PACE: PHASE-DISPERSION-AWARE CACHE SCHEDULING FOR ACCELERATING DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Feature caching accelerates Diffusion Transformers (DiTs) by reusing or forecasting internal features across denoising steps, but the placement of the few steps that are still fully computed, the cache schedule, is usually set by fixed intervals or by generic feature distances. Both criteria are blind to *what* is changing in the image and can skip the brief window in which global structure first forms; the resulting errors are never repaired by later steps. We identify this window spectrally: phase-swap and perturbation analyses show that cache-induced structural corruption is carried by the Fourier phase, not the amplitude, of the model's predicted clean signal. Building on this, we propose PACE, a scheduler that calibrates, once per backbone and without training, a phase-dispersion profile measuring how fast the in-band spectral coefficients bend along the trajectory, and places computation anchors at equal quantiles of its cumulative mass. This equal-mass rule is the minimax- and convex-cost-optimal partition of the calibrated profile, is prompt-invariant after normalization, and plugs into reuse- and prediction-based caches with an anchor lookup and no runtime cost. Under identical numbers of function evaluations, PACE improves fidelity to full sampling on FLUX.1-dev, SiT-XL/2, Wan2.1-1.3B, and HunyuanVideo-13B; on video generation it raises PSNR over the strongest baseline, SeaCache, by 6.5–13.6 dB while matching its latency, and it is the only schedule that recovers the motion dynamics of full sampling.

1 INTRODUCTION

Diffusion Transformers (DiTs) (Peebles & Xie, 2023) power high-fidelity image and video generation (Black Forest Labs, 2024; Kong et al., 2024), but their iterative denoising requires many sequential forward passes through a large Transformer denoiser, hindering latency-sensitive deployment. Existing acceleration methods either reduce the number of sampling steps (Lu et al., 2022; Zhao et al., 2023; Luo et al., 2023; Yin et al., 2024) or lower the per-step cost (Li et al., 2023; Dao et al., 2022; Bolya & Hoffman, 2023). Feature caching follows the latter direction: it exploits the temporal redundancy of hidden states to replace selected full computations with reused or forecasted features (Ma et al., 2024b; Liu et al., 2025c), and is training-free.

Despite this progress, deciding when to refresh the cache remains an open problem. Existing schedulers either refresh at fixed intervals (Liu et al., 2025c; Feng et al., 2026), ignoring the highly non-uniform dynamics of the denoising trajectory, or allocate computation from generic estimates of feature change, obtained online or searched offline (Adnan et al., 2025; Bu et al., 2025; Chung et al., 2026; Cui et al., 2026; Shao et al., 2025). Both criteria are agnostic to the structural content of the image and can overlook the short window in which global structure first forms. As shown in Figure 1, one long cache jump of SeaCache (Chung et al., 2026) skips the steps where the sign text first becomes determined; the error introduced there propagates through the remaining denoising steps, which cannot repair it. Under a fixed compute budget, the central scheduling question is therefore not how much feature discrepancy can be tolerated, but where full computation must be spent to preserve emerging structure. To identify what carries the emerging image structure, we perform a phase-swap experiment (Figure 2): decomposing the generated images into Fourier amplitude and phase, swapping the correct phase into a corrupted output restores the sign text, whereas swapping

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.

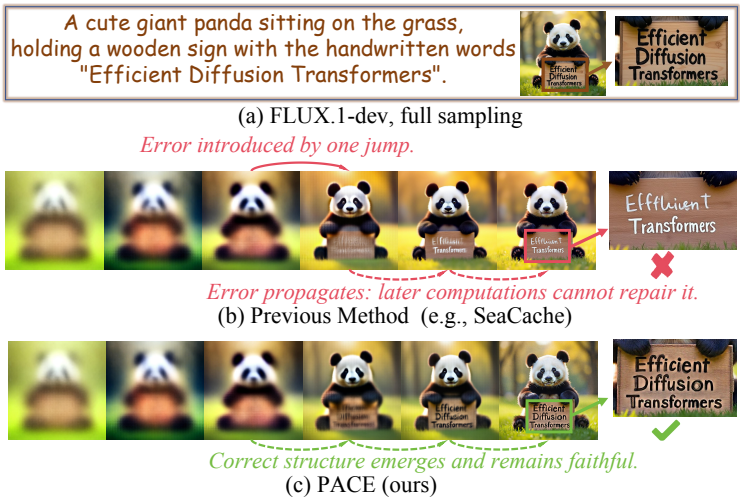

Figure 1: A motivating failure case. Both schedulers accelerate 50-step FLUX.1-dev with eight full computations ($\approx 4\times$ speedup). (a) Full sampling forms the sign text correctly. (b) SeaCache jumps across the early structure-formation window; the text error introduced there (solid red) propagates irreparably (dashed red). (c) PACE concentrates anchors on this window and preserves the text (green).

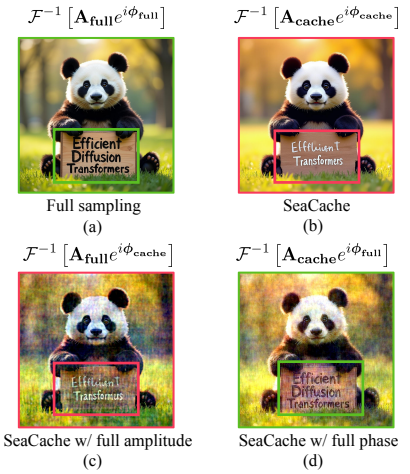

Figure 2: Phase-swap analysis on the motivating scene. Outputs of (a) full sampling and (b) SeaCache are written as $\mathcal{F}^{-1}[\mathbf{A}e^{i\phi}]$. (c) The amplitude of (a) with the phase of (b) leaves the text corrupted; (d) the reverse swap restores it.

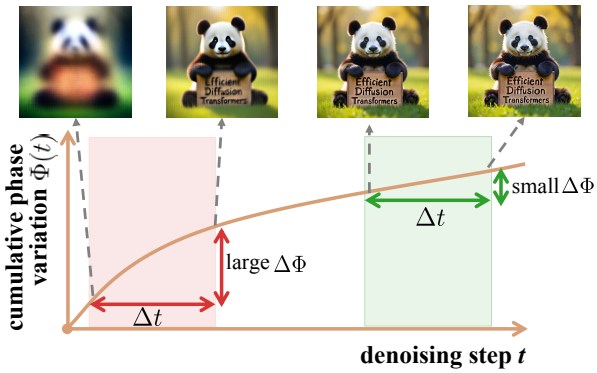

Figure 3: Cumulative phase variation $\Phi(t)$ of the predicted clean signal along a full FLUX.1-dev trajectory. Structure emerges where phase changes most rapidly, so equal-length cache intervals Δt carry unequal risk: the early one spans a large $\Delta\Phi$ and causes the corruption of Figure 2; the late one is nearly harmless.

the correct amplitude does not. Together with a first-order perturbation analysis, this indicates that structural corruption is primarily associated with Fourier phase rather than amplitude.

This observation yields a concrete scheduling criterion: a cache interval is harmful precisely when the phase spectrum evolves rapidly within it. As Figure 3 shows, this variation concentrates in a short structure-formation window, so full computations should be metered by the per-step rate of phase-sensitive spectral variation, which we term *phase dispersion*. We instantiate this criterion as **PACE**, a **P**hase-Dispersion-**A**ware **C**ache sch**E**duler operating under a fixed NFE budget. PACE consists of two offline stages. In the *calibration* stage, it estimates a phase-dispersion profile that measures the curvature of complex spectral coefficients of predicted clean signals within a fixed structure band; the normalized profile is empirically stable across prompts and is therefore calibrated once per

backbone. In the *scheduling* stage, it places computation anchors at equal intervals of cumulative dispersion mass, which minimizes the total proxy error under an explicit convex interval-cost model. The resulting static schedule requires no training, per-prompt tuning, or online gating. In summary, our main contributions are:

- **A phase-based perspective on cache scheduling.** Through phase-swap experiments and perturbation analysis, we show that cache-induced structural corruption is more directly associated with Fourier phase. Under locally coherent extrapolation residuals, phase errors can accumulate across reuse steps and persist after subsequent refreshes (Appendix B).
- **The PACE framework.** We propose a training-free, plug-and-play cache scheduler that calibrates an empirically stable normalized phase-dispersion profile offline and reallocates computation anchors toward the structure-formation window, integrating with both reuse- and prediction-based caches with identical NFE and zero runtime overhead.
- **Outstanding performance.** Across image generation (FLUX.1-dev, SiT-XL/2) and video generation (Wan2.1-1.3B, HunyuanVideo-13B), PACE consistently outperforms existing caching methods under identical NFE budgets, surpassing the strongest baseline SeaCache by 6.5–13.6 dB PSNR on video generation at the same latency.

