot be explained solely as a property acquired after late feature mixing. After re-entry at Block 12, the persistent representation is further refined and incorporates more detailed visual information.

The actively refined features exhibit a markedly different trajectory. Their intermediate RGB readouts remain dominated by residual and high-frequency patterns, while their feature-space organization continuously changes across depth. Only toward later blocks do increasingly recognizable local structures and boundaries emerge. Together, these trajectories provide direct evidence that heterogeneous refinement induces distinct depth-wise representational behaviors: persistent features acquire globally organized information after limited refinement and preserve it during bypass, whereas active features continue to evolve as computation proceeds. This behavior further motivates the use of persistent features as a stable conditioning signal for ongoing active refinement in Section 3.1.

B.2 Quantifying Feature Specialization in the Frequency Domain

The qualitative decomposition in Figure 1 suggests that feature groups with different refinement budgets develop systematically different spatial characteristics. We further quantify this specialization in the frequency domain by measuring the spectral content of each group's contribution to the final RGB prediction.

Frequency metric. For each feature group, we first obtain its additive RGB contribution through the model's output projection. To avoid confounding refinement budget with the number of contributing channels, we evaluate each group using an equal-sized subset of 64 feature dimensions. We subtract the spatial mean and compute the 2D Fourier power spectrum of the RGB contribution. We then measure its normalized spectral centroid, defined as the energy-weighted mean radial spatial frequency. Higher values therefore indicate that a larger fraction of the contribution is concentrated toward finer spatial frequencies.

Frequency organization across refinement budgets. Figure 8 reports the normalized spectral centroid across refinement budgets at several diffusion timesteps. A clear coarse-to-fine organization emerges. Feature groups with small refinement budgets ($R = 2$–6) are consistently dominated by low spatial frequencies, whereas groups receiving more refinement exhibit substantially higher

Model	Depth	$d_{\max}$	$d_{\min}$	Width schedule $\{d_\ell\}$
PerF-B/16	12	1024	512	[1024, 896, 832, 768, 640, 576, 512, 512, 576, 704, 896, 1024]
PerF-L/16	24	1600	512	[1600, 1472, 1408, 1280, 1152, 1088, 1088, 960, 896, 832, 832, 768, 704, 640, 640, 576, 576, 512, 640, 768, 896, 1088, 1280, 1600]
PerF-H/16	32	2000	640	[2000, 1920, 1840, 1760, 1600, 1520, 1440, 1360, 1360, 1280, 1200, 1120, 1040, 1040, 1040, 960, 880, 800, 800, 800, 720, 720, 720, 640, 800, 880, 1040, 1120, 1280, 1520, 1680, 2000]

Table 7: **Layer-wise width schedules used in PerF.** For each model scale, the variable-width backbone is approximately parameter-matched to the corresponding JiT backbone by preserving the attention and MLP parameter budget, $\sum_{\ell=1}^{L} d_\ell^2 \approx Ld^2$.

	PerF-B/16	PerF-L/16	PerF-H/16	PerF-H/32
Architecture				
Image size	256	256	256	512
Depth	12	24	32	32
Patch size	16	16	16	32
Hidden dim $d_{\max}$	1024	1600	2000	2000
Hidden dim $d_{\min}$	512	512	640	640
P-to-A rank	128	128	192	192
Parameters	137M	471M	987M	992M
Training				
Epochs	600	600	600	600
Optimizer	Adam, $\beta_1 = 0.9,\ \beta_2 = 0.95$			
Batch size	1024			
Learning rate	2e-4			
Learning rate schedule	constant			
Weight decay	0			
EMA decay	$\{0.9996, 0.9999\}$			
Time sampler	$\mathrm{logit}(t) \sim \mathcal{N}(-0.8, 0.8^2)$			
Noise scale	1.0	1.0	1.0	2.0
Class drop prob.	0.1			
P-to-A cond. drop prob.	0.1			
Sampling				
ODE solver	Heun			
ODE steps	50			
CFG scale	2.8	2.7	3.0	2.9
PG scale	3.0	2.8	1.9	2.0
CFG interval	$[0.1, 1.0]$			
PG interval	$[0, 1]$			

Table 8: **Configurations of PerF experiments.**

spectral centroids. The largest transition occurs between intermediate and highly refined groups, with $R \geq 8$ contributing increasingly finer spatial structure. Importantly, the same overall trend is observed across all evaluated timesteps, indicating that the relationship between refinement budget and frequency specialization is not restricted to a particular point along the denoising trajectory. Although the ordering is not strictly monotonic for every individual timestep, larger refinement budgets consistently shift the representation toward higher-frequency content overall.

These results quantitatively support the qualitative observation in Figure 1: heterogeneous refinement organizes feature groups according to their refinement histories, with sparsely refined features primarily carrying coarse global structure and more actively refined features progressively specializing toward finer visual details.

C Implementation Details

Table 8 summarizes the architecture, training, and sampling configurations of our main PerF models. We provide additional implementation details below.

Architecture details. We build PerF-B, PerF-L, and PerF-H upon the corresponding JiT architectures (Li & He, 2026). For each model scale, we replace the uniform hidden width with a fixed contraction–expansion schedule, where the ℓ-th Transformer block actively transforms only the first d_ℓ feature dimensions and the remaining dimensions bypass the block unchanged. The complete layer-wise width schedules are reported in Table 7. To isolate the effect of heterogeneous refinement from changes in backbone capacity, each variable-width backbone approximately matches the attention and MLP parameter budget of its corresponding JiT model by preserving

$$\sum_{\ell=1}^{L} d_\ell^2 \approx L d^2,$$

where d denotes the hidden dimension of the corresponding uniform-width JiT backbone. The P-to-A conditioning pathway introduces only a small additional parameter overhead. We use a low-rank bottleneck of rank $r = 128$ for PerF-B/L and $r = 192$ for PerF-H. PerF-H/16 and PerF-H/32 share the same Transformer depth, hidden dimensions, and layer-wise width schedule. Class conditioning and P-to-A conditioning are independently dropped with probability 0.1 during training.

Training details. Unless otherwise specified, we follow the optimization and noise-sampling setup of JiT (Li & He, 2026). All main models are trained for 600 epochs with Adam ($\beta_1 = 0.9$, $\beta_2 = 0.95$), using a global batch size of 1024, a constant learning rate of 2×10^{-4}, and no weight decay. The diffusion timestep is sampled according to $\mathrm{logit}(t) \sim \mathcal{N}(-0.8, 0.8^2)$. We track exponential moving averages with decay rates $\{0.9996, 0.9999\}$. The noise scale is set to 1.0 for 256×256 generation and 2.0 for 512×512 generation.

For the main scaling results in Tables 2 and 3, we additionally employ REPA (Yu et al., 2025; Singh et al., 2026) during only the first 100 training epochs. Extending representation alignment throughout the full training schedule degrades final generation quality, consistent with similar observations in (Shin et al., 2026; Wang et al., 2025). All ablation variants that require retraining are trained without REPA, unless otherwise stated. Sampling-only PG analyses instead reuse the corresponding trained checkpoint from the main experiments and vary only the sampling configuration.

Sampling details. We use the Heun solver with 50 sampling steps for all reported results. CFG is applied over the interval $[0.1, 1.0]$, while PG is applied over the full interval $[0, 1]$ unless otherwise specified. For PerF-B/16, PerF-L/16, PerF-H/16, and PerF-H/32, we use CFG scales of 2.8, 2.7, 3.0, and 2.9, respectively, with corresponding PG scales of 3.0, 2.8, 1.9, and 2.0. The detailed configurations for each model are summarized in Table 8.

D Implementation of Persistent-to-Active Conditioning

The P-to-A projector is instantiated and learned independently at each Transformer block. At block ℓ, the hidden representation is partitioned into active features $\mathbf{h}_\ell^a \in \mathbb{R}^{N \times d_\ell}$ and persistent features $\mathbf{h}_\ell^p \in \mathbb{R}^{N \times (D - d_\ell)}$. Persistent features are first normalized and then mapped through a low-rank two-layer projector,

$$\mathbf{m}_\ell^p = W_\ell^{(2)} \,\mathrm{SiLU}\Big(W_\ell^{(1)} \,\mathrm{RMSNorm}(\mathbf{h}_\ell^p)\Big),$$

where

$$W_\ell^{(1)} \in \mathbb{R}^{r \times (D - d_\ell)}, \qquad W_\ell^{(2)} \in \mathbb{R}^{6 d_\ell \times r}.$$

Thus, the bottleneck rank r directly controls the capacity of the persistent-conditioning pathway. The resulting token-wise modulation $\mathbf{m}_\ell^p \in \mathbb{R}^{N \times 6 d_\ell}$ is split into six components corresponding to the shift, scale, and gate parameters of the self-attention and MLP sublayers. These terms are added to the standard timestep- and class-conditioned AdaLN modulation (Wang et al., 2026b) before updating the active features. The persistent features themselves bypass the block unchanged and are concatenated back with the updated active representation afterward.

During training, the entire P-to-A pathway is independently dropped for each sample with probability 0.1, enabling the same model to produce predictions with and without persistent conditioning for Persistence Guidance. The final linear layer of each P-to-A projector is zero-initialized so that the persistent-conditioning branch is introduced as a zero residual at the start of training.

E GENERATION WITHOUT CLASSIFIER-FREE GUIDANCE.

Table 9 evaluates PerF without external classifier-free guidance. The improvement over the corresponding JiT baselines remains consistent across model scales, showing that the benefit of PerF does not rely on CFG or its interaction with the class-conditioning direction. Instead, the persistent-conditioning pathway and its induced guidance remain effective on their own, supporting the view that PerF provides an internal signal complementary to external semantic guidance. The simultaneous improvements in IS and recall further suggest that this benefit extends beyond a simple shift in sample fidelity.

Model	Params.	Epochs	FID ↓	IS ↑	Prec. ↑	Rec. ↑
ADM (Dhariwal & Nichol, 2021)	554M	400	10.94	–	0.69	0.63
PixelFlow-XL (Chen et al., 2025)	677M	320	12.23	103.3	0.63	0.66
PixNerd-XL (Wang et al., 2026a)	700M	320	15.61	88.9	0.59	0.68
DeCo-XL/16 (Ma et al., 2026a)	682M	320	14.88	88.2	0.60	0.68
PixelGen-XL/16 (Ma et al., 2026b)	676M	80	5.11	159.2	**0.72**	0.63
JiT-B/16 (Li & He, 2026)	131M	600	25.42	63.2	0.54	0.66
JiT-L/16 (Li & He, 2026)	458M	600	13.85	104.2	0.62	0.67
JiT-H/16 (Li & He, 2026)	953M	600	7.15	151.7	0.68	0.67
PerF-B/16	137M	600	12.53	104.0	0.63	0.66
PerF-L/16	471M	600	5.60	160.7	0.68	**0.69**
PerF-H/16	987M	600	**3.48**	**194.4**	0.70	**0.69**

Table 9: Comparison of class-conditional generation on ImageNet at 256×256 resolution without CFG. All other settings and notation follow Table 2.

Model / Training Recipe	FID ↓
JiT-B/16	3.66
PerF-B/16 w/o REPA	3.10
PerF-B/16 + REPA (full)	3.38
PerF-B/16 + REPA (first 100 ep.)	**2.81**

(a) REPA schedule in PerF-B/16.

Model	FID ↓	IS ↑
PixelREPA-B/16	3.17	284.6
PerF-B/16	**2.81**	**288.3**
PixelREPA-L/16	2.11	309.5
PerF-L/16	**1.91**	**311.2**
PixelREPA-H/16	1.81	317.2
PerF-H/16	**1.63**	**324.5**

(b) Comparison with PixelREPA.

Table 10: **Interaction with representation alignment.** Left: effect of different REPA schedules on PerF-B/16. Right: comparison with PixelREPA, which specifically adapts representation alignment to JiT.

F INTERACTION WITH REPRESENTATION ALIGNMENT.

We further study how Persistence Forcing interacts with representation alignment. Different from standard REPA (Yu et al., 2025; Singh et al., 2026), we apply the alignment objective only to the persistent feature subspace, leaving actively refined features unconstrained to model finer pixel-level details.

Table 10(a) shows that PerF already improves over JiT-B without representation alignment, indicating that its benefit does not depend on REPA. Moreover, applying REPA throughout training is less

effective than using no alignment, consistent with recent observations that prolonged representation alignment can interfere with later-stage generative learning (Shin et al., 2026; Wang et al., 2025). Restricting alignment to the first 100 epochs instead gives the strongest result, suggesting that representation alignment is most useful during early optimization while the persistent representation is still being formed.

We further compare against PixelREPA (Shin et al., 2026) in Table 10(b), which is specifically designed to make representation alignment effective for JiT. PerF consistently achieves better FID across B, L, and H scales, while also improving IS. Together, these results show that the gains of PerF cannot be explained by representation alignment alone. Instead, REPA serves as an auxiliary early-training objective that further improves the persistent representation learned by PerF.

G Architectural Overhead

Table 11 compares the parameter count and single-forward computational cost of PerF with the corresponding JiT backbones.

Scale	JiT		PerF		Overhead	
	Params.	**GFLOPs**	**Params.**	**GFLOPs**	**Params.**	**GFLOPs**
Base	131M	25.22	137M	26.51	+4.6%	+5.1%
Large	459M	87.94	471M	90.83	+2.6%	+3.3%
Huge	953M	181.95	987M	190.44	+3.6%	+4.7%

Table 11: **Per-forward computational cost on ImageNet** 256×256**.** PerF introduces modest parameter and FLOP overhead over the corresponding JiT backbones across model scales.

Despite introducing heterogeneous refinement and persistent-to-active conditioning, PerF incurs only modest overhead across model scales. The additional parameters remain below 5%, while the increase in single-forward FLOPs is approximately 3–5%. This shows that the persistent–active interaction can be incorporated with limited architectural overhead.

H Additional Qualitative Results

We provide additional class-conditional samples generated by PerF-H/16. As seen in Figure 9, Figure 10, and Figure 11, our method generates high-quality samples.

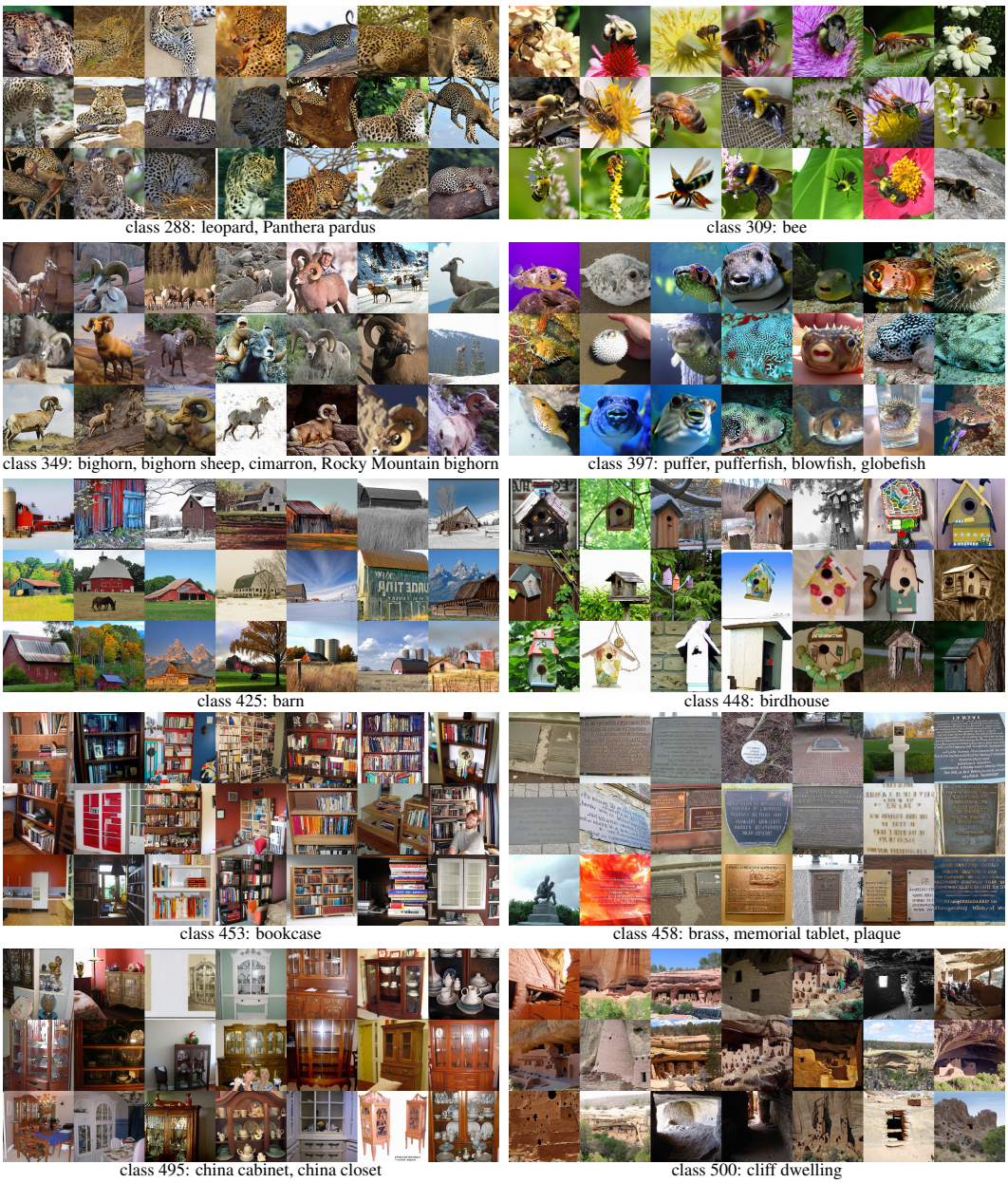

Figure 9: Uncurated class-conditional samples on ImageNet 256×256 using PerF-H/16



Figure 10: Uncurated class-conditional samples on ImageNet 256×256 using PerF-H/16

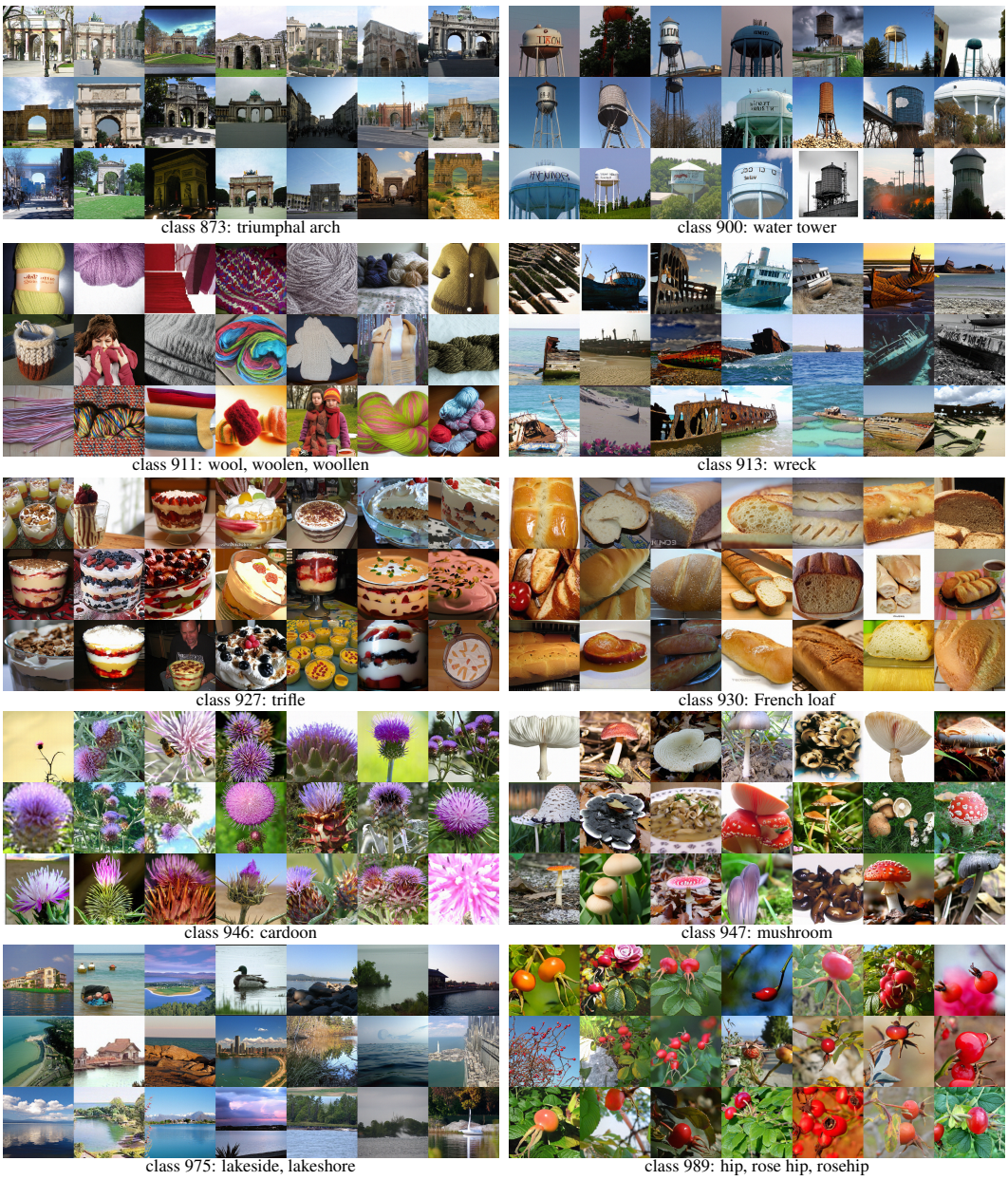

Figure 11: Uncurated class-conditional samples on ImageNet 256×256 using PerF-H/16.