DIFFERENT NOISE LEVELS, DIFFERENT NEEDS: DISCOVERING DIFFUSION PHASES FOR EFFICIENT DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models apply a shared denoiser across noise levels, even though the importance of its components changes throughout denoising. Recent work has explored this heterogeneity through phase-aware student design. However, phase boundaries and student capacity are typically prescribed without directly measuring how the teacher's parameter importance evolves. We propose a diffusion-distillation framework that identifies denoising phases from time-dependent parameter-usage profiles, which quantify the importance of different parameter groups to the denoising objective. These profiles guide both phase partitioning and the allocation of a fixed student-capacity budget across phases and network depth. We evaluate our method using DiT and convolutional U-Net-based denoisers on CIFAR-10, ImageNet, FFHQ, and LSUN Bedrooms, observing nonuniform phase structure across model families and datasets. Notably, **allocating student capacity according to this structure improves the quality-throughput trade-off over uniform allocation under matched parameter budgets**. These results highlight the teacher's evolving parameter importance as a practical signal for efficient denoiser design.

1 INTRODUCTION

Diffusion models establish a versatile foundation for generative modeling across images (Ho et al., 2020; Rombach et al., 2021; Meng et al., 2021), video (Ho et al., 2022; Singer et al., 2022), audio (Kong et al., 2020), and graphs (Jo et al., 2022; Vignac et al., 2022; Berman et al., 2024). This generative performance relies on an iterative trajectory in which a neural network repeatedly transforms pure noise into structured representations. However, this multi-step process creates a severe computational bottleneck: inference requires repeated evaluation of an overparameterized denoiser, making deployment expensive even with moderate step counts. To ease this burden, diffusion distillation (Luhman & Luhman, 2021; Salimans & Ho, 2022) transfers the denoiser's generative capacity into a streamlined, compact network while maintaining sample quality.

Existing diffusion-distillation methods enhance efficiency across multiple dimensions. Sampler and trajectory distillation techniques compress the reverse process into fewer steps, for example, through progressive distillation (Salimans & Ho, 2022), consistency objectives (Song et al., 2023), adversarial training (Xiao et al., 2021; Sauer et al., 2023), or direct prediction of multi-step transitions (Berthelot et al., 2023; Kim et al., 2024b). Knowledge-distillation approaches transfer the teacher model's outputs, scores, intermediate representations, or generated samples to a student denoiser (Luhman & Luhman, 2021; Meng et al., 2023; Yin et al., 2024). In contrast, architecture-oriented methods reduce the cost of individual denoising evaluations by employing smaller networks, structured pruning, low-rank approximations, feature inheritance, or dynamic computation (Kim et al., 2024a; Fang et al., 2023; Bolya & Hoffman, 2023; Ma et al., 2024). These approaches are largely orthogonal: reducing the number of function evaluations does not necessarily decrease the cost of each evaluation, and architectural compression can often be combined with accelerated samplers. However, most architecture-oriented approaches treat the denoiser as a single model compressed uniformly across the entire diffusion trajectory. Under a fixed parameter budget, this formulation implicitly assumes that every noise level requires identical computational and representational capacity.

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

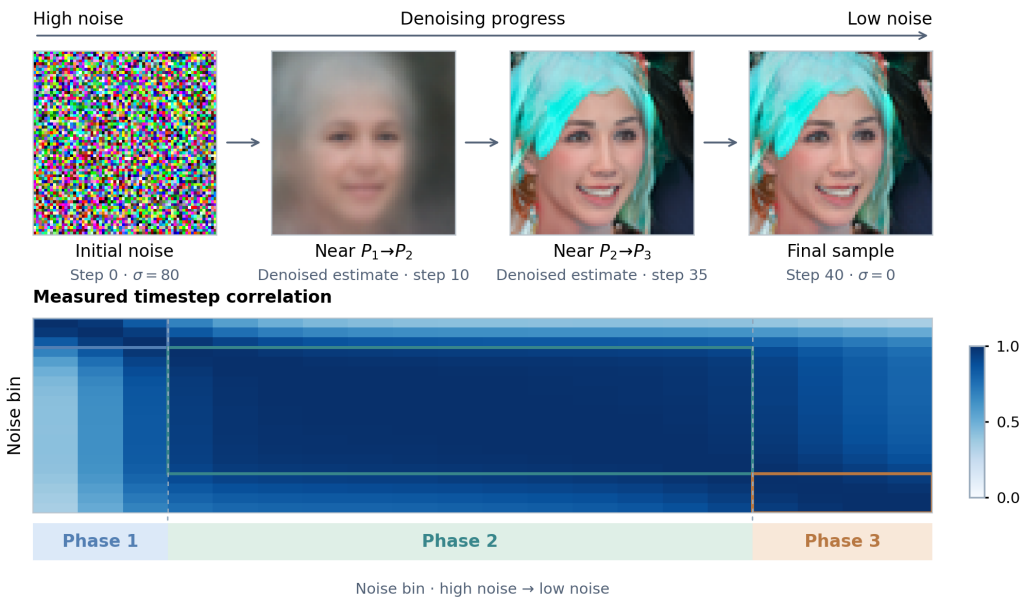

Figure 1: **Denoising across a diffusion process. Top:** An FFHQ trajectory showing initial noise, clean-image estimates near the two phase boundaries, and the final sample. The estimates progress from coarse facial structure to a refined image. **Bottom:** Correlations between parameter-usage profiles across noise levels distributed into 20 bins, with the three identified phases highlighted.

As illustrated in Figure 1, this assumption neglects the phase structure inherent in diffusion processes. The denoising process represents a continuum of related tasks indexed by noise level. At high noise levels, the model establishes coarse semantic and spatial structures. As noise decreases, the model incrementally resolves object shapes and composition, followed by refinement of local appearance and high-frequency details (Meng et al., 2021; Lee et al., 2024). From a multi-task learning perspective, a diffusion model utilizes a single set of parameters across all tasks. While complete parameter sharing is efficient when tasks at different noise levels require similar computations, it can lead to interference or redundant capacity when their computational requirements diverge significantly (Ma et al., 2025).

We advance the understanding of the phase structure of diffusion processes by introducing Phase-Aware Capacity-Efficient diffusion distillation (PACE), a framework that converts the internal phase structure of a diffusion process into a principled signal for architecture distillation. Instead of relying on manual heuristics, PACE quantifies the evolution of representation learning requirements during denoising by measuring the teacher model's time-dependent parameter-usage profile. This profile reveals coherent intervals with consistent computational needs and distinct transition points where parameter demands change. By leveraging this intrinsic structure, PACE simultaneously identifies boundaries that optimize the profile-similarity objective and allocates a fixed student-capacity budget across both specialized phase models and network depth.

Our main contributions are:

- We characterize the phenomenon of *diffusion phases* through time-dependent parameter usage in pretrained denoisers, establishing a direct link between evolving representation-learning demands and parameter-sharing inefficiencies across noise levels.
- We introduce PACE, which uses these profiles to partition the denoising trajectory and allocate a shared student-capacity budget across phases and network depth.
- We evaluate our method across various architectures and diverse datasets. Across all benchmarks, PACE consistently outperforms uniform capacity baselines, delivering superior sample quality and throughput under matched parameter budgets.

2 Related Work

2.1 Efficient Denoising Networks

Architectural acceleration reduces the cost of individual denoiser evaluations. Zhang et al. (2024) combine a shared encoder with stage-specific decoders. DyDiT (Zhao et al., 2025) adapts transformer width across timesteps and selectively skips spatial-token computation, while ALTER (Yang et al., 2026) jointly optimizes layer pruning, temporal expert routing, and model weights. TinyFusion (Fang et al., 2025) learns a shallower transformer through recovery-aware layer pruning.

PACE uses sensitivity profiles measured in a frozen teacher to allocate a fixed aggregate parameter budget across phase-specialized students and network depth. This targets per-evaluation cost, a complementary axis to reducing sampling steps through progressive distillation (Salimans & Ho, 2022) or consistency models (Song et al., 2023).

2.2 Diffusion Phases and Denoising-Task Specialization

Denoising requirements change along the diffusion trajectory. eDiff-I (Balaji et al., 2022) observes stronger text-conditioning influence during early generation, motivating experts for different noise intervals. Theoretical studies also identify distinct regimes: Biroli et al. (2024) characterize structure formation and subsequent collapse toward training examples under an optimal empirical-score model, while Ventura et al. (2025) identify manifold-coverage and consolidation phases through score-Jacobian spectral analysis. MEME (Lee et al., 2024) uses frequency analysis to tailor the balance of convolution and self-attention across timestep intervals. We share its goal of matching architecture to noise-dependent requirements, but use activation sensitivity to guide both phase partitioning and capacity allocation.

Temporal heterogeneity also motivates methods that address cross-timestep interference. Go et al. (2023) use task clustering and multi-task optimization. Decouple-Then-Merge (Ma et al., 2025) fine-tunes interval-specific models and merges them into one denoiser, while Denoising Task Routing (Park et al., 2024b) and Switch-DiT (Park et al., 2024a) introduce timestep-dependent computation paths. Our framework retains separate phase specialists at inference, with their capacities determined by the teacher's measured sensitivity.

3 Phase-Aware Capacity-Efficient Diffusion Distillation

We present PACE, a phase-aware framework that transforms a frozen diffusion teacher into a suite of noise-specialized student denoisers under a fixed aggregate budget P_{tot}. As illustrated in Figure 2, PACE decouples offline structural compilation from online phase-conditioned inference. First, we measure time-dependent representation-sensitivity profiles to discover contiguous trajectory phases. Second, we distribute the total budget P_{tot} into phase allocations P_k to compile specialized student architectures. Finally, we dynamically route each noise level s to its designated phase specialist, evaluating exactly one lightweight network per denoising call.

3.1 Problem Setup

Let D_Θ be a pretrained, frozen diffusion model, and let $\mathcal{S} = \{s_1, \ldots, s_T\}$ denote the ordered sequence of diffusion noise levels used during sampling. Our goal is to discover an optimal partition of this trajectory into K contiguous, non-overlapping phases,

$$\mathcal{B} = \{B_1, \ldots, B_K\}, \qquad B_1 \prec \cdots \prec B_K,$$

where each B_k is a contiguous interval of noise levels and $\prec$ denotes order along the sampling trajectory, from high to low noise.

We refer to a *phase* as a segment of the trajectory during which the denoiser shows a consistent dependence on the parameters. More precisely, for each diffusion level s, we measure the contribution of a given set of internal parameter groups $\mathcal{G}$ by means of a nonnegative importance profile $\mathbf{u}_s \in \mathbb{R}^{|\mathcal{G}|}$. The objective of phase discovery is therefore to find boundaries that maximise the similarity of the profiles within each phase while identifying the clear transition points between consecutive phases.

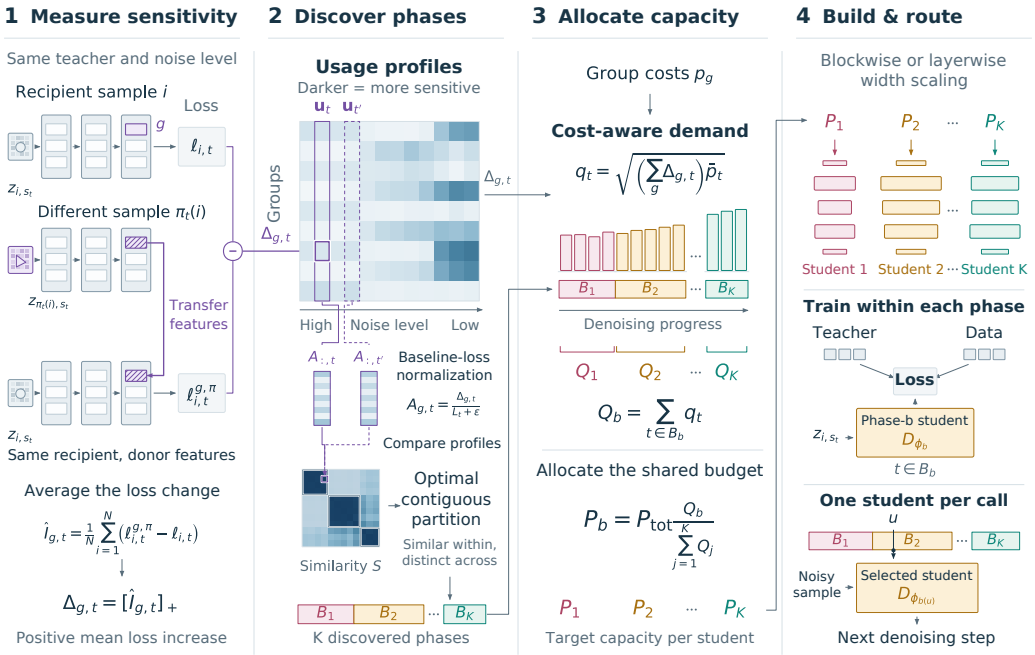

Figure 2: **PACE overview.** We measure the teacher's sensitivity by permuting individual activation groups between examples at the same noise level and recording the change in denoising loss. Similarity between the resulting profiles guides contiguous phase partitioning, while sensitivity-weighted parameter costs guide capacity allocation across phases and network depth. The resulting specialist architectures are trained by distillation. During sampling, each denoiser evaluation executes only the specialist assigned to the current noise level; profiling is performed only offline.

We estimate the profiles on a fixed calibration set $\mathcal{C} = \{x_i\}_{i=1}^{N}$, pairing baseline and perturbed evaluations on the same noisy inputs. This reduces sampling variability when comparing interventions. The resulting profiles characterize the teacher's dependence on individual computational groups across noise levels and guide phase discovery and capacity allocation.

3.2 Noise-Conditioned Activation Permutation Importance

To identify phases along the denoising trajectory, we measure how strongly the denoiser depends on individual computational groups at each noise level. A *group* $g \in \mathcal{G}$ is a unit whose complete output activation tensor can be exchanged between calibration examples; the set of all groups is denoted by $\mathcal{G}$. For a Diffusion Transformer, a group $g = (\ell, a)$ corresponds to attention head a in layer ℓ, and h_g is its output for all tokens before concatenation with the other heads. For a convolutional U-Net, $g = (\ell, c)$ denotes output channel c in layer ℓ, and h_g is its complete spatial feature map.

We seek an importance measure that captures a group's functional effect on the denoising objective rather than its parameter or activation magnitude. We therefore adapt Permutation Feature Importance (PFI) (Altmann et al., 2010) to intermediate activations. Specifically, we replace a group's activation with that produced by another calibration example at the same noise level and measure the resulting increase in loss. This intervention disrupts the correspondence between the activation and its original example while preserving its shape and marginal scale.

For brevity, let $h_{g,i,t} = h_g(i, s_t)$ and $\ell_{i,t} = \ell_{i,s_t}$ denote the activation of group g and the loss for example i at diffusion level s_t, respectively. For each s_t, we draw a seeded, fixed-point-free permutation π_t over the N calibration examples and apply $h_{g,i,t} \leftarrow h_{g,\pi_t(i),t}, \pi_t(i) \neq i$. Then, we recompute the downstream forward pass. All groups evaluated at s_t use the same permutation π_t. Permutations are restricted to examples at the same noise level; cross-timestep swaps would introduce distribution shifts that confound the importance estimates.

Let $\ell_{i,t}^{g,\pi}$ denote the loss under this intervention. We define the baseline loss and PFI score as

$$L_t = \frac{1}{N}\sum_{i=1}^{N} \ell_{i,t}, \qquad \widehat{I}_{g,t} = \frac{1}{N}\sum_{i=1}^{N} \left(\ell_{i,t}^{g,\pi} - \ell_{i,t}\right).$$

Thus, $\widehat{I}_{g,t} > 0$ indicates that disrupting the instance-specific information carried by group g degrades denoising performance at s_t. For phase discovery and capacity allocation, we construct the nonnegative importance profile

$$\Delta_{g,t} = \left[\widehat{I}_{g,t}\right]_+, \qquad \mathbf{u}_t = \left(\Delta_{g,t}\right)_{g\in\mathcal{G}},$$

where $[a]_+ = \max(a, 0)$. We retain signed scores for diagnostic analysis but truncate negative values when constructing the profiles used by the method.

Activation PFI measures functional dependence through a finite, loss-based intervention. It uses activations produced by the teacher itself, preserving the empirical marginal distribution of the intervened group while disrupting its alignment with the recipient example. These measurements provide a task-dependent signal for identifying phases and allocating student capacity.

3.3 Automatic Phase Discovery

To identify contiguous phases along the sampling trajectory, the method compares group-sensitivity profiles across timesteps. At each timestep t, the profile $A_{:,t}$ expresses each group's sensitivity relative to the baseline denoising loss L_t. Its entries and the similarities between profiles are defined as

$$A_{g,t} = \frac{\Delta_{g,t}}{L_t + \epsilon}, \qquad S_{t,t'} = \mathrm{sim}(A_{:,t}, A_{:,t'}),$$

where $\epsilon > 0$ stabilizes the normalization and sim is a similarity measure between profiles, such as Pearson correlation, with larger values indicating more similar sensitivity patterns. In the resulting similarity matrix $S \in \mathbb{R}^{T\times T}$, blocks of high similarity along the diagonal mark consecutive timesteps at which the denoiser responds similarly to interventions across groups. Each block therefore suggests a phase of the sampling trajectory during which the same groups play similar roles in denoising.

Using this similarity matrix, we formulate phase discovery as a trajectory-partitioning objective. For any candidate interval B, let $W(B) = \sum_{t\neq t'\in B} S_{t,t'}$ quantify internal profile cohesion. For a partition $\mathcal{B} = \{B_1, \ldots, B_K\}$ into K contiguous, we maximize the objective $J(\mathcal{B})$:

$$J(\mathcal{B}) = \sum_{B\in\mathcal{B}} \frac{W(B)}{|B|} - \lambda_{\mathrm{sep}} \sum_{\substack{B,B'\in\mathcal{B}\\ B\neq B'}} \sum_{t\in B,\, t'\in B'} S_{t,t'}.$$

The first term rewards high average intra-phase similarity normalized by interval length $|B|$, while the second term penalizes cross-phase similarity scaled by separation weight λ_{sep}. All summations iterate over ordered bin pairs.

The additive interval structure of $J(\mathcal{B})$ permits exact dynamic programming, which finds the maximizing partition for each $K \in \{1, \ldots, K_{\max}\}$ in $O(K_{\max}T^2)$ time. In automatic mode, we select the partition with the largest objective value, breaking ties toward smaller K. Alternatively, the number of phases can be specified manually, and dynamic programming then optimizes the boundaries for that fixed K. The correlation matrix also provides an interpretable basis for manual partitioning: users can select both the phase count and boundaries by inspecting regions of high within-phase similarity and transitions between them.

3.4 Fixed-Budget Capacity Allocation

After establishing discrete trajectory phases, the underlying sensitivity profiles are translated into specific capacity allocations for the phase-specialized student models. To determine the required capacity for each phase, the computational demand at each timestep t is quantified by weighting group sensitivity $\Delta_{g,t}$ against group parameter costs p_g:

$$w_{g,t} = \frac{\Delta_{g,t}}{\sum_{g'\in\mathcal{G}} \Delta_{g',t} + \epsilon}, \quad \bar{p}_t = \sum_g w_{g,t} p_g, \quad q_t = \sqrt{\Big(\sum_g \Delta_{g,t}\Big)\bar{p}_t}, \tag{1}$$

where $w_{g,t}$ represents normalized group sensitivity, $\bar{p}_t$ is the weighted average group cost, and q_t measures total timestep demand. Note that $w_{g,t}$ and $\bar{p}_t$ are introduced primarily for numerical stability via ϵ; ignoring this stabilization term, Eq. 1 simplifies to $q_t \approx \sqrt{\sum_g \Delta_{g,t}\, p_g}$, i.e., the timestep demand is the square root of the sensitivity-weighted parameter cost, consistent with the intuition that model capacity (parameter count) scales quadratically with layer width. Summing q_t across all timesteps in phase B_k yields the cumulative phase demand $Q_k = \sum_{t \in B_k} q_t$. We then allocate the target student parameter budget P_{tot} across phases in direct proportion to their measured representation demands:

$$P_k = P_{\text{tot}} \frac{Q_k}{\sum_j Q_j}, \qquad \sum_k P_k = P_{\text{tot}}.$$

We compare our proposed phase-aware layerwise allocation with three reference designs under the same target budget P_{tot} for all stored student parameters. All phase-specialized designs use the same partition. **Global** uses one student throughout the trajectory. **Uniform blockwise** divides the budget equally among phase specialists, with $P_k = P_{\text{tot}}/K$. Their comparison evaluates the benefit of specialization at fixed aggregate capacity.

Phase-aware blockwise, an ablation, replaces equal budgets with budgets proportional to Q_k, evaluating sensitivity-guided allocation across phases. These reference designs scale all layers within each student by a common width factor, preserving the teacher's relative layer widths. **Phase-aware layerwise** retains the same phase budgets and additionally redistributes capacity across network depth. Its comparison with Phase-aware blockwise evaluates the benefit of adapting each specialist's internal allocation. Within each phase B_b, group budgets are proportional to accumulated cost-aware importance:

$$r_{b,g} = \sum_{t \in B_b} \sqrt{\Delta_{g,t} p_g}, \qquad P_{b,g} = P_b \frac{r_{b,g}}{\sum_{g' \in \mathcal{G}} r_{b,g'}},$$

Budgets are aggregated for groups that share a structural width control, then converted into a realizable student architecture.

3.5 Routing and Training

The resulting students learn to denoise within their assigned phases. The training objective balances teacher imitation with clean-data reconstruction, while profiling and allocation remain independent of the chosen loss. During sampling, the current noise level selects a single specialist to evaluate; the others remain stored but inactive. Appendix A.5 details the routing rule and training objective.

4 Experiments

We evaluate PACE across four image-generation datasets and two denoiser families. Our goal is to improve sampling efficiency by allocating denoising capacity according to timestep-dependent parameter usage. The objectives of this empirical study are threefold:

- Analyze whether timestep-conditioned parameter-usage profiles exhibit distinct and consistent phases along the denoising trajectory.
- Examine how allocating a fixed student-capacity budget across phases and network depth affects generation quality.
- As our primary objective is efficient generation, assess the quality–throughput trade-off of phase-specialized students relative to global and uniform allocation strategies and existing efficient-denoising methods.

4.1 Experimental Setup

We evaluate PACE on CIFAR-10, ImageNet, FFHQ, and LSUN-Bedrooms, spanning diffusion transformers and convolutional U-Net denoisers. For image generation, we assess sample quality using FID, sFID, precision, recall, and Inception Score, and measure efficiency through sampling throughput. Unless stated otherwise, methods within each comparison share the same evaluation protocol, hardware, and sampling configuration.

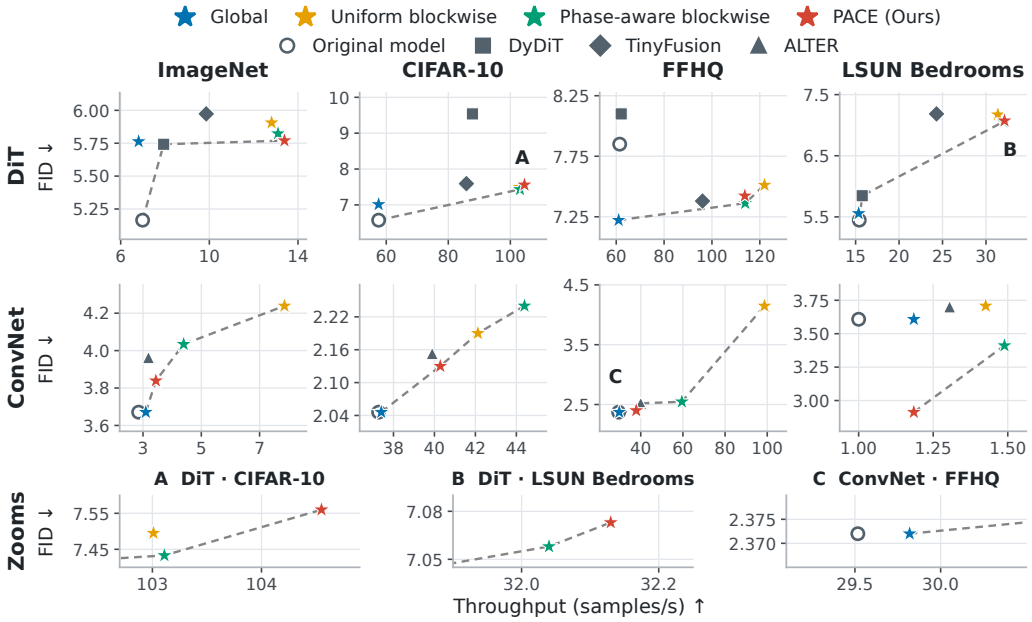

Figure 3: **FID–throughput tradeoffs for DiT and ConvNet.** Stars denote allocation variants (PACE: phase-aware layerwise); gray markers denote baselines. Dashed lines connect Pareto-optimal points; zooms A–C enlarge marked clusters. Lower FID and higher throughput are preferred. Compare throughput within panels; hardware and evaluation protocols differ.

Table 1: Quality, coverage, and throughput for the original teacher, DyDiT (Zhao et al., 2025), TinyFusion (Fang et al., 2025), ALTER (Yang et al., 2026), and PACE with phase-aware layerwise allocation. Precision (P) and recall (R) distinguish fidelity from coverage, complementing the FID–throughput curves in Figure 3. Throughput is measured in samples per second (S/s); compare methods within each model-dataset pair.

Model	Method	ImageNet					CIFAR-10				
		FID ↓	P ↑	R ↑	IS ↑	S/s ↑	FID ↓	P ↑	R ↑	IS ↑	S/s ↑
DiT	Original	5.165	0.823	0.669	269.96	6.99	6.574	0.704	0.618	9.68	57.55
	DyDiT	5.742	0.842	0.626	253.41	7.94	9.538	0.689	0.576	9.36	87.78
	TinyFusion	5.973	0.836	0.624	226.70	9.86	7.593	0.679	0.602	9.55	85.84
	PACE	5.770	0.838	0.624	240.65	13.39	7.560	0.683	0.610	9.37	104.55
ConvNet	Original	3.671	0.716	0.648	46.967	3.096	2.046	0.679	0.625	9.850	37.403
	ALTER	3.963	0.674	0.594	41.64	3.187	2.1531	0.642	0.618	9.251	39.891
	PACE	3.839	0.696	0.629	42.274	3.442	2.130	0.669	0.621	9.441	40.286
Model	**Method**	**FFHQ**					**LSUN Bedrooms**				
		FID ↓	P ↑	R ↑	IS ↑	S/s ↑	FID ↓	P ↑	R ↑	IS ↑	S/s ↑
DiT	Original	7.850	0.743	0.531	4.57	61.45	5.446	0.559	0.533	2.41	15.42
	DyDiT	8.100	0.728	0.519	4.35	62.08	5.846	0.558	0.513	2.33	15.76
	TinyFusion	7.380	0.719	0.521	4.53	96.06	7.185	0.523	0.499	2.30	24.31
	PACE	7.423	0.725	0.524	4.49	113.73	7.073	0.534	0.502	2.31	32.13
ConvNet	Original	2.372	0.725	0.571	3.328	29.819	3.608	0.659	0.453	2.207	1.000
	ALTER	2.520	0.704	0.565	3.198	39.989	3.701	0.655	0.497	2.200	1.305
	PACE	2.400	0.751	0.521	3.206	37.858	2.913	0.661	0.455	2.301	1.185

All allocation variants target the same aggregate student parameter budget, set to the corresponding teacher's parameter count. At each denoiser evaluation, only the routed specialist executes. This constrains stored model capacity while allowing the computational cost to vary across noise levels.

Table 2: **Capacity-allocation results across datasets and model families.** All variants target the teacher's parameter count. Highlighted rows show our proposed phase-aware layerwise allocation. Throughput is reported in samples per second (S/s). Hardware and evaluation protocols differ across datasets and model families, so throughput should not be compared across them.

Model	Allocation	ImageNet		CIFAR-10		FFHQ		LSUN Bedrooms	
		FID ↓	S/s ↑	FID ↓	S/s ↑	FID ↓	S/s ↑	FID ↓	S/s ↑
DiT	Global	5.763	6.81	7.013	57.55	7.222	61.01	5.556	15.35
	Uniform blockwise	5.906	12.81	7.495	103.01	7.512	121.95	7.172	31.37
	Phase-aware blockwise	5.823	13.10	7.433	103.11	7.362	113.89	7.058	32.04
	Phase-aware layerwise	5.770	13.39	7.560	104.55	7.423	113.73	7.073	32.13
ConvNet	Global	3.671	3.096	2.046	37.403	2.372	29.819	3.608	1.185
	Uniform blockwise	4.239	7.853	2.190	42.121	4.149	98.693	3.708	1.426
	Phase-aware blockwise	4.034	4.395	2.240	44.397	2.546	59.508	3.411	1.489
	Phase-aware layerwise	3.839	3.442	2.130	40.286	2.400	37.858	2.913	1.185

Full details of the image datasets, model architectures, training, and evaluation protocols are provided in Appendix A. We additionally examine whether the identified phase structure extends beyond image generation through sensitivity profiling of an audio diffusion model. The sensitivity profiling is presented in Appendix B.

4.2 Temporal Structure of Parameter Usage

The image U-Net profiles (Figure 5) reveal phases of unequal length, with transition locations that vary across model–dataset pairs. This variation motivates deriving phase boundaries from each teacher's sensitivity profile. Appendix C provides the full profiles, phase-selection choices, and detailed statistics.

4.3 Comparison with Efficient Denoisers

Table 1 compares PACE with the original teachers, DyDiT (Zhao et al., 2025) and TinyFusion (Fang et al., 2025) for DiTs, and ALTER (Yang et al., 2026) for convolutional U-Nets. Figure 3 places these methods alongside our allocation variants in the FID–throughput plane. In this comparison, PACE denotes the complete phase-aware layerwise design.

Diffusion transformers. Across the four datasets, PACE achieves 1.82–$2.08\times$ the throughput of the corresponding original teacher. On ImageNet, it produces 13.39 samples/s at FID 5.770, compared with 9.86 samples/s at FID 5.973 for TinyFusion. Relative to DyDiT, throughput increases by $1.69\times$, with FID changing from 5.742 to 5.770. On CIFAR-10 and LSUN Bedrooms, PACE also improves both reported FID and throughput over TinyFusion. On FFHQ, it offers 18.4% higher throughput than TinyFusion, with FID 7.423 versus 7.380. These results show that phase-specialized students provide competitive high-throughput operating points, although acceleration is not uniformly quality-preserving. For example, on LSUN Bedrooms, the $2.08\times$ throughput gain over the original teacher accompanies an FID increase from 5.446 to 7.073.

Convolutional U-Nets. PACE achieves lower FID than ALTER on all four datasets, with higher throughput on ImageNet and CIFAR-10 and lower throughput on FFHQ and LSUN Bedrooms. On ImageNet, FID improves from 3.963 to 3.839 while throughput increases from 3.187 to 3.442 samples/s. On FFHQ, PACE attains FID 2.400, close to the teacher's 2.372 and below ALTER's 2.520, at 37.858 rather than 39.989 samples/s for ALTER. On LSUN Bedrooms, FID decreases to 2.913 from 3.701 for ALTER and 3.608 for the teacher, corresponding to a 19.3% reduction relative to the teacher. Here, PACE favors quality over ALTER's higher throughput of 1.305 versus 1.185 samples/s.

The additional metrics reveal fidelity–coverage trade-offs that FID alone does not capture. For example, on FFHQ with U-Nets, PACE increases precision from ALTER's 0.704 to 0.751 but reduces recall from 0.565 to 0.521.

4.4 Effect of Capacity Allocation

Table 2 compares the four allocation strategies in Section 3.4 under a shared target budget. The uniform variant already uses the phase partition, so its comparison with the phase-aware variants tests capacity allocation rather than the choice of phase boundaries.

Specialization and phase-dependent budgeting. Uniform blockwise increases throughput over Global in all eight model–dataset settings, but also increases FID in every setting. Thus, specialization alone provides acceleration without necessarily preserving sample quality. Replacing equal budgets with phase-aware blockwise allocation lowers FID in seven of the eight settings. For DiTs, it lowers FID on all four datasets and increases throughput on ImageNet, CIFAR-10, and LSUN Bedrooms. On FFHQ, FID improves from 7.512 to 7.362 while throughput decreases from 121.95 to 113.89 samples/s. The quality recovery is particularly pronounced for the FFHQ U-Net, where FID decreases from 4.149 to 2.546, with throughput changing from 98.693 to 59.508 samples/s. The exception is the CIFAR-10 U-Net: phase-aware blockwise allocation increases throughput from 42.121 to 44.397 samples/s but also increases FID from 2.190 to 2.240.

Redistribution across network depth. The effect of layerwise allocation depends on the model family. For U-Nets, it lowers FID relative to phase-aware blockwise allocation on all four datasets, but reduces throughput in each case. On LSUN Bedrooms, for example, FID decreases from 3.411 to 2.913 while throughput decreases from 1.489 to 1.185 samples/s. For DiTs, layerwise allocation improves both metrics on ImageNet, reducing FID from 5.823 to 5.770 and increasing throughput from 13.10 to 13.39 samples/s. However, phase-aware blockwise allocation has lower FID on the other three datasets; on FFHQ, it also has slightly higher throughput.

Together, these results demonstrate that a shared aggregate parameter budget can yield different quality–throughput trade-offs depending on where capacity is placed. Since each denoiser call executes only its routed specialist, matching total stored capacity does not match the computation performed along the sampling trajectory. Sensitivity-guided allocation improves several measured operating points, while the choice between blockwise and layerwise designs remains dependent on the dataset and the desired quality–throughput balance.

Qualitative comparisons. Appendix D presents matched-seed samples from the teacher and all four allocation variants, with shared initial noise and, where applicable, conditioning classes. The displayed examples illustrate how broad object, facial, and scene structure can be retained while local appearance changes across allocations, complementing the distributional metrics with direct comparisons of individual samples.

5 Discussion

PACE uses the teacher's evolving sensitivity profiles to guide phase discovery and capacity allocation across noise levels and network depth. Allocating phase budgets according to sensitivity lowers FID relative to equal budgets in seven of eight settings. Allocation granularity depends on architecture: layerwise redistribution further lowers FID for all evaluated convolutional denoisers at reduced throughput, whereas blockwise allocation achieves lower FID on three of four DiT datasets. Since each denoising call executes only one specialist, equal aggregate parameter budgets can yield different computational costs. These findings support using the teacher not only as a source of training targets but also as a guide to student design, considering both where capacity is allocated and when it is executed.

Limitations and future work. PACE uses hard phase boundaries, although neighboring phases can retain similar sensitivity patterns. This motivates exploring overlapping phases or partial parameter sharing between specialists. The sensitivity estimates also depend on the calibration data, making boundary stability and transfer across data distributions important directions for further study. Finally, offline profiling and the construction and training of multiple specialists introduce additional complexity. More efficient profiling and shared student components are promising directions for reducing this overhead.

AI USE STATEMENT

We used Claude Code and OpenAI Codex to assist with code development and manuscript writing. The authors reviewed and edited the AI-assisted text and reviewed and tested the AI-assisted code. The authors take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

To support reproducibility, we describe the importance estimation, phase discovery, and capacity allocation procedures in Section 3. The appendix provides the denoiser architectures (Appendix A.3), datasets and preprocessing (Appendix A.1), and evaluation protocol (Appendix A.2).

ETHICS STATEMENT

This work studies capacity allocation for efficient diffusion models. Improving sampling efficiency may lower the computational cost of image generation, but may also facilitate harmful uses, including the production of misleading content. Distillation does not inherently remove biases or potentially harmful behaviors inherited from pretrained teachers and their training data.

REFERENCES

André Altmann, Laura Toloşi, Oliver Sander, and Thomas Lengauer. Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347, 2010.

Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, et al. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

Nimrod Berman, Eitan Kosman, Dotan Di Castro, and Omri Azencot. Reviving life on the edge: Joint score-based graph generation of rich edge attributes. *arXiv preprint arXiv:2402.04046*, 2024.

David Berthelot, Arnaud Autef, Jierui Lin, Dian Ang Yap, Shuangfei Zhai, Siyuan Hu, Daniel Zheng, Walter Talbott, and Eric Gu. Tract: Denoising diffusion models with transitive closure time-distillation. *arXiv preprint arXiv:2303.04248*, 2023.

Giulio Biroli, Tony Bonnaire, Valentin De Bortoli, and Marc Mézard. Dynamical regimes of diffusion models. *Nature Communications*, 15(1):9957, 2024.

Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *2023 IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW)*, pp. 4599–4603. IEEE, 2023.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://arxiv.org/abs/2105.05233.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Structural pruning for diffusion models, 2023. *URL https://arxiv. org/abs/2305.10924*, 2023.

Gongfan Fang, Kunjun Li, Xinyin Ma, and Xinchao Wang. Tinyfusion: Diffusion transformers learned shallow. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18144–18154. IEEE, 2025.

Hyojun Go, Kim Kim, Yunsung Lee, Seunghyun Lee, Shinhyeok Oh, Hyeongdon Moon, and Seungtaek Choi. Addressing negative transfer in diffusion models. *Advances in Neural Information Processing Systems*, 36:27199–27222, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022.

Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. In *International conference on machine learning*, pp. 10362–10383. PMLR, 2022.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 4396–4405. IEEE, 2019.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Bo-Kyeong Kim, Hyoung-Kyu Song, Thibault Castells, and Shinkook Choi. Bk-sdm: A lightweight, fast, and cheap version of stable diffusion. In *European Conference on Computer Vision*, pp. 381–399. Springer, 2024a.

Dongjun Kim, Chieh-Hsin Lai, WeiHsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. In *International Conference on Learning Representations*, volume 2024, pp. 44493–44525, 2024b.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yunsung Lee, JinYoung Kim, Hyojun Go, Myeongho Jeong, Shinhyeok Oh, and Seungtaek Choi. Multi-architecture multi-expert diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 13427–13436, 2024.

Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*, 2021.

Qianli Ma, Xuefei Ning, Dongrui Liu, Li Niu, and Linfeng Zhang. Decouple-then-merge: Finetune diffusion models as multi-task learning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23281–23291. IEEE, 2025.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15762–15772. IEEE, 2024.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14297–14306. IEEE, 2023.

Byeongjun Park, Hyojun Go, Jin-Young Kim, Sangmin Woo, Seokil Ham, and Changick Kim. Switch diffusion transformer: Synergizing denoising tasks with sparse mixture-of-experts. In *European Conference on Computer Vision*, pp. 461–477. Springer, 2024a.

Byeongjun Park, Sangmin Woo, Hyojun Go, Jin-Young Kim, and Changick Kim. Denoising task routing for diffusion models. In *International Conference on Learning Representations*, volume 2024, pp. 10334–10358, 2024b.

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11400–11410. IEEE, 2022.

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

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. *URL https://arxiv. org/abs/2112.10752*, 2112, 2021.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. arxiv 2014. *arXiv preprint arXiv:1409.0575*, 2014.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation, 2023. *URL https://arxiv. org/abs/2311.17042*, 2023.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL `https://arxiv.org/abs/2011.13456`.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Enrico Ventura, Beatrice Achilli, Gianluigi Silvestri, Carlo Lucibello, and Luca Ambrogioni. Manifolds, random matrices and spectral gaps: The geometric phases of generative diffusion. In *International Conference on Learning Representations*, volume 2025, pp. 88317–88338, 2025.

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. *arXiv preprint arXiv:2209.14734*, 2022.

Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans. *arXiv preprint arXiv:2112.07804*, 2021.

Xiaomeng Yang, Lei Lu, Qihui Fan, Changdi Yang, Juyi Lin, Yanzhi Wang, Xuan Zhang, and Shangqian Gao. Alter: All-in-one layer pruning and temporal expert routing for efficient diffusion generation. *Advances in Neural Information Processing Systems*, 38:128571–128599, 2026.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6613–6623, 2024.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Huijie Zhang, Yifu Lu, Ismail Alkhouri, Saiprasad Ravishankar, Dogyoon Song, and Qing Qu. Improving training efficiency of diffusion models via multi-stage framework and tailored multi-decoder architecture. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7372–7381. IEEE, 2024.

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.

A EXPERIMENTAL SETUP

A.1 DATASETS

We consider four RGB image-generation benchmarks spanning object categories, faces, and indoor scenes. The counts below refer to distinct real images used for distillation; noise corruptions are sampled online during training. All inputs are scaled to $[-1, 1]$.

- **CIFAR-10** (Krizhevsky et al., 2009). We use the standard training split of 50,000 images at the native 32×32 resolution, with conditioning on 10 classes. The official 10,000-image test split supplies monitoring data.
- **ImageNet-1K** (Russakovsky et al., 2014). We use the ILSVRC2012 classification training split of 1,281,167 images, stored as WebDataset shards, with conditioning on 1,000 classes.
- **FFHQ** (Karras et al., 2019). We use all 70,000 aligned faces for unconditional generation, downsampled offline from 256×256 to 64×64 using Lanczos interpolation. Monitoring uses image IDs 60000–69999, a 10,000-image subset that overlaps the training population.
- **LSUN Bedroom** (Yu et al., 2015). We use the repository's derived one-million-image JPEG subset for unconditional generation, with short side 256 pixels and a deterministic 256×256 center crop. Monitoring uses a deterministic 10,000-image subset selected by hashing image paths with seed 12345; this subset overlaps the training population.

A.2 EVALUATION PROTOCOL

We evaluate the tradeoff between generation quality and sampling throughput through the quality–throughput Pareto frontier. We report FID on all datasets and additionally report sFID, precision, recall, and Inception Score on FFHQ and LSUN Bedroom.

A.2.1 QUALITY METRICS

FID and spatial FID. For real and generated feature distributions with empirical means μ_r, μ_g and covariances Σ_r, Σ_g, the Fréchet distance is

$$\mathcal{D}(\mu_r, \Sigma_r; \mu_g, \Sigma_g) = \|\mu_r - \mu_g\|_2^2 + \mathrm{Tr}\left(\Sigma_r + \Sigma_g - 2\left(\Sigma_r^{1/2}\Sigma_g\Sigma_r^{1/2}\right)^{1/2}\right).$$

FID applies this distance to globally pooled Inception features, whereas sFID uses intermediate spatial features. Lower values indicate better agreement with the real distribution.

Precision and recall. Precision measures sample fidelity, while recall measures coverage of the real distribution. Let $\mathcal{F}_r$ and $\mathcal{F}_g$ denote the real and generated feature sets. We estimate the support of a feature set $\mathcal{F}$ as

$$\mathcal{M}_k(\mathcal{F}) = \bigcup_{z \in \mathcal{F}} \{u : \|u - z\|_2 \le r_k(z; \mathcal{F})\},$$

where $r_k(z; \mathcal{F})$ is the distance to the kth nearest neighbor of z in $\mathcal{F} \setminus \{z\}$. Then

$$\text{Precision} = \frac{1}{|\mathcal{F}_g|} \sum_{z \in \mathcal{F}_g} \mathbf{1}[z \in \mathcal{M}_k(\mathcal{F}_r)],$$

$$\text{Recall} = \frac{1}{|\mathcal{F}_r|} \sum_{z \in \mathcal{F}_r} \mathbf{1}[z \in \mathcal{M}_k(\mathcal{F}_g)].$$

We use $k = 3$, following the OpenAI ADM evaluator. Higher values are preferred for both metrics.

Inception Score. IS measures the confidence and marginal diversity of Inception class predictions:

$$\mathrm{IS} = \exp\big(\mathbb{E}_{x \sim p_g}\left[D_{\mathrm{KL}}(p(y \mid x) \,\|\, p_g(y))\right]\big), \qquad p_g(y) = \mathbb{E}_{x \sim p_g}[p(y \mid x)],$$

where p_g is the generated image distribution and $p(y \mid x)$ is the predicted class distribution. We report the arithmetic mean of IS over ten nonoverlapping splits of 5,000 generated images, estimating the class marginal separately within each split. Higher values are preferred.

A.2.2 DATASET-SPECIFIC EVALUATION

Each reported score is associated with its evaluation backend, reference data, and generated sample count.

CIFAR-10 and ImageNet. We compute FID using Clean-FID in clean mode (Parmar et al., 2022), with reference statistics from the CIFAR-10 training split and ImageNet validation split, respectively. Training-time monitoring uses 5,000 generated images per evaluation, and ImageNet checkpoint evaluation uses 50,000.

FFHQ. We compute FID using the NVLabs EDM implementation and its released FFHQ reference statistics (Karras et al., 2022). For each checkpoint, we report the mean FID over three independent runs of 50,000 samples, using seed ranges 0–49,999, 50,000–99,999, and 100,000–149,999, and retain the individual scores.

Additional metrics use the OpenAI ADM evaluator (Dhariwal & Nichol, 2021) with 50,000 generated samples (seeds 0–49,999) and custom reference statistics built from FFHQ IDs 00000–49999. The real-image pool for precision and recall contains IDs 00000–09999 (10,000 images). Scores from this custom ADM protocol are reported separately from NVLabs FID.

LSUN Bedroom. We use the OpenAI ADM evaluator with the official Bedroom-256 reference and 50,000 generated images, using seeds 2,100,000–2,149,999. The real-image pool for precision and recall contains 5,000 images.

Image conversion. For image values $x \in [-1, 1]$, the NVLabs and ADM protocols convert samples to 8-bit images as

$$Q_{\mathrm{NVLabs}}(x) = \mathrm{uint8}(\mathrm{clip}(127.5x + 128, 0, 255)),$$
$$Q_{\mathrm{ADM}}(x) = \mathrm{uint8}(\mathrm{clip}(127.5(x + 1), 0, 255)).$$

Both conversions operate elementwise, with the integer cast truncating fractional values. The two FFHQ protocols therefore differ in both image quantization and reference statistics.

A.2.3 SAMPLING AND RUNTIME MEASUREMENT

U-Net sampling. We use the deterministic EDM Heun solver (Karras et al., 2022) with 18 steps (35 denoiser evaluations) for CIFAR-10 and ImageNet, and 40 steps (79 evaluations) for FFHQ and LSUN Bedroom. All configurations use $\sigma_{\min} = 0.002$, $\sigma_{\max} = 80$, $\rho = 7$, and zero stochastic churn. Models evaluated under the same dataset protocol share sampling schedules and sample seeds.

FFHQ denoiser inference uses FP32; both teacher and students in the reported LSUN PFI comparison use mixed FP16. Earlier LSUN evaluations with FP32 students are treated as a separate protocol. Both FFHQ and LSUN samplers maintain FP64 state. LSUN denoised predictions are clipped to $[-1, 1]$ at every predictor and corrector evaluation; FFHQ denoised predictions are not clipped.

DiT sampling. The DDPM-trained DiTs (ImageNet, FFHQ, LSUN Bedroom) use the DDPM sampler of Peebles & Xie (2023): ϵ-prediction, learned log-variance interpolated between $\log \beta_t$ and $\log \tilde{\beta}_t$, unclipped denoised predictions, no noise at the final step. The 1000-step linear-β process is respaced to 250 evenly strided timesteps (ImageNet, FFHQ) or run at all 1000 steps (LSUN Bedroom). ImageNet is class-conditional with classifier-free guidance at scale 1.5; FFHQ and LSUN Bedroom are unconditional (scale 1.0, one pass). CIFAR-10 uses the deterministic EDM Heun solver (Karras et al., 2022) with 18 steps (35 denoiser evaluations), $\sigma_{\min} = 0.002$, $\sigma_{\max} = 80$, $\rho = 7$, zero churn, guidance scale 1.25 on the denoised output (two network passes per evaluation), and final samples clipped to $[-1, 1]$. Models evaluated under the same dataset protocol share sampling schedules and the sampling seeds.

Latent-space denoisers run under bfloat16 autocast with sampler state and VAE decode in FP32; CIFAR-10 runs entirely in FP32. Throughput is measured at the same precision. A phase-split student is sampled as one routed network: at each denoiser evaluation the current timestep (or σ) is mapped to its bin of the 20-bin noise partition used in training, and the specialist D_{ϕ_b} whose phase B_b contains it makes the prediction. Sampling thus costs one denoiser evaluation per step regardless of K, as for the teacher D_Θ and the global student.

Throughput measurement. Sampling throughput is measured in images per second, with hardware, GPU count, batch size, and inference precision recorded for each comparison. Sampling is timed with CUDA synchronization, excluding warm-up, image writing, and metric computation. For distributed sampling, throughput is the total number of newly generated images divided by the maximum accumulated sampling time across ranks. Reused samples are excluded.

The FFHQ and LSUN PFI comparisons use four and two NVIDIA RTX PRO 6000 Blackwell Max-Q Workstation Edition GPUs, respectively, with a batch size of 32 per GPU. FFHQ throughput is averaged over two fresh sampling runs for the teacher and three for each student. LSUN throughput is measured using one fresh 50,000-image run per model. Training throughput, where reported, is measured separately in optimizer steps per second.

A.3 Denoiser architectures

We evaluate two denoiser families—diffusion U-Nets and diffusion transformers (DiTs)—with variable-width students specialized to different noise intervals. Section 3.4 describes budget construction and matching; Section A.4 provides checkpoint and implementation details.

U-Net teachers. The U-Net teachers operate in pixel space. For CIFAR-10 and FFHQ, we use the DDPM++ architecture of Song et al. (2021); for ImageNet, we use the ADM U-Net of Dhariwal & Nichol (2021). These checkpoints come from the original 2022 EDM release (Karras et al., 2022). For LSUN Bedroom, we use the ADM-derived diffusion teacher released with Consistency Models (Song et al., 2023). The CIFAR-10 and ImageNet teachers are class-conditional; the FFHQ and LSUN Bedroom teachers are unconditional.

U-Net students. The U-Net students use a variable-width DDPM++-style architecture with dataset-appropriate image resolution and conditioning. Allocation changes channel widths while preserving the configured encoder–decoder structure: blockwise allocation scales widths for each noise interval, whereas layerwise allocation also adjusts widths across structural modules within each student. The ImageNet and LSUN Bedroom U-Net experiments therefore distill ADM-derived teachers into DDPM++-style students.

DiT teachers. The DiT teachers follow Peebles & Xie (2023), using a patch-embedding stem with patch size 2, adaLN-Zero transformer blocks, and an adaLN output layer. The blocks are conditioned on the sum of timestep and class embeddings; unconditional models use a single constant class token. The CIFAR-10 teacher operates in pixel space; the other DiT teachers operate on $32 \times 32 \times 4$ latents of the Stable Diffusion VAE (Rombach et al., 2021). For ImageNet, we use the released class-conditional DiT-XL/2 teacher (hidden width 1152, 28 transformer blocks, 16 attention heads). For LSUN Bedroom and FFHQ, we train unconditional DiT-B/2 teachers from scratch (hidden width 768, 12 transformer blocks, 12 attention heads). For CIFAR-10, we train a class-conditional DiT-S/2-style teacher from scratch (hidden width 384, 12 transformer blocks, 6 attention heads).

DiT students. The DiT students retain their teacher's depth, patch size, attention head count, and output parameterization. Their allocatable dimensions are the residual-stream width, the attention inner width of the query, key, and value projections, and the MLP hidden width. Embedding dimensions adapt to the student's residual width. Blockwise allocation selects one width configuration per noise interval, shared across that student's transformer blocks. Layerwise allocation additionally varies attention and MLP widths across transformer blocks; the residual width remains shared throughout each student.

A.4 Implementation details

Teacher parameter counts. The DiT-XL/2, DiT-B/2, and CIFAR-10 DiT-S/2-style teachers have approximately 675M, 130M, and 32.5M parameters, respectively.

DiT teacher training. The released ImageNet DiT-XL/2 teacher was trained with a 1000-step DDPM objective (Peebles & Xie, 2023). We pretrain the LSUN Bedroom and FFHQ DiT-B/2 teachers with the same 1000-step DDPM objective until convergence. For CIFAR-10, we use

EDM preconditioning and loss (Karras et al., 2022), non-leaky augmentation with $p = 0.12$ and augmentation-label conditioning, and dropout 0.13.

Student implementation and width rounding. For DiT students with H attention heads, residual and attention inner widths are rounded to multiples of $\mathrm{lcm}(H, 8)$, and MLP hidden widths to multiples of 8. After rounding, DiT parameter counts are recomputed exactly and checked against instantiated models. The combined DiT student parameter count targets the teacher's parameter budget; discrete width choices can leave a difference between the realized count and the target.

A.5 Specialist routing and training

The router assigns each denoising call to one phase specialist using the nearest profiled noise level. Let $\mathcal{S}$ denote the profiled levels and let $\tau(s)$ map a level to its bin in the phase partition. Writing each phase B_b as an interval of bin indices, the routing rule is

$$s^\star(u) \in \arg\min_{s\in\mathcal{S}} |u - s|, \qquad b(u) = b \quad \text{such that} \quad \tau(s^\star(u)) \in B_b.$$

The selected specialist $D_{\phi_{b(u)}}$ evaluates the input at the queried noise level u. All specialists remain stored in memory, but each call executes only the selected specialist.

Profiling and capacity allocation determine the specialist architectures independently of their training objective. A default instantiation trains data-predicting specialists to match both the frozen teacher's output and the clean target:

$$\begin{aligned}\mathcal{L}_{\mathrm{hyb}} = \mathbb{E}_{(i,s)}\big[\lambda(s)\big(\beta_{\mathrm{KD}}\|D_{\phi_{b(s)}}(z_{i,s}, s, c_i) - \mathrm{sg}[D_\Theta(z_{i,s}, s, c_i)]\|_2^2 \\ + \beta_{\mathrm{data}}\|D_{\phi_{b(s)}}(z_{i,s}, s, c_i) - x_i\|_2^2\big)\big].\end{aligned} \tag{2}$$

Here, x_i is the clean target, $z_{i,s}$ is its noisy input at level s, and c_i denotes the conditioning inputs. The expectation averages over sampled training examples and noise levels, including the noise used to construct $z_{i,s}$. The operator sg stops gradients through the teacher output, $\lambda(s)$ weights noise levels, and β_{KD} and β_{data} balance teacher imitation and data reconstruction. Other compatible distillation objectives can be used without changing the profiling and allocation stages.

B Audio experiment

To show that PACE applies to additional modalities, we use audio as a profiling-only transfer scenario for PACE. The goal is to ask whether a waveform diffusion model exhibits distinct timestep regimes and, under a fixed three-specialist design, how the residual-model capacity would be divided among them. We do not train audio students in this experiment, so the result is a candidate partition and relative capacity demand rather than an audio-quality comparison.

B.1 Experiment

We profile the unconditional one-million-update DiffWave teacher (Kong et al., 2020) on the ten spoken-digit subset (SC09) of Speech Commands v0.02 (Warden, 2018). The model works with one-second, 16-kHz waveforms and has 36 residual blocks, 256 residual/skip and 200 diffusion timesteps, selected from a fixed pool of 100 validation At each native time step we assess four examples which have been corrupted. For each Conv1d output filter, we exchange that activation channel between paired examples at the same timestep and measure the increase in epsilon-prediction. The positive PFI scores are then put into 20 bins, each containing ten timesteps, Ordered from the highest noise ($t = 199$) to the lowest noise ($t = 0$).

The profile contains $37,377$ filters. Of these, 513 belong to the fixed input/output stem and head, while the remaining $36,864$ filters belong to the residual stack, which defines the candidate allocation scope. Because the single output-head filter dominates the all-filter PFI profile, the phase correlations and capacity demand are computed from this residual-only perspective. correlations and capacity demand are derived from this residual-only view. The resulting timestep-bin correlation matrix and normalized demand vector are the two artifacts carried forward below.

A 3-specialists scenario. The optimal partitioning results with $K = 3$ to represent three routed audio specialists and optimize only the two contiguous phase boundaries using Pearson similarity between normalized positive residual-filter PFI fingerprints. This gives $t = 199$–70, $t = 69$–30, and $t = 29$–0, containing 13, 4, and 3 bins, respectively.

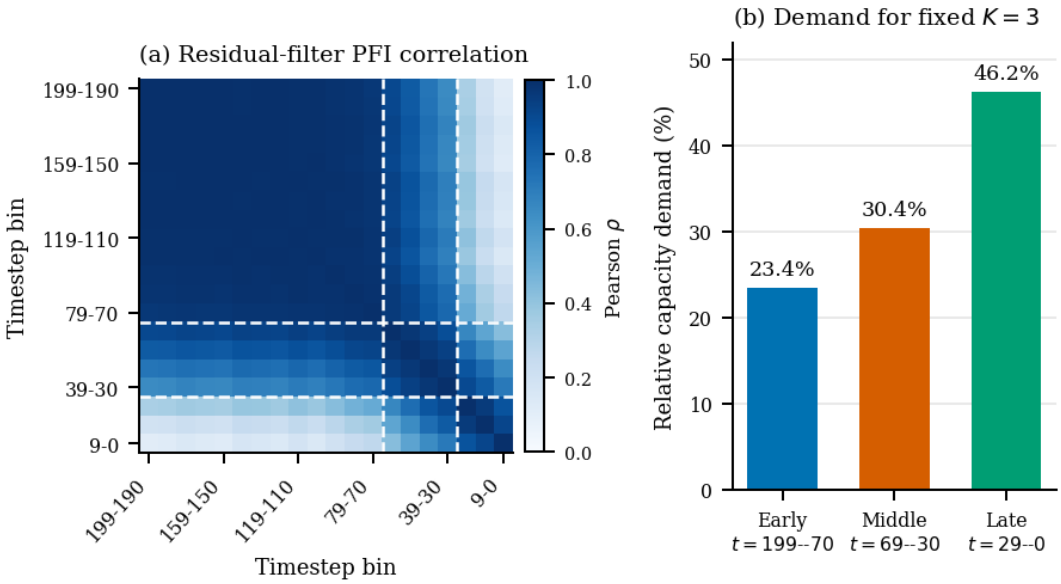

Figure 4: $K = 3$ **audio profiling result. (a)** Pearson correlation matrix between the 20 positive relative-PFI fingerprints; dashed lines mark the optimized phase boundaries. **(b)** Relative capacity demand d_b for each specialist.

Figure 4(a) shows high within-phase similarity (0.992, 0.943, and 0.912 for the early, middle, and late phases) and a much lower early–late similarity of 0.268. The middle phase remains related to both sides, with correlations of 0.815 to the early phase and 0.693 to the late phase, which is consistent with a transition between high- and low-noise denoising regimes.

The corresponding relative capacity demands are 23.4%, 30.4%, and 46.2% for the early, middle, and late specialists (Figure 4(b)). In this scenario, the final low-noise phase receives the largest share even though it contains only three bins, while the thirteen-bin high-noise phase receives the smallest share. This is the main audio result: the residual sensitivity pattern changes across the reverse process, and the three-specialists profile concentrates capacity demand in the late phase.

C Sensitivity correlations and discovered phases

We examine the temporal sensitivity structure of each model–dataset pair with a complete saved profile. All matrices contain 20 bins in denoising order, from high to low noise. Bin indices are zero-based; $[a, b)$ includes bins $a, \ldots, b-1$. Dashed lines show the optimized boundaries, and the colored strips identify phases $P_1, \ldots, P_K$. Every panel uses the paper's white-to-blue Pearson-correlation scale from 0 to 1, with a vertical colorbar. Negative DiT-Micro correlations appear in light gray, indicated by the colorbar extension below zero.

C.1 Image U-Nets

Figure 5 compares the relative filter-sensitivity fingerprints. Phase counts are selected automatically with the cross-phase-penalty objective and $\lambda_{\text{sep}} = 0.02$, except for FFHQ, where $K = 3$ is chosen manually. Phase boundaries are optimized in all cases.

DDPM++ on CIFAR-10. The two phases cover $[0, 8)$ and $[8, 20)$. Within-phase correlations are 0.912 and 0.923, compared with 0.363 between phases.

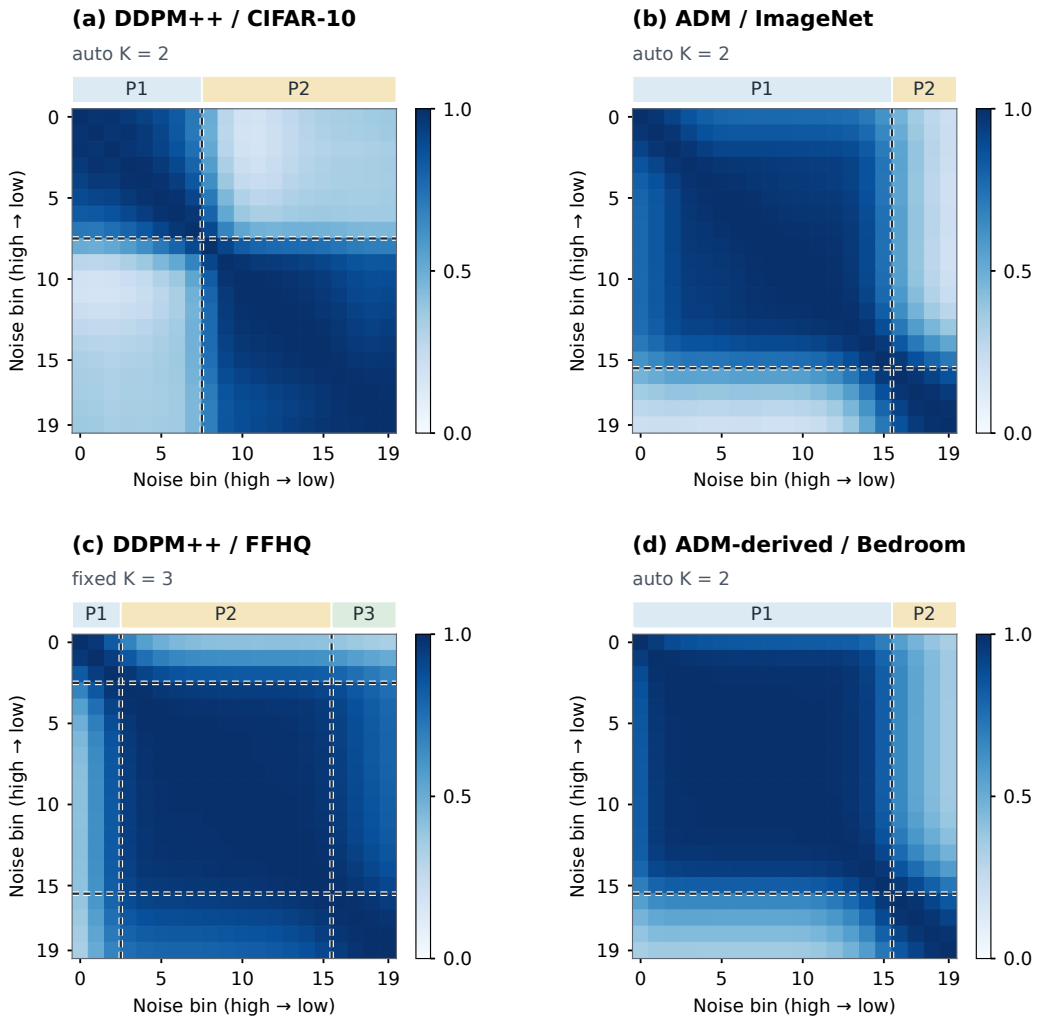

Figure 5: Sensitivity correlations and optimized phases for image U-Nets. Each cell compares the filter-sensitivity patterns at two noise bins. Phase boundaries are drawn on both axes. The phase-count selection rule is stated above each panel.

Table 3: U-Net phase intervals and mean within-phase correlations, in denoising order. Means exclude bin self-correlations.

Teacher / dataset	K	Phase bins	Within-phase $\bar{S}$
DDPM++ / CIFAR-10	2	$[0, 8), [8, 20)$	0.912, 0.923
ADM / ImageNet-64	2	$[0, 16), [16, 20)$	0.904, 0.951
DDPM++ / FFHQ-64	3	$[0, 3), [3, 16), [16, 20)$	0.917, 0.984, 0.982
ADM-derived / Bedroom-256	2	$[0, 16), [16, 20)$	0.950, 0.940

ADM on ImageNet. The split occurs later, at bin 16. The phases $[0, 16)$ and $[16, 20)$ have within-phase correlations of 0.904 and 0.951; their cross-phase mean is 0.429. The final four bins form a compact, distinct sensitivity.

DDPM++ on FFHQ. We manually choose $K = 3$ by visually inspecting the PFI correlation matrix and optimize the boundaries using the objective in Section 3.3. The resulting phases $[0, 3)$, $[3, 16)$, and $[16, 20)$ contain 3, 13, and 4 bins, respectively. Within-phase correlations are 0.917, 0.984,

and 0.982. The middle–late correlation is 0.884, compared with 0.554 for early–late. These values suggest some continuity across the phase boundaries.

ADM-derived teacher on LSUN Bedroom-256. The weighted PFI profile selects $[0, 16)$ and $[16, 20)$, with within-phase means of 0.950 and 0.940 and a cross-phase mean of 0.547. This profile samples 4,387 filters representing a population of 119,555, so retaining the expansion weights is necessary to reproduce the partition.

C.2 DiT-Micro on CIFAR-10

This analysis uses an eight-block DiT-Micro teacher with hidden width 192 and three attention heads per block. It is a separate model from the larger CIFAR-10 DiT-S/2-style teacher described in the experimental setup. The profile permutes each of the 24 attention-head activations before the output projection, using 1,000 images and 64 EDM noise levels aggregated into 20 bins. This archived permutation protocol predates the image U-Net batch-local exact-noise-level PFI protocol.

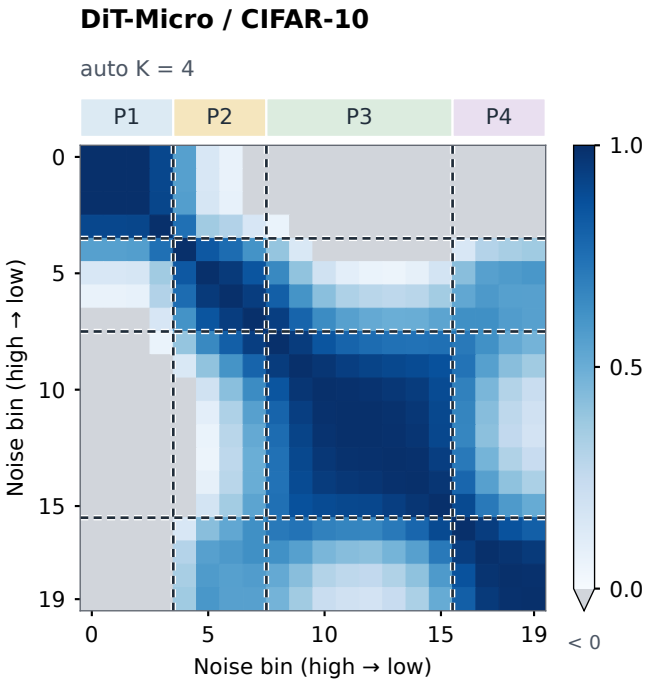

Figure 6: Archived DiT-Micro/CIFAR-10 analysis. Pearson correlations between relative attention-head sensitivity fingerprints, with the automatically selected four-phase partition. Light-gray cells denote negative correlations.

The cross-phase-penalty objective with $\lambda_{\text{sep}} = 0.02$ selects four phases: $[0, 4)$, $[4, 8)$, $[8, 16)$, and $[16, 20)$. Their within-phase correlations are 0.956, 0.832, 0.914, and 0.928. The early phase has negative mean correlations with the third and fourth phases (-0.146 and -0.189), showing a change in the relative pattern of head sensitivity across the denoising trajectory. The second phase is less internally homogeneous and remains positively correlated with each of the other phases. These observations describe the archived profile; they do not establish phase stability across independent profiling runs or the partition of a different DiT teacher.

D QUALITATIVE RESULTS

D.1 CONVNET

We compare the convolutional teacher with the Global, Uniform blockwise, Phase-aware blockwise, and Phase-aware layerwise students on CIFAR-10 (Figure 7), ImageNet (Figure 8), FFHQ (Figure 9), and LSUN Bedroom-256 (Figure 10). Each figure shows ten samples per model, with rows corresponding to fixed random seeds and columns to model variants. Within each row, models share the same initial noise and, where applicable, conditioning class, using the same dataset-specific sampling settings. This arrangement enables direct comparison of composition, structure, and local detail across allocation strategies. All seeds were fixed before generation, and samples were included without visual quality screening.

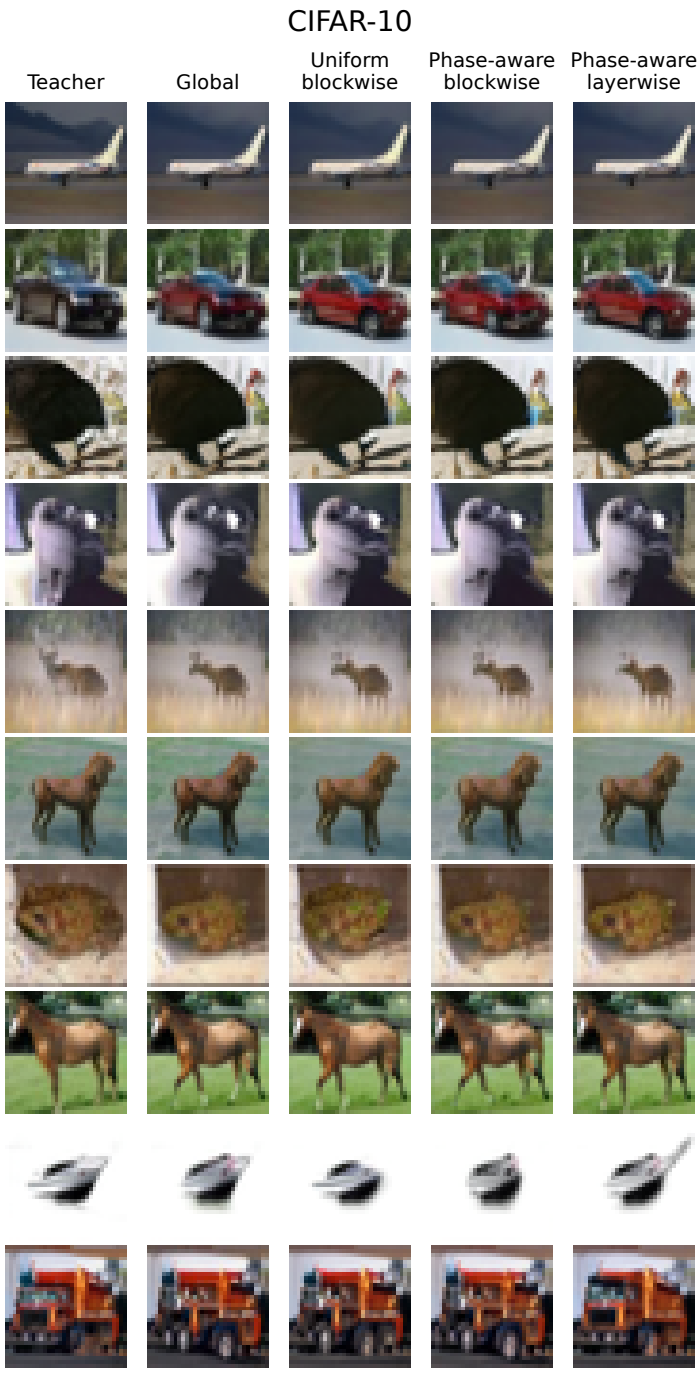

Figure 7: CIFAR-10: ten predetermined samples per model, generated afresh from EMA weights. Each row shares the random seed and conditioning class across models. Columns show Teacher, Global, Uniform blockwise, Phase-aware blockwise, and Phase-aware layerwise. EDM Heun uses 18 steps (35 denoiser evaluations), FP32 inference. Examples were fixed before generation and were not screened for visual quality.

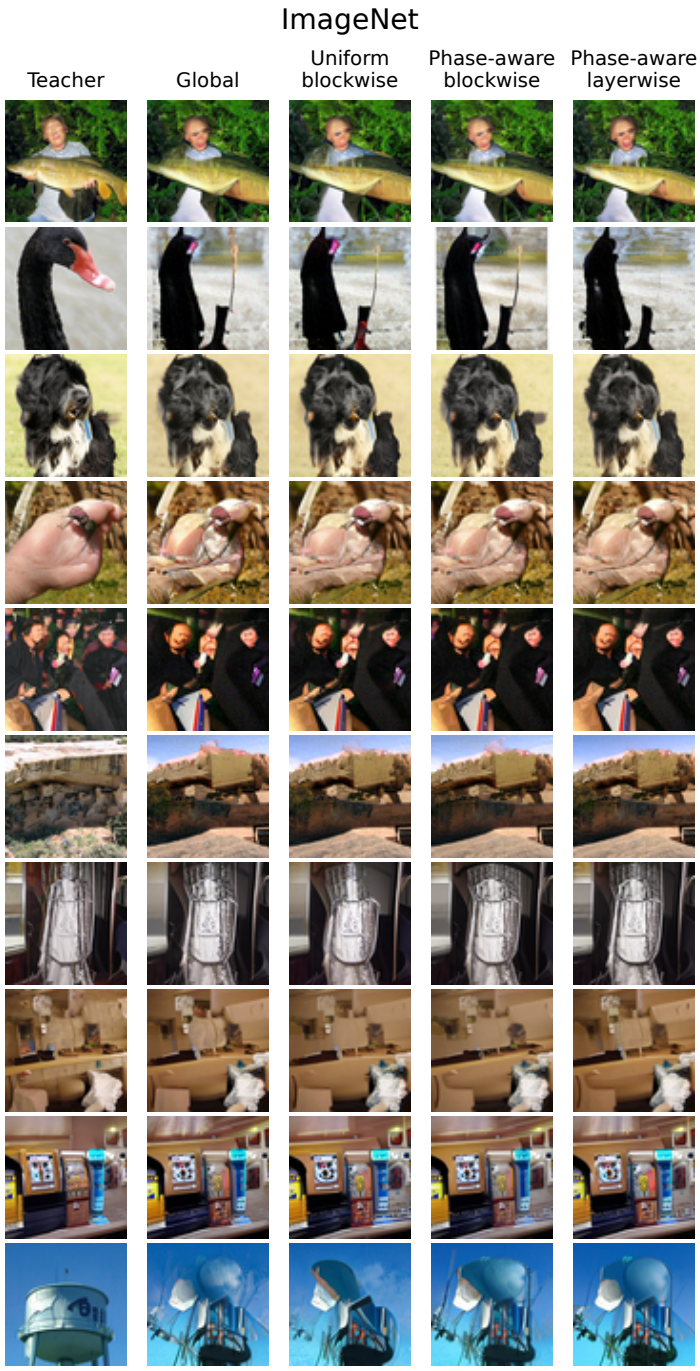

Figure 8: ImageNet: ten predetermined samples per model, generated afresh from EMA weights. Each row shares the random seed and conditioning class across models. Columns show Teacher, Global, Uniform blockwise, Phase-aware blockwise, and Phase-aware layerwise. EDM Heun uses 18 steps (35 denoiser evaluations), FP32 inference. Examples were fixed before generation and were not screened for visual quality.



Figure 9: FFHQ: ten predetermined samples per model, generated afresh from EMA weights. Each row shares the random seed across models. Columns show Teacher, Global, Uniform blockwise, Phase-aware blockwise, and Phase-aware layerwise. Students use permutation feature importance (PFI) profiling. EDM Heun uses 40 steps (79 denoiser evaluations), FP32 inference. Examples were fixed before generation and were not screened for visual quality.

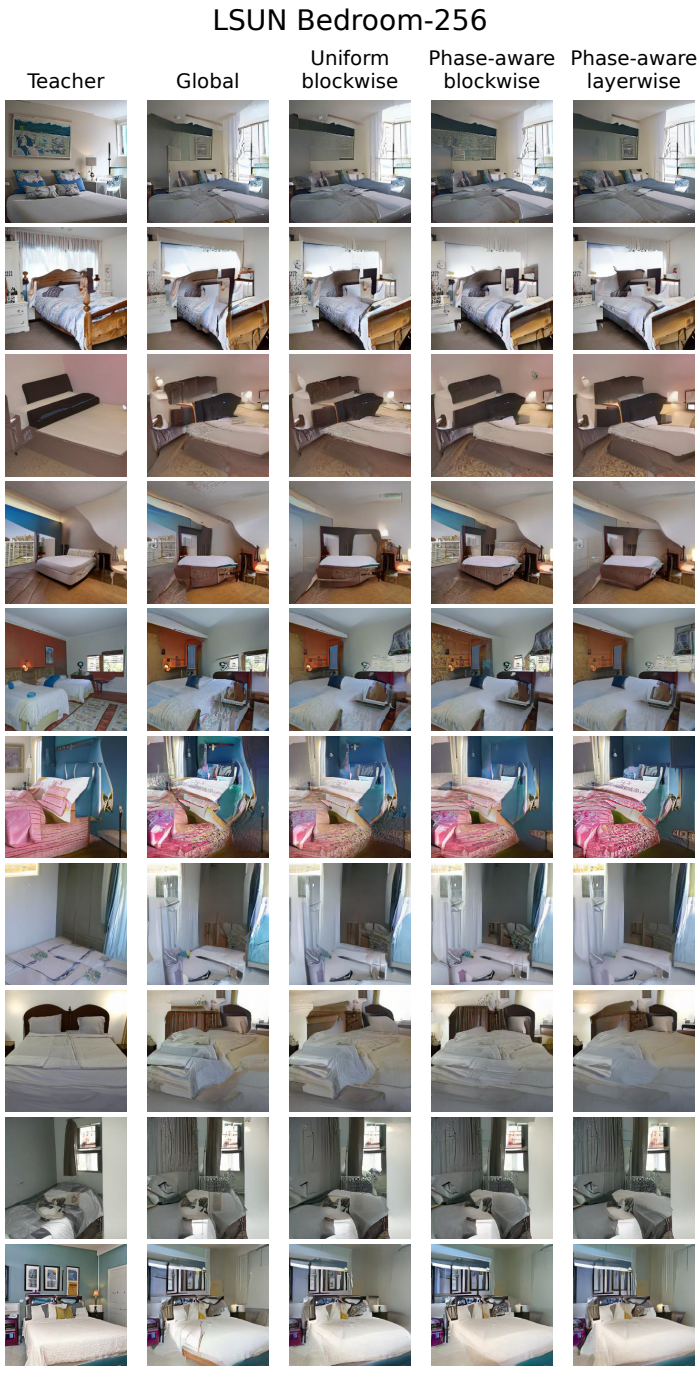

Figure 10: LSUN Bedroom-256: ten predetermined samples per model, generated afresh from EMA weights. Each row shares the random seed across models. Columns show Teacher, Global, Uniform blockwise, Phase-aware blockwise, and Phase-aware layerwise. Students use permutation feature importance (PFI) profiling. EDM Heun uses 40 steps (79 denoiser evaluations), mixed FP16 inference with denoised clipping. Examples were fixed before generation and were not screened for visual quality.

D.2 DiT

Figures 11–14 show ten visually selected examples per model and dataset.

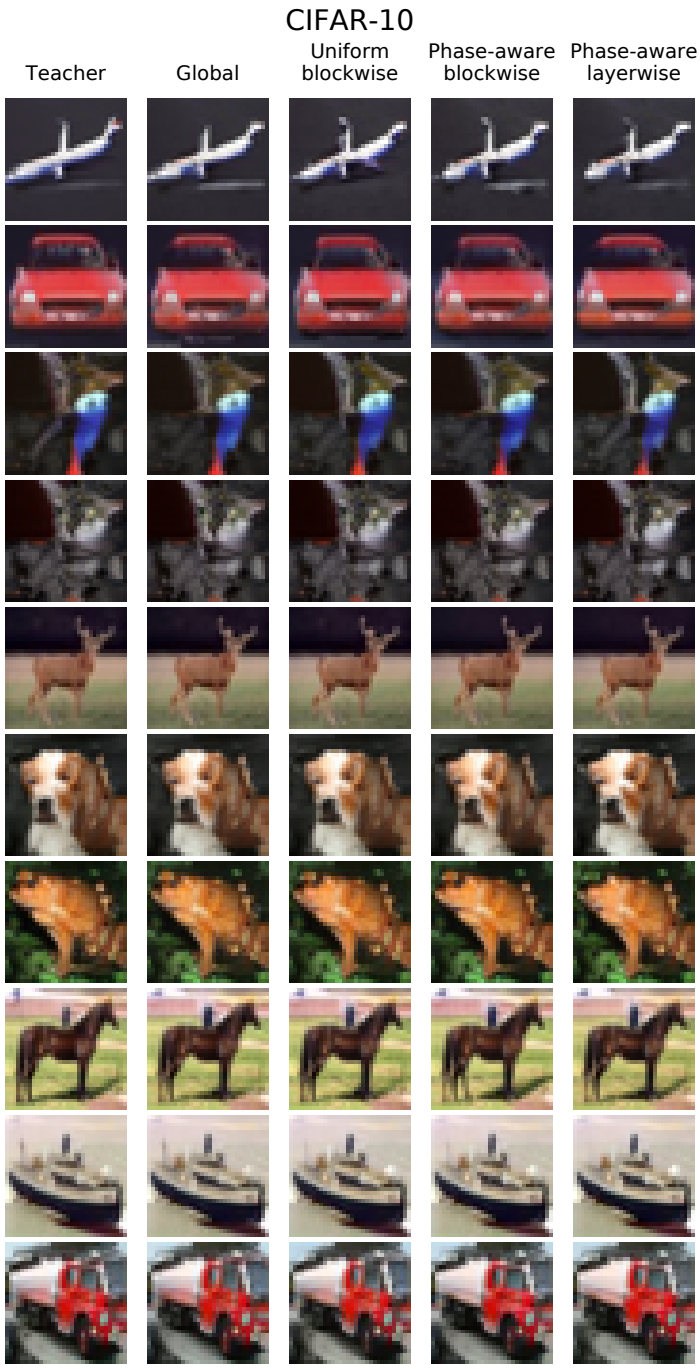

Figure 11: CIFAR-10: ten selected samples per model. Each row shares the random seed and conditioning class across models. Columns show Teacher, Global, Uniform blockwise, Phase-aware blockwise, and Phase-aware layerwise. Sampling uses 18 steps, guidance scale 1.25, and FP32 inference.



Figure 12: ImageNet: ten selected samples per model. Each row shares the random seed and conditioning class across models. Columns show Teacher, Global, Uniform blockwise, Phase-aware blockwise, and Phase-aware layerwise. DDPM uses 250 steps and guidance scale 1.5, with BF16 denoising.

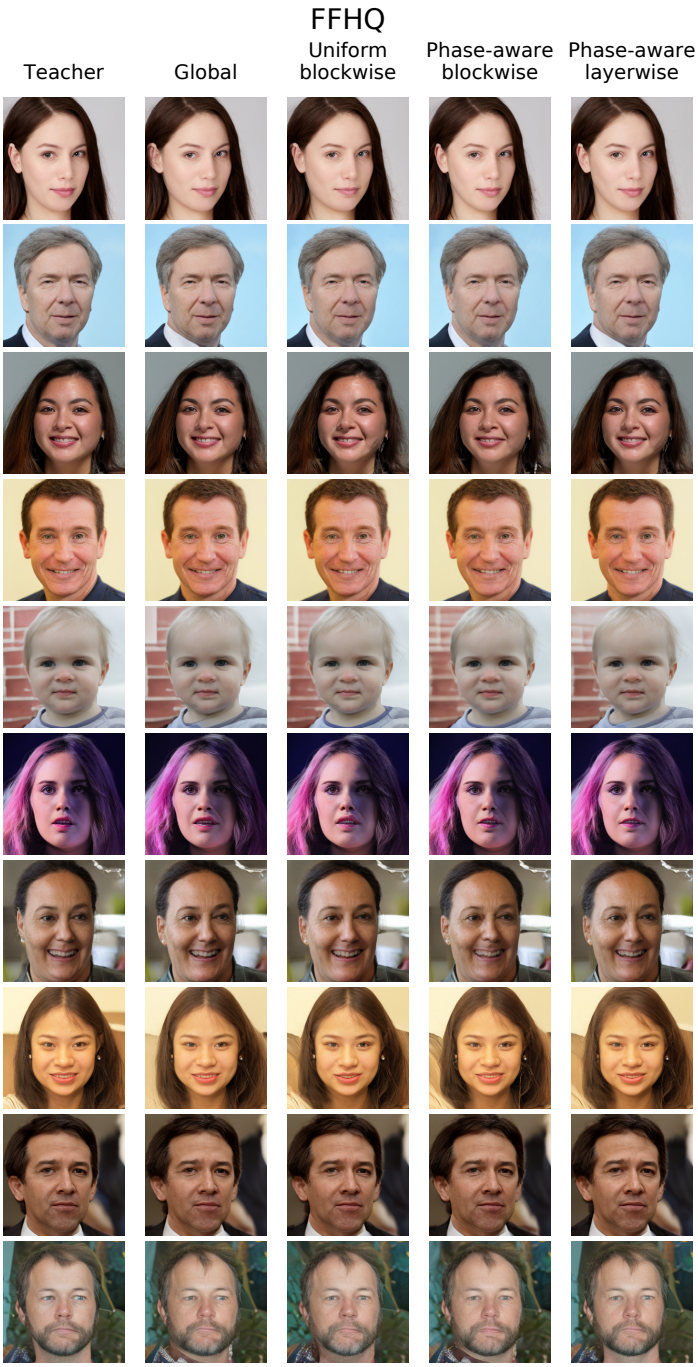

Figure 13: FFHQ: ten selected samples per model. Each row shares the random seed across models. Columns show Teacher, Global, Uniform blockwise, Phase-aware blockwise, and Phase-aware layerwise. DDPM uses 250 steps and guidance scale 1.0, with BF16 denoising.

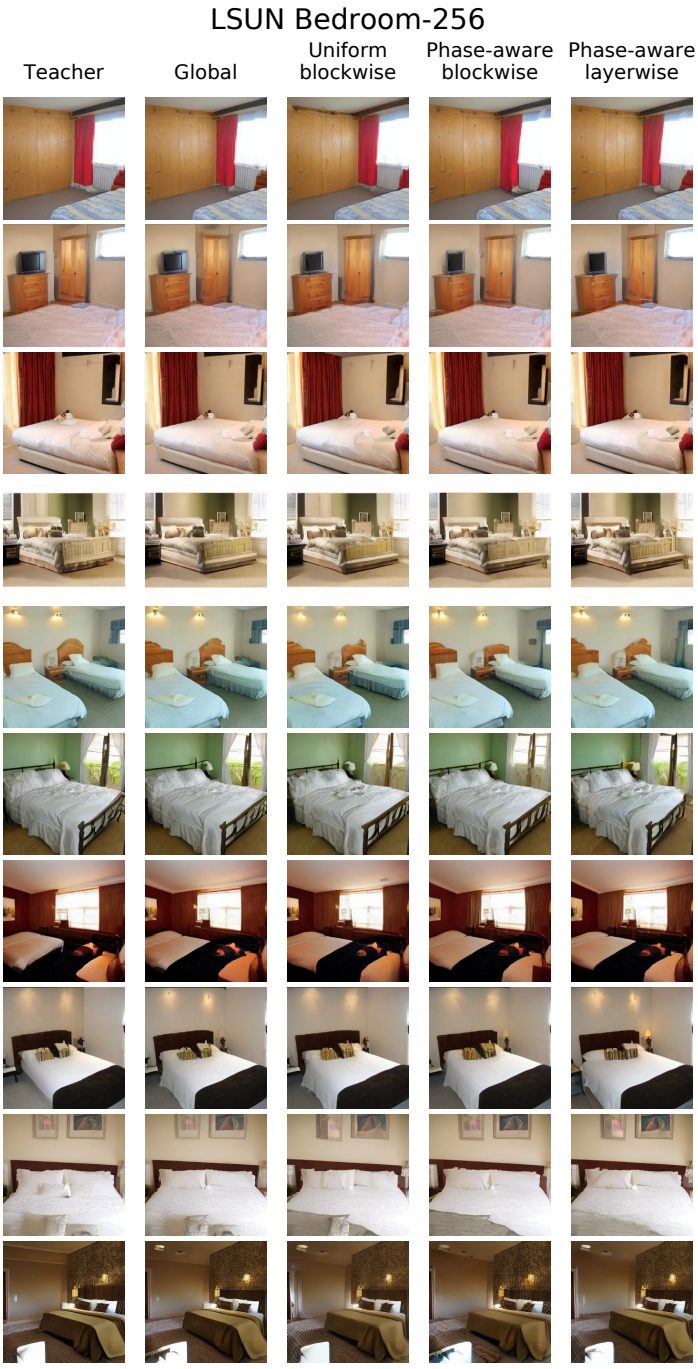

Figure 14: LSUN Bedroom-256: ten selected samples per model. Each row shares the random seed across models. Columns show Teacher, Global, Uniform blockwise, Phase-aware blockwise, and Phase-aware layerwise. DDPM uses 1000 steps and guidance scale 1.0, with BF16 denoising.