2 Related Work

2.1 Generative Model Acceleration

Diffusion inference is accelerated along two main directions. The first reduces the number of sampling steps: high-order solvers such as DPM-Solver (Lu et al., 2022) and UniPC (Zhao et al., 2023) admit larger step sizes, while distillation methods such as LCM (Luo et al., 2023) and DMD (Yin et al., 2024) compress the trajectory into a few steps at the cost of additional training. The second lowers the per-step cost by quantizing the denoiser (Li et al., 2023), optimizing attention kernels (Dao et al., 2022), or pruning redundant tokens (Bolya & Hoffman, 2023), all of which modify the model or its implementation. Feature caching also reduces per-step cost, but is training-free and leaves the model, solver, and sampling procedure untouched; PACE belongs to this family and is complementary to step-reduction methods.

2.2 Caching-based Acceleration

Existing caching methods differ in what is reused and when reuse is triggered.

Reuse-based caches exploit feature similarity at multiple granularities: DeepCache (Ma et al., 2024b) at the block level of U-Net backbones, FORA (Selvaraju et al., 2024) at the module level of DiTs, and ToCa (Zou et al., 2025), ClusCa (Zheng et al., 2025), and FastCache (Liu et al., 2025a) at the token level. For video generation, PAB (Zhao et al., 2025), ProfilingDiT (Ma et al., 2025), and FasterCache (Lv et al., 2025) extend reuse to the temporal dimension.

Predictive caches replace plain reuse with forecasting. TaylorSeer (Liu et al., 2025c) extrapolates the features of skipped steps with a Taylor expansion of cached history, and HiCache (Feng et al., 2026) stabilizes higher-order extrapolation with a Hermite polynomial basis. FoCa (Zheng et al., 2026) recasts caching as solving a feature ODE, and AB-Cache (Yu et al., 2025) derives the extrapolation from Adams–Bashforth integration. CG-Taylor (Guan et al., 2025) and SpeCa (Liu et al., 2025d) further gate or verify the forecasts on the fly, SpecDiff (Pan et al., 2026) refines token selection with speculative future information, while SVD-Cache (Chen et al., 2026) forecasts the principal feature subspace and reuses the residual.

Adaptive schedulers decide online when a refresh is necessary. TeaCache (Liu et al., 2025b) accumulates differences of timestep-modulated inputs against a threshold, AdaCache (Kahatapitiya et al., 2025) adapts the schedule to motion complexity, DiCache (Bu et al., 2025) and RFC (Son et al., 2026) anticipate output changes from shallow-layer probes or input features, and TaoCache (Fan et al., 2025) and SenCache (Haghighi & Alahi, 2026) account for late-stage structure and output sensitivity. SeaCache (Chung et al., 2026) filters features with a spectral-evolution-aware Wiener filter before measuring reuse metrics, thereby injecting a frequency prior into the gating decision.

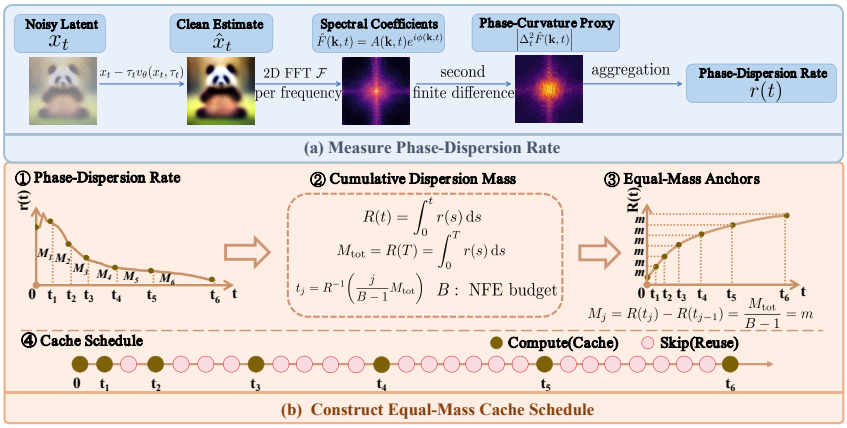

Figure 4: Overview of PACE. (a) Second finite differences of in-band Fourier coefficients of predicted clean signals, normalized by in-band amplitude, yield the phase-dispersion rate $r(t)$. (b) Its cumulative mass $R(t)$ is partitioned into $B-1$ equal intervals, whose boundaries map back to the anchor positions.

Global schedule optimization determines the set of fully computed timesteps offline. CacheQuant (Liu et al., 2025e) and CEM (Shao et al., 2025) minimize accumulated error proxies with dynamic programming, OmniCache (Chu et al., 2025) allocates reuse from a trajectory-level perspective, DPCache (Cui et al., 2026) formulates anchor selection as a path planning problem, and ProCache (Cao et al., 2026) searches non-uniform activation schedules by constrained sampling. ERTACache (Peng et al., 2025) jointly rectifies caching errors and adjusts timesteps.

Unlike these schedulers, which are driven by generic feature distances, PACE schedules from the phase-sensitive spectral curvature of predicted clean signals via equal-mass partitioning and, changing only where full computations are placed, complements both reuse-based and prediction-based caches.

3 Method

3.1 Preliminaries

Feature Caching and Cache Scheduling. We consider a pretrained flow-based denoiser (Lipman et al., 2023; Liu et al., 2023) v_θ evaluated at ordered sampling positions $t \in \{0, \dots, T\}$ along the denoising trajectory, where $\mathbf{x}_t$ denotes the latent at position t and τ_t its flow time, decreasing from one at the noise end to zero at the data end. Feature caching reduces inference cost by fully evaluating the denoiser at only a subset of positions: the anchor set $\mathcal{A} \subseteq \{0, \dots, T\}$ with $|\mathcal{A}| = B$ under an NFE budget of B. At an anchor $t \in \mathcal{A}$, the model performs full computation and stores the intermediate feature $\mathbf{h}_t$. At a non-anchor position, the feature is approximated by

$$\widetilde{\mathbf{h}}_t = \mathcal{C}\left(t; \{(a, \mathbf{h}_a) : a \in \mathcal{A},\ a < t\}\right), \tag{1}$$

where $\mathcal{C}$ denotes the underlying cache operator. Direct-reuse methods (Ma et al., 2024b; Selvaraju et al., 2024) instantiate $\mathcal{C}$ by retrieving a previously cached feature, whereas prediction-based methods (Liu et al., 2025c; Feng et al., 2026) estimate the feature from cached history.

The cache operator determines how a skipped feature is approximated, whereas the anchor set determines when approximation is used. PACE leaves $\mathcal{C}$ unchanged and optimizes only the placement of $\mathcal{A}$ under the fixed budget B, and therefore applies to both direct-reuse and prediction-based caches.

Fourier Amplitude and Phase. To characterize structural variation along the denoising trajectory, we analyze the predicted clean signal, an internal estimate of the final sample that the model provides at every position. For the linear flow parameterization used in our experiments,

$$\hat{\mathbf{x}}_t = \mathbf{x}_t - \tau_t v_\theta(\mathbf{x}_t, \tau_t), \tag{2}$$

which requires no ground-truth sample. Its channel-wise spatial Fourier transform is

$$\hat{F}(\mathbf{k}, t) = \mathcal{F}[\hat{\mathbf{x}}_t](\mathbf{k}) = A(\mathbf{k}, t)e^{i\phi(\mathbf{k},t)}. \tag{3}$$

Here, $\mathbf{k}$ denotes spatial frequency, $A(\mathbf{k}, t) = |\hat{F}(\mathbf{k}, t)|$ is the amplitude, and $\phi(\mathbf{k}, t) = \arg \hat{F}(\mathbf{k}, t)$ is the phase. Amplitude specifies the strength of each frequency component, whereas phase governs their spatial alignment (Oppenheim & Lim, 1981).

3.2 Overview of PACE

As illustrated in Figure 4, PACE calibrates a phase-dispersion profile offline from full-computation trajectories, partitions its cumulative mass into equal intervals to obtain anchors, and at inference invokes the original cache operator outside these anchors.

Phase-Dispersion Rate. The phase-swap analysis in Figure 2 indicates that cache-induced structural corruption is more closely associated with Fourier phase than with amplitude. We first characterize this relation for an individual spectral coefficient. Let $\hat{F} = Ae^{i\phi}$ denote a coefficient of Equation (3) under full computation and let $\widetilde{F} = (A + \delta A)e^{i(\phi+\delta\phi)}$ denote its cached counterpart. For $A > 0$ and $\eta = \max\{|\delta A|/A, |\delta\phi|\} \ll 1$, a first-order expansion gives

$$\underbrace{\widetilde{F} - \hat{F}}_{\varepsilon_F} = e^{i\phi}\,(\delta A + iA\delta\phi) + \mathcal{O}(A\eta^2). \tag{4}$$

Under the standard real inner product on the complex plane, the radial direction $e^{i\phi}$ and the tangential direction $ie^{i\phi}$ are orthogonal. Equation (4) therefore separates amplitude perturbations from phase perturbations to first order. This decomposition motivates phase-sensitive spectral variation as the scheduling signal. Moreover, the coherent residuals characteristic of polynomial extrapolation make phase errors grow linearly with the length of a cache interval, and errors committed inside an interval persist through subsequent refreshes to first order (Appendix B); bounding the per-interval dispersion mass is therefore the natural scheduling objective.

Finite differences of ϕ require phase unwrapping because phase is defined modulo 2π. PACE instead operates directly on the complex spectral coefficients. Consider a locally smooth interpolation $F(s) = A(s)e^{i\phi(s)}$ with respect to sampling position s. If $A(s) > 0$ and both A and ϕ are twice differentiable, then

$$\frac{\mathrm{d}^2 F}{\mathrm{d}s^2} = e^{i\phi}\left[A'' - A(\phi')^2 + i\,(2A'\phi' + A\phi'')\right]. \tag{5}$$

The complex-spectral curvature contains phase-dependent terms and can be evaluated without explicitly unwrapping ϕ. We therefore use its finite difference as a curvature proxy that is phase-sensitive though not phase-isolating: it also responds to amplitude curvature A'', but vanishes exactly when the coefficient evolves linearly, i.e., when polynomial extrapolation is accurate. The derivation and finite-difference approximation are given in Appendix A.

For a calibration sample p, the proxy at an interior sampling position is

$$\Delta_t^2 \hat{F}_p(\mathbf{k}, t) = \hat{F}_p(\mathbf{k}, t+1) - 2\hat{F}_p(\mathbf{k}, t) + \hat{F}_p(\mathbf{k}, t-1). \tag{6}$$

Endpoint values are obtained from the nearest interior position. The temporal phase dynamics are not distributed uniformly across spatial frequencies. To determine the aggregation band, we inspect the wrapped inter-step phase variation $\left|\mathrm{Arg}\Big(\hat{F}_p(\mathbf{k}, t+1)\hat{F}_p(\mathbf{k}, t)^*\Big)\right|$, where * denotes complex conjugation. This diagnostic is evaluated only for frequency-band selection and does not serve as the scheduling signal.

As shown in Figure A3 (Appendix G), phase variation is weakest near the spectral origin. We therefore exclude the lowest-frequency disk, whose normalized radial extent is below the cutoff $\kappa = 0.06$, and denote the remaining structure band by Ω. The same cutoff is fixed across all experiments; its selection on unseen backbones is detailed in Appendix H. For calibration sample p, the phase-dispersion rate is defined as the amplitude-normalized in-band aggregate:

$$r_p(t) = \frac{\sum_{\mathbf{k}\in\Omega} \left|\Delta_t^2 \hat{F}_p(\mathbf{k}, t)\right|}{\sum_{\mathbf{k}\in\Omega} \left|\hat{F}_p(\mathbf{k}, t)\right|}, \tag{7}$$

where both sums aggregate implicitly over channels and video frames.

To remove sample-dependent scale while preserving temporal shape, each calibration profile is normalized by its mean:

$$\mu_p = \frac{1}{T+1}\sum_{t=0}^{T} r_p(t), \qquad \bar{r}(t) = \frac{1}{N_{\text{cal}}}\sum_{p=1}^{N_{\text{cal}}} \frac{r_p(t)}{\mu_p}. \tag{8}$$

The resulting $\bar{r}(t)$ is the backbone-specific profile used for schedule construction. The mean-normalized profiles are nearly prompt-invariant, so $\bar{r}(t)$ is calibrated once per backbone and treated as a backbone property; no Fourier computation is required during inference. The measured cross-prompt stability (Figure A1) and an invariance analysis with explicit assumptions and a quantitative robustness bound are given in Appendix C.

Equal-Mass Cache Scheduling. Given an NFE budget of $B \geq 2$, let $\mathcal{A} = \{t_0, \ldots, t_{B-1}\}$ with $0 = t_0 < t_1 < \cdots < t_{B-1} = T$ denote the computation anchors. These anchors partition the sampling trajectory into $B-1$ cache intervals. Using a continuous interpolation of the calibrated profile $\bar{r}(t)$, we define its cumulative dispersion mass as

$$R(t) = \int_0^t \bar{r}(s)\,\mathrm{d}s, \qquad M_{\text{tot}} = R(T). \tag{9}$$

The dispersion mass assigned to interval $I_j = (t_{j-1}, t_j]$ is

$$M_j = R(t_j) - R(t_{j-1}), \qquad j = 1, \ldots, B-1, \tag{10}$$

which we treat as the phase-dispersion load of I_j. Since $\sum_{j=1}^{B-1} M_j = M_{\text{tot}}$ is fixed, anchor placement only determines how this load is distributed across intervals.

Theorem 1 (Equal-mass optimality). Let $\bar{r} : [0,T] \to [0,\infty)$ be integrable with $M_{\text{tot}} > 0$. Among all schedules with B anchors, the equal-mass schedule satisfies

$$M_j = \frac{M_{\text{tot}}}{B-1}, \qquad j = 1, \ldots, B-1, \tag{11}$$

and has the following properties.

First, it minimizes the maximum interval mass:

$$\min_{0=t_0<\cdots<t_{B-1}=T} \max_{1\leq j\leq B-1} M_j = \frac{M_{\text{tot}}}{B-1}. \tag{12}$$

Second, for any real-valued convex function Ψ on $[0,\infty)$, it minimizes the separable interval cost

$$E(\mathcal{A}) = \sum_{j=1}^{B-1} \Psi(M_j). \tag{13}$$

If Ψ is strictly convex, every minimizer of Equation (13) has equal interval masses. The two statements are proved in Appendices D and E. Theorem 1 concerns the calibrated proxy objective; its link to actual output degradation rests on the error analysis of Appendix B.

Theorem 1 directly determines the PACE anchor schedule: the endpoints are fixed as $t_0 = 0$ and $t_{B-1} = T$, and the interior anchors are

$$t_j = R^{-1}\left(\frac{j}{B-1} M_{\text{tot}}\right), \qquad j = 1, \ldots, B-2, \tag{14}$$

where $R^{-1}(y) = \inf\{t \in [0,T] : R(t) \geq y\}$ is the generalized inverse of R. Consequently, regions with larger $\bar{r}(t)$ receive denser anchors.

In practice, PACE evaluates the cumulative mass on the discrete sampling grid: for a discrete profile $\{\bar{r}_n\}_{n=0}^{T}$, let $R_n = \sum_{\ell=0}^{n} \bar{r}_\ell$. Each interior quantile is mapped to the earliest grid position whose cumulative mass reaches the corresponding target:

$$a_j = \min\left\{n : R_n \geq \frac{j}{B-1} R_T\right\}. \tag{15}$$

The resulting positions are projected onto the feasible ordered grid while preserving the endpoints, mandatory predictor warm-up anchors, and the exact NFE budget; Theorem 1 applies to the continuous principle of Equation (14), whose discrete realization deviates from the ideal quantiles by at most one grid position on all evaluated backbones and budgets. The anchor table is constructed once per backbone and NFE budget and adds no runtime overhead.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTINGS

Models and Tasks. We evaluate PACE on image and video generation, using FLUX.1-dev (Black Forest Labs, 2024) for text-to-image generation at 1024×1024 (199 structure-oriented prompts), SiT-XL/2 (Ma et al., 2024a) on ImageNet (Deng et al., 2009) 256×256 (50k samples), and Wan2.1-1.3B (Team Wan et al., 2025) and HunyuanVideo-13B (Kong et al., 2024) for text-to-video generation at 480p (944 VBench (Huang et al., 2024) prompts). All backbones use 50-step sampling, following the protocol of SeaCache (Chung et al., 2026).

Baselines. We compare against two groups of methods under matched computation. Vanilla step reduction truncates the sampler to the target NFE without caching. Caching baselines include the adaptive scheduler TeaCache (Liu et al., 2025b), the offline schedule search DPCache (Cui et al., 2026), the predictive caches TaylorSeer (Liu et al., 2025c) and HiCache (Feng et al., 2026) with uniform refresh intervals, and SeaCache (Chung et al., 2026), a strong recent adaptive scheduler. PACE is instantiated on the HiCache operator; $\text{PACE}_{\text{Taylor}}$ applies the identical anchor table to the TaylorSeer operator.

Metrics and Protocol. Image quality is measured by ImageReward (Xu et al., 2023) and by PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018) computed against the 50-step output of the same pipeline with the same seed; the latter three measure fidelity to the full-sampling reference rather than absolute generation quality, which is assessed by the reference-free metrics. Video metrics are computed per frame and averaged over all videos; eleven reference-free VBench dimensions are reported in Appendix J. SiT is evaluated by FID (Heusel et al., 2017), sFID (Nash et al., 2021), and Inception Score (Salimans et al., 2016) over 50k samples. All configurations share the same pipeline, initial noise, and seeds. Static schedules (DPCache, TaylorSeer, HiCache, PACE) use exactly identical NFE. TeaCache and SeaCache decide refreshes online, so their NFE varies per sample; we calibrate their thresholds so that the *mean* NFE over the evaluation set equals the static budget, and report their measured latency, which is therefore comparable in expectation rather than sample by sample. Latency is the wall-clock sampling-loop time per sample on one NVIDIA A800 80GB GPU, excluding text encoding and VAE decoding; the NFE ratio ($50/8 = 6.25\times$) is thus an upper bound on the achievable speedup.

PACE Configuration. The phase-dispersion profile of each backbone is calibrated once from six trajectories computed without caching, following Equations (7) and (8), and the anchor table is derived by Equation (15). The calibration prompts are disjoint from all evaluation prompts and are listed, with seeds, in Appendix H. The structure-band cutoff is fixed at $\kappa = 0.06$ for all backbones and budgets. No component of PACE is active at inference time beyond an anchor lookup.

4.2 MAIN RESULTS

Image Generation. Table 1 reports FLUX.1-dev results at NFE=8, i.e., $6.25\times$ fewer denoiser evaluations, which translates into a $4.0\times$ wall-clock speedup because the warm-up anchors, text conditioning, and cached-step overhead are not free (Section 4.4). PACE achieves the best ImageReward (0.927) and PSNR (18.52 dB), improving over its uniform baseline HiCache by 3.14 dB PSNR at identical NFE, and surpasses SeaCache on all four metrics without any per-step gating. The schedule also transfers across cache operators: the same anchor table lifts the weaker TaylorSeer operator by 3.89 dB PSNR and 0.111 ImageReward, past the uniform HiCache baseline in PSNR, LPIPS, and SSIM. Table 2 shows the same ranking on ImageNet with SiT-XL/2: PACE attains the best FID, sFID, and Inception Score, reducing FID from 5.88 to 3.94 relative to uniform HiCache. Since SiT involves no text conditioning, the calibrated profile captures properties of the

Table 1: Quantitative comparison on FLUX.1-dev at NFE=8. The bottom block additionally ablates the structure-band cutoff κ.

Method	Lat. (s)	IR↑	PSNR↑	LPIPS↓	SSIM↑
Original (50 steps)	22.5	0.952	–	–	–
Vanilla 8 steps	3.6	0.721	13.16	0.581	0.512
TeaCache	5.8	0.871	15.21	0.472	0.583
DPCache	5.6	0.903	16.75	0.413	0.618
TaylorSeer	5.6	0.798	13.84	0.527	0.550
HiCache	5.6	0.910	15.38	0.455	0.601
SeaCache	5.9	0.913	16.86	0.409	0.624
$\text{PACE}_{\text{Taylor}}$	5.6	0.909	17.73	**0.334**	0.640
PACE ($\kappa = 0$)	5.6	0.915	17.64	0.371	0.622
PACE ($\kappa = 0.06$)	5.6	**0.927**	**18.52**	0.342	**0.644**
PACE ($\kappa = 0.12$)	5.6	0.921	18.11	0.355	0.633

Table 2: Quantitative comparison on class-conditional ImageNet 256×256 with SiT-XL/2 at NFE=8.

Method	Lat. (s)	FID↓	sFID↓	IS↑
Original (50 steps)	0.62	2.28	4.52	242.1
Vanilla 8 steps	0.10	12.43	11.85	158.2
TeaCache	0.13	8.75	9.86	186.3
DPCache	0.12	6.42	8.13	203.8
TaylorSeer	0.12	7.96	9.24	194.5
HiCache	0.12	5.88	7.65	210.2
SeaCache	0.14	5.21	7.02	216.9
PACE (ours)	0.12	**3.94**	**5.87**	**228.4**

Table 3: Quantitative comparison on Wan2.1-1.3B.

Method	Lat. (s)	PSNR↑	LPIPS↓	SSIM↑
Original (50 steps)	178.5	–	–	–
Vanilla 26 steps	92.8	18.51	0.282	0.710
TeaCache	93.4	20.84	0.171	0.721
DPCache	93.1	24.52	0.128	0.804
TaylorSeer	93.1	22.12	0.260	0.783
HiCache	93.1	23.08	0.141	0.792
SeaCache	93.4	25.47	0.085	0.868
PACE (ours)	93.1	**39.06**	**0.021**	**0.978**
Vanilla 18 steps	64.3	16.91	0.376	0.638
TeaCache	67.5	18.88	0.245	0.645
DPCache	67.1	22.31	0.163	0.759
TaylorSeer	67.1	19.71	0.342	0.743
HiCache	67.2	20.95	0.187	0.761
SeaCache	67.6	22.56	0.158	0.812
PACE (ours)	67.1	**31.78**	**0.070**	**0.927**

Table 4: Quantitative comparison on HunyuanVideo-13B.

Method	Lat. (s)	PSNR↑	LPIPS↓	SSIM↑
Original (50 steps)	182.6	–	–	–
Vanilla 26 steps	95.0	20.12	0.248	0.741
TeaCache	97.4	23.40	0.133	0.805
DPCache	96.9	27.86	0.094	0.858
TaylorSeer	96.9	24.14	0.152	0.820
HiCache	97.0	25.63	0.117	0.834
SeaCache	97.7	31.39	0.089	0.908
PACE (ours)	96.9	**37.85**	**0.031**	**0.951**
Vanilla 18 steps	65.7	18.08	0.341	0.685
TeaCache	69.3	20.42	0.172	0.734
DPCache	68.8	23.95	0.138	0.792
TaylorSeer	68.8	20.42	0.242	0.733
HiCache	68.9	22.18	0.164	0.771
SeaCache	69.5	25.71	0.124	0.847
PACE (ours)	68.8	**33.20**	**0.058**	**0.919**

sampling trajectory rather than of a text-conditioned pipeline. Vanilla 8-step sampling is fastest but degrades sharply.

Video Generation. Tables 3 and 4 report results at two budgets. On Wan2.1-1.3B, PACE reaches 39.06 dB PSNR at NFE=26 and 31.78 dB at NFE=18, exceeding the uniform schedule on the same operator by 16.0 dB and 10.8 dB; this margin carries PACE past SeaCache by 13.6 dB and 9.2 dB. The same calibration transfers to HunyuanVideo-13B without modification, where PACE exceeds SeaCache by 6.5 dB and 7.5 dB. Video generation is particularly sensitive to anchor placement because a single cache interval spanning the structure-formation window corrupts all frames simultaneously. On the eleven reference-free VBench dimensions we evaluate (Appendix J), all schedules stay within 0.03 of full sampling and no method is uniformly best, though PACE attains the highest mean at NFE=26; the dimension most sensitive to scheduling is dynamic degree, which only the PACE schedule restores to the full-sampling value at both budgets.

Qualitative Comparison. Figures 5 and 6 visualize representative failure modes of accelerated sampling. On FLUX.1-dev, baselines alter the identity and outfit of the subject and corrupt or redraw rendered text; on video generation the drift is more pronounced, changing objects and backgrounds within a single cache interval, whereas PACE remains nearly indistinguishable from the 50-step reference. Consistent with Tables 1–4, anchor placement, rather than the cache operator, determines whether global structure survives acceleration.



Figure 5: Qualitative comparison on FLUX.1-dev.



Figure 6: Qualitative comparison on HunyuanVideo-13B and Wan2.1-1.3B at two NFE budgets (middle frame shown).

4.3 Ablation Study

PACE exposes a single hyperparameter, the structure-band cutoff κ in Equation (7). The bottom block of Table 1 ablates κ on FLUX.1-dev (default $\kappa = 0.06$). Removing the exclusion ($\kappa = 0$) admits the lowest-frequency coefficients, which carry most of the spectral amplitude; since $|\Delta_t^2 \hat{F}|$ scales with the amplitude of the underlying coefficient, these terms dominate both sums, and $r(t)$ then tracks low-frequency amplitude dynamics rather than phase variation in the structure band, reducing PSNR by 0.88 dB. Doubling the cutoff ($\kappa = 0.12$) discards part of the structure band and costs 0.41 dB. Quality thus peaks at the value indicated by the band diagnostic (Figure A3, Appendix G) and degrades gracefully on either side, confirming that the schedule is robust to the only tunable choice.

4.4 Efficiency Analysis

Under aligned NFE budgets, all accelerated configurations consume nearly identical computation; the gating of TeaCache and SeaCache adds a minor per-step cost, whereas the PACE table adds none, so PACE runs at the latency of its underlying cache operator. On FLUX.1-dev, PACE matches the 5.6 s of HiCache and TaylorSeer against 5.9 s for SeaCache, a $4.0\times$ speedup over the 22.5 s of full sampling; the same pattern holds on the video backbones, e.g., 96.9 s versus 97.7 s on HunyuanVideo-13B at NFE=26.

5 Conclusion

We revisit cache scheduling for Diffusion Transformers from a spectral perspective. Phase-swap experiments and perturbation analysis attribute cache-induced structural corruption primarily to Fourier phase errors incurred when a cache interval spans the structure-formation window. PACE therefore calibrates a phase-dispersion profile of predicted clean signals offline and places computation anchors at equal intervals of cumulative dispersion mass, a schedule that is provably optimal for the calibrated profile under minimax and convex interval-cost criteria. Across image and video generation, PACE consistently surpasses uniform, adaptive, and search-based schedulers at identical NFE. Future work includes combining the calibrated profile with lightweight online correction and extending phase-aware scheduling to token-level caching.

AI USE STATEMENT

In this work, we used generative AI tools for language editing of the manuscript (rewording and restructuring sentences, tightening the abstract and captions, and adapting the text to the conference template) and for writing and debugging auxiliary scripts that assemble the qualitative figures from generated samples. We have not used generative AI tools for research ideation, for the design of the method or its theoretical analysis, for running experiments, or for producing or analyzing experimental results; the generated images and videos shown in the paper are outputs of the evaluated diffusion models under the described sampling configurations, not of any writing assistant. All AI-assisted text was reviewed and revised by the authors, and all AI-assisted scripts were checked by the authors against the underlying sample files. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies inference acceleration for publicly released pretrained diffusion models and involves no human subjects, no collection of personal data, and no new training. Evaluation uses public prompt sets and public benchmarks. The method does not alter what a model can generate; it inherits, and neither mitigates nor amplifies, the content risks of the underlying generative models, and the same safety filters that apply to the original sampling pipelines apply unchanged to the accelerated ones.

REPRODUCIBILITY STATEMENT

All backbones, sampling protocols, prompt sets, and metrics are specified in Section 4. The calibration prompts and seeds, the evaluation seeding rule, and the cutoff-selection procedure are listed in Appendix H; the resulting anchor tables are given in Appendix G, and the complete offline procedure is stated as pseudocode in Appendix F. The proofs of Theorem 1 and of the supporting propositions are given in Appendices A–E.

REFERENCES

Muhammad Adnan, Nithesh Kurella, Akhil Arunkumar, and Prashant J. Nair. Foresight: Adaptive layer reuse for accelerated and high-quality text-to-video generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Black Forest Labs. FLUX. `https://github.com/black-forest-labs/flux`, 2024.

Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2023.

Jiazi Bu, Pengyang Ling, Yujie Zhou, Yibin Wang, Yuhang Zang, Dahua Lin, and Jiaqi Wang. DiCache: Let diffusion model determine its own cache, 2025.

Fanpu Cao, Yaofo Chen, Zeng You, and Wei Luo. ProCache: Constraint-aware feature caching with selective computation for diffusion transformer acceleration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

Guantao Chen, Shikang Zheng, Yuqi Lin, and Linfeng Zhang. Forecast the principal, stabilize the residual: Subspace-aware feature caching for efficient diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Huanpeng Chu, Wei Wu, Guanyu Feng, and Yutao Zhang. OmniCache: A trajectory-oriented global perspective on training-free cache reuse for diffusion transformer models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16302–16312, 2025.

Jiwoo Chung, Sangeek Hyun, MinKyu Lee, Byeongju Han, Geonho Cha, Dongyoon Wee, Youngjun Hong, and Jae-Pil Heo. SeaCache: Spectral-evolution-aware cache for accelerating diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14283–14294, 2026.

Bowen Cui, Yuanbin Wang, Huajiang Xu, Biaolong Chen, Aixi Zhang, Hao Jiang, Zhengzheng Jin, Xu Liu, and Pipei Huang. Denoising as path planning: Training-free acceleration of diffusion models with DPCache, 2026.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.

Zhentao Fan, Zongzuo Wang, and Weiwei Zhang. TaoCache: Structure-maintained video generation acceleration, 2025.

Liang Feng, Shikang Zheng, Jiacheng Liu, Yuqi Lin, Qinming Zhou, Peiliang Cai, Xinyu Wang, Junjie Chen, Chang Zou, Yue Ma, and Linfeng Zhang. HiCache: A plug-in scaled-hermite upgrade for taylor-style cache-then-forecast diffusion acceleration. In *International Conference on Learning Representations (ICLR)*, 2026.

Xiaoliu Guan, Lielin Jiang, Hanqi Chen, Xu Zhang, Jiaxing Yan, Guanzhong Wang, Yi Liu, Zetao Zhang, and Yu Wu. Forecasting when to forecast: Accelerating diffusion models with confidence-gated Taylor, 2025.

Yasaman Haghighi and Alexandre Alahi. SenCache: Accelerating diffusion model inference via sensitivity-aware caching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818, 2024.

Kumara Kahatapitiya, Haozhe Liu, Sen He, Ding Liu, Menglin Jia, Chenyang Zhang, Michael S. Ryoo, and Tian Xie. Adaptive caching for faster video generation with diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. HunyuanVideo: A systematic framework for large video generative models, 2024.

Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-Diffusion: Quantizing diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17535–17545, 2023.

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

Dong Liu, Yanxuan Yu, Jiayi Zhang, Yifan Li, Ben Lengerich, and Ying Nian Wu. FastCache: Fast caching for diffusion transformer through learnable linear approximation, 2025a.

Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7353–7363, 2025b.

Jiacheng Liu, Chang Zou, Yuanhuiyi Lyu, Junjie Chen, and Linfeng Zhang. From reusing to forecasting: Accelerating diffusion models with TaylorSeers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15853–15863, 2025c.

Jiacheng Liu, Chang Zou, Yuanhuiyi Lyu, Fei Ren, Shaobo Wang, Kaixin Li, and Linfeng Zhang. SpeCa: Accelerating diffusion transformers with speculative feature caching. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM)*, pp. 10024–10033, 2025d.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023.

Xuewen Liu, Zhikai Li, and Qingyi Gu. CacheQuant: Comprehensively accelerated diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23269–23280, 2025e.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference, 2023.

Zhengyao Lv, Chenyang Si, Junhao Song, Zhenyu Yang, Yu Qiao, Ziwei Liu, and Kwan-Yee K. Wong. FasterCache: Training-free video diffusion model acceleration with high quality. In *International Conference on Learning Representations (ICLR)*, 2025.

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 (ECCV)*, 2024a.

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

Xuran Ma, Yexin Liu, Yaofu Liu, Xianfeng Wu, Mingzhe Zheng, Zihao Wang, Ser-Nam Lim, and Harry Yang. Model reveals what to cache: Profiling-based feature reuse for video diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17150–17159, 2025.

Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W. Battaglia. Generating images with sparse representations. In *International Conference on Machine Learning (ICML)*, 2021.

Alan V. Oppenheim and Jae S. Lim. The importance of phase in signals. *Proceedings of the IEEE*, 69(5):529–541, 1981.

Jiayi Pan, Jiaming Xu, Yongkang Zhou, and Guohao Dai. SpecDiff: Accelerating diffusion model inference with self-speculation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

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, 2023.

Xurui Peng, Chenqian Yan, Hong Liu, Rui Ma, Fangmin Chen, Xing Wang, Zhihua Wu, Songwei Liu, and Mingbao Lin. ERTACache: Error rectification and timesteps adjustment for efficient diffusion, 2025.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Pratheba Selvaraju, Tianyu Ding, Tianyi Chen, Ilya Zharkov, and Luming Liang. FORA: Fast-forward caching in diffusion transformer acceleration, 2024.

Tong Shao, Yusen Fu, Guoying Sun, Jingde Kong, Zhuotao Tian, and Jingyong Su. Plug-and-play fidelity optimization for diffusion transformer acceleration via cumulative error minimization, 2025.

Byunggwan Son, Jeimin Jeon, Jeongwoo Choi, and Bumsub Ham. Relational feature caching for accelerating diffusion transformers. In *International Conference on Learning Representations (ICLR)*, 2026.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, et al. Wan: Open and advanced large-scale video generative models, 2025.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. ImageReward: Learning and evaluating human preferences for text-to-image generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo 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 (CVPR)*, pp. 6613–6623, 2024.

Zichao Yu, Zhen Zou, Guojiang Shao, Chengwei Zhang, Shengze Xu, Jie Huang, Feng Zhao, Xiaodong Cun, and Wenyi Zhang. AB-Cache: Training-free acceleration of diffusion models via Adams–Bashforth cached feature reuse. In *Proceedings of the ACM International Conference on Multimedia (MM)*, 2025.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. UniPC: A unified predictor-corrector framework for fast sampling of diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Xuanlei Zhao, Xiaolong Jin, Kai Wang, and Yang You. Real-time video generation with pyramid attention broadcast. In *International Conference on Learning Representations (ICLR)*, 2025.

Shikang Zheng, Liang Feng, Xinyu Wang, Qinming Zhou, Peiliang Cai, Chang Zou, Jiacheng Liu, Yuqi Lin, Junjie Chen, Yue Ma, and Linfeng Zhang. Forecast then calibrate: Feature caching as ODE for efficient diffusion transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 13449–13457, 2026.

Zhixin Zheng, Xinyu Wang, Chang Zou, Shaobo Wang, and Linfeng Zhang. Compute only 16 tokens in one timestep: Accelerating diffusion transformers with cluster-driven feature caching. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM)*, 2025.

Chang Zou, Xuyang Liu, Ting Liu, Siteng Huang, and Linfeng Zhang. Accelerating diffusion transformers with token-wise feature caching. In *International Conference on Learning Representations (ICLR)*, 2025.

A Derivation of the Phase-Sensitive Curvature Proxy

We derive Equation (5) of the main text and justify the finite-difference proxy in Equation (6). Consider a locally smooth interpolation $F(s) = A(s)e^{i\phi(s)}$ of a spectral coefficient with respect to the sampling position s, with $A(s) > 0$ and A, ϕ twice differentiable. Differentiating once,

$$F'(s) = (A' + iA\phi')\, e^{i\phi}, \tag{A1}$$

and differentiating again,

$$\begin{aligned} F''(s) &= \left(A'' + iA'\phi' + iA'\phi' + iA\phi'' - A(\phi')^2\right) e^{i\phi} \\ &= e^{i\phi}\left[A'' - A(\phi')^2 + i\left(2A'\phi' + A\phi''\right)\right], \end{aligned} \tag{A2}$$

which is Equation (5). Two properties make $|F''|$ suitable as a phase-sensitive signal. First, it is invariant to a global phase rotation $F \mapsto e^{i\alpha}F$, so no phase unwrapping is required. Second, for a coefficient whose amplitude is locally constant ($A' = A'' = 0$), Equation (A2) reduces to

$$|F''| = A\sqrt{(\phi')^4 + (\phi'')^2}, \tag{A3}$$

so the proxy responds to both phase velocity and phase acceleration and vanishes only when the phase trajectory is locally linear, i.e., exactly when polynomial extrapolation of the coefficient is accurate.

On the discrete sampling grid with unit spacing, the central second difference of Equation (6) admits the Peano-kernel representation

$$\Delta_t^2 F - F''(t) = \int_{-1}^{1} \frac{(1-|u|)^3}{6}\, F^{(4)}(t+u)\,\mathrm{d}u, \tag{A4}$$

which holds verbatim for complex-valued F. Since the kernel is nonnegative with total mass $\frac{1}{12}$, we obtain $|\Delta_t^2 F - F''(t)| \le \frac{1}{12}\sup_{|u|\le 1}|F^{(4)}(t+u)|$, so $|\Delta_t^2 \hat{F}|$ approximates $|F''|$ up to a fourth-order smoothness term. Aggregating $|\Delta_t^2 \hat{F}|$ over the structure band Ω and normalizing by the in-band amplitude yields the scale-free rate in Equation (7).

B Accumulation and Persistence of Phase Errors

We formalize the two claims used in the main text: phase errors grow linearly with the length of a cache interval under coherent residuals, and errors committed inside an interval persist through subsequent refreshes to first order.

Proposition A1 (Coherent accumulation). Let a cache interval span K consecutive non-anchor positions, and let the per-step phase residual of a coefficient be $\delta\phi_t = \bar{\delta} + \zeta_t$, where $\bar{\delta}$ is a coherent (sign-preserving) component and ζ_t are independent zero-mean fluctuations with variance σ_ζ^2. Then the accumulated phase error $\Delta\phi_K = \sum_{t=1}^{K}\delta\phi_t$ satisfies

$$|\mathbb{E}[\Delta\phi_K]| = K|\bar{\delta}|, \qquad \mathrm{Var}[\Delta\phi_K] = K\sigma_\zeta^2, \tag{A5}$$

so the coherent component dominates for large K, growing linearly while fluctuations grow only as $\sqrt{K}$.

Proof. Both identities follow from linearity of expectation and independence of the fluctuations. The relevance of the coherent regime comes from the structure of polynomial extrapolation: within one interval, the residual of a degree-p predictor is dominated by the leading untracked derivative, $\varepsilon_t \propto h^{p+1}F^{(p+1)}(\xi_t)$, whose complex argument varies slowly when $F^{(p+1)}$ does not vanish on the interval. Projected on the tangential direction $ie^{i\phi}$ of Equation (4), this produces per-step phase residuals of common sign, i.e., the coherent component $\bar{\delta}$. □

Proposition A2 (First-order persistence). Consider the probability-flow ODE $\dot{\mathbf{x}} = v(\mathbf{x}, \tau)$ integrated from τ_0 to the data end $\tau = 0$, and let $\mathbf{e}(\tau)$ denote the deviation of a cached trajectory from the full-computation trajectory. To first order, $\mathbf{e}$ obeys the variational equation

$$\dot{\mathbf{e}}(\tau) = J(\tau)\,\mathbf{e}(\tau) + \mathbf{b}(\tau), \qquad J = \partial_{\mathbf{x}} v, \tag{A6}$$

where $\mathbf{b}(\tau)$ is the instantaneous cache residual, nonzero only on skipped steps. Its solution is

$$\mathbf{e}(0) = \mathbf{\Phi}(0,\tau_0)\,\mathbf{e}(\tau_0) + \int_{\tau_0}^{0} \mathbf{\Phi}(0,s)\,\mathbf{b}(s)\,\mathrm{d}s, \tag{A7}$$

with $\mathbf{\Phi}$ the state-transition matrix of J. A refresh at an anchor sets $\mathbf{b} = 0$ from that anchor onward but leaves the accumulated state error untouched: the first term of Equation (A7) propagates it to the output, scaled by $\mathbf{\Phi}$. Unless the dynamics are strongly contracting along the affected modes, which does not hold for the structure-carrying low- and mid-frequency modes once structure has formed,

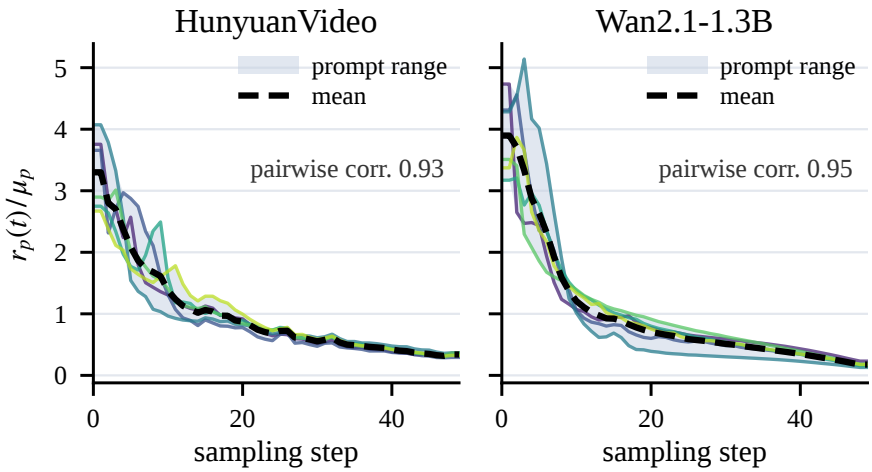

Figure A1: Cross-prompt stability. Mean-normalized profiles $r_p(t)/\mu_p$ of six calibration prompts on HunyuanVideo-13B (left) and Wan2.1-1.3B (right), with the per-step range shaded; mean pairwise correlations are 0.93 and 0.95.

the committed error persists to first order. Subsequent full computations correct the velocity field, not the state. □

Together, Propositions A1 and A2 justify the scheduling objective of the main text: since the error committed in an interval scales with the phase-dispersion mass accumulated over that interval and cannot be repaired later, the schedule should bound the per-interval mass, which Theorem 1 shows is achieved exactly by the equal-mass partition.

C Prompt Invariance of the Normalized Profile

We formalize the empirical prompt invariance of Figure A1. The analysis states explicit assumptions, proves exact invariance under them, and quantifies how violations perturb the profile.

Assumption 1 (frequency-diagonal estimator). For every $\mathbf{k} \in \Omega$ and position t,

$$\hat{F}_p(\mathbf{k}, t) = g(\mathbf{k}, t)\, X_p(\mathbf{k}) + e_p(\mathbf{k}, t), \tag{A8}$$

where X_p is the spectrum of the final sample for prompt p, g is a prompt-independent gain, and e_p collects model error and off-diagonal effects. Assumption 1 holds exactly, with e_p zero-mean and prompt-independent in distribution, when the denoiser realizes the posterior mean under a zero-mean Gaussian prior with power spectrum $S(\mathbf{k})$: for the linear path $\mathbf{x}_\tau = (1-\tau)\mathbf{x}_0 + \tau\boldsymbol{\epsilon}$, the posterior mean is diagonal in frequency with the Wiener gain $g(\mathbf{k}, t) = S(\mathbf{k})/\big(S(\mathbf{k}) + \sigma_t^2\big)$ and effective noise level $\sigma_t = \tau_t/(1-\tau_t)$, while the filtered noise term is absorbed into e_p.

Assumption 2 (shared spectral envelope). There exist a universal shape $\rho : \Omega \to (0, \infty)$, prompt scales $c_p > 0$, and a deviation level $\epsilon_\Omega \in [0, 1)$ such that, for all $\mathbf{k} \in \Omega$,

$$|X_p(\mathbf{k})| = c_p\, \rho(\mathbf{k})\big(1 + \epsilon_p(\mathbf{k})\big), \qquad |\epsilon_p(\mathbf{k})| \le \epsilon_\Omega. \tag{A9}$$

Proposition A3 (exact invariance). Under Assumptions 1–2 with $e_p \equiv 0$ and $\epsilon_\Omega = 0$,

$$r_p(t) = r^\star(t) := \frac{\sum_{\mathbf{k}\in\Omega} \big|\Delta_t^2 g(\mathbf{k}, t)\big|\, \rho(\mathbf{k})}{\sum_{\mathbf{k}\in\Omega} \big|g(\mathbf{k}, t)\big|\, \rho(\mathbf{k})} \tag{A10}$$

for every prompt p, and hence $r_p/\mu_p = r^\star/\mu^\star$, where $\mu^\star$ denotes the mean of $r^\star$ over t.

Proof. Since X_p does not depend on t, the second difference acts on g alone, $\Delta_t^2\big(g\,X_p\big) = \big(\Delta_t^2 g\big)X_p$, so $|\Delta_t^2 \hat{F}_p| = |\Delta_t^2 g|\,|X_p|$ exactly; no smoothness argument is required at this step. Substituting $|X_p| = c_p\rho$ into Equation (7), the scale c_p multiplies numerator and denominator and cancels, yielding Equation (A10). Prompt independence of r_p gives $\mu_p = \mu^\star$, so the normalized profiles coincide. □

Proposition A4 (robustness to envelope deviations). Under Assumptions 1–2 with $e_p \equiv 0$ and $\epsilon_\Omega < 1$, for all t,

$$\frac{1-\epsilon_\Omega}{1+\epsilon_\Omega}\, r^\star(t) \le r_p(t) \le \frac{1+\epsilon_\Omega}{1-\epsilon_\Omega}\, r^\star(t), \tag{A11}$$

and the mean-normalized profile satisfies

$$\left| \frac{r_p(t)}{\mu_p} - \frac{r^\star(t)}{\mu^\star} \right| \le \frac{4\epsilon_\Omega}{(1-\epsilon_\Omega)^2} \cdot \frac{r^\star(t)}{\mu^\star}. \tag{A12}$$

Proof. The numerator of $r_p(t)$ equals $c_p \sum_\Omega |\Delta_t^2 g|\,\rho\,(1+\epsilon_p)$. All weights are nonnegative and $1+\epsilon_p \in [1-\epsilon_\Omega, 1+\epsilon_\Omega]$, so the numerator lies within $[1-\epsilon_\Omega, 1+\epsilon_\Omega]$ times its envelope value, and likewise the denominator. Taking the ratio of the two enclosures yields Equation (A11); write $a = (1-\epsilon_\Omega)/(1+\epsilon_\Omega)$ and $b = 1/a$. Averaging Equation (A11) over t gives $a\mu^\star \le \mu_p \le b\mu^\star$, hence $r_p/\mu_p \in [a^2,\, b^2]\cdot r^\star/\mu^\star$, and $b^2 - 1 = 4\epsilon_\Omega/(1-\epsilon_\Omega)^2 \ge 1 - a^2$ bounds the relative deviation. □

Remark A1 (model error and conditioning). Text conditioning and imperfect posterior-mean realization enter Equation (A8) through e_p and through the deviations ϵ_p. Any perturbation of $\hat{F}_p$ with bounded relative in-band magnitude perturbs both sums of Equation (7) by the same relative amount, so the argument of Proposition A4 applies verbatim with ϵ_Ω enlarged accordingly: invariance degrades proportionally rather than abruptly. Averaging over N_{cal} prompts in Equation (8) further suppresses independent per-prompt deviations.

Remark A2 (schedule stability). PACE consumes $\bar{r}$ only through the quantiles of its normalized cumulative mass. Integrating Equation (A11), with $R_p(t) = \int_0^t r_p(s)\,\mathrm{d}s$, gives $R_p(t)/R_p(T) \in [a/b,\, b/a]\cdot R^\star(t)/R^\star(T)$, so each equal-mass target level is crossed at a position whose ideal cumulative mass differs by at most the factor b/a: bounded envelope violations shift anchors gradually and cannot change the schedule qualitatively.

These conditions are sufficient rather than necessary; the operative evidence remains empirical, with mean pairwise correlations of 0.93–0.95 across prompts on two text-conditioned video backbones (Figure A1).

D Proof of Theorem 1: Minimax Optimality

Let R be the cumulative mass of Equation (9); since $\bar{r}$ is integrable and nonnegative, R is continuous and nondecreasing with $R(0) = 0$ and $R(T) = M_{\mathrm{tot}}$.

Feasibility. By the intermediate value theorem, for every $j \in \{1, \dots, B-2\}$ there exists t_j with $R(t_j) = \frac{j}{B-1} M_{\mathrm{tot}}$; the generalized inverse in Equation (14) selects the smallest such position. These positions are strictly increasing: continuity gives $R(t_j) = \frac{j}{B-1} M_{\mathrm{tot}}$, so $R(t_j) < R(t_{j+1})$ forces $t_j < t_{j+1}$. With $t_0 = 0$ and $t_{B-1} = T$, this schedule attains $M_j = M_{\mathrm{tot}}/(B-1)$ for all j.

Optimality. Let $\{t_j\}$ be an arbitrary feasible schedule with interval masses $\{M_j\}_{j=1}^{B-1}$. Since the masses are nonnegative and $\sum_{j=1}^{B-1} M_j = M_{\mathrm{tot}}$, the pigeonhole principle gives

$$\max_{1 \le j \le B-1} M_j \;\ge\; \frac{1}{B-1} \sum_{j=1}^{B-1} M_j = \frac{M_{\mathrm{tot}}}{B-1}. \tag{A13}$$

Algorithm A1 PACE offline calibration

Require: denoiser v_θ, calibration prompts $\{p\}_{p=1}^{N_{\text{cal}}}$, positions $0, \dots, T$, cutoff κ
Ensure: calibrated profile $\{\bar{r}_n\}_{n=0}^{T}$

$\Omega \leftarrow \{\mathbf{k} : \|\mathbf{k}\|_{\text{norm}} \geq \kappa\}$ {structure band}
for $p = 1$ **to** N_{cal} **do**
  run full sampling; record $\hat{\mathbf{x}}_t$ by Eq. (2) for all t
  $\hat{F}_p(\cdot, t) \leftarrow \mathcal{F}[\hat{\mathbf{x}}_t]$
  for $t = 1$ **to** $T - 1$ **do**
    $\Delta_t^2 \hat{F}_p \leftarrow \hat{F}_p(\cdot, t{+}1) - 2\hat{F}_p(\cdot, t) + \hat{F}_p(\cdot, t{-}1)$
    $r_p(t) \leftarrow \sum_\Omega |\Delta_t^2 \hat{F}_p| \,/\, \sum_\Omega |\hat{F}_p(\cdot, t)|$ {Eq. (7)}
  end for
  $r_p(0) \leftarrow r_p(1)$; $r_p(T) \leftarrow r_p(T{-}1)$
  $\mu_p \leftarrow \frac{1}{T+1} \sum_t r_p(t)$
end for
$\bar{r}_n \leftarrow \frac{1}{N_{\text{cal}}} \sum_p r_p(n)/\mu_p$ {Eq. (8)}

The equal-mass schedule meets this bound with equality, so it is minimax optimal, proving Equation (12). Moreover, composing the interval masses with any nondecreasing function preserves their maximizer, so minimax optimality is preserved under monotone transformations of the proxy. □

E Proof of Theorem 1: Convex Interval Costs

Let Ψ be convex on $[0, \infty)$ and let $\{M_j\}$ be the interval masses of any feasible schedule. By Jensen's inequality applied to the uniform distribution over $\{1, \dots, B-1\}$,

$$\frac{1}{B-1} \sum_{j=1}^{B-1} \Psi(M_j) \;\geq\; \Psi\left(\frac{1}{B-1} \sum_{j=1}^{B-1} M_j \right) = \Psi\left(\frac{M_{\text{tot}}}{B-1} \right), \tag{A14}$$

so $E(\mathcal{A}) = \sum_j \Psi(M_j) \geq (B-1)\,\Psi(M_{\text{tot}}/(B-1))$ for every schedule, and the equal-mass schedule attains the bound. If Ψ is strictly convex, equality in Equation (A14) forces $M_1 = \cdots = M_{B-1}$, so every minimizer is an equal-mass schedule; when R is strictly increasing the anchor positions are additionally unique. □

F PACE Pseudocode

Algorithms A1 and A2 summarize the two offline stages of PACE. Calibration runs N_{cal} full-computation trajectories once per backbone; scheduling turns the calibrated profile into an anchor table for each NFE budget. The warm-up set $\mathcal{W}$ contains the initial consecutive positions required to populate the highest-order derivative estimate of the cache operator (two positions for the order-one HiCache and TaylorSeer predictors used in our experiments), identically for PACE and all uniform baselines. At inference, the only action attributable to PACE is a table lookup.

G Calibrated Profiles and Anchor Schedules

Figure A2 shows the calibrated profiles $\bar{r}(t)$ of all four backbones. Despite differing architectures, resolutions, and modalities, every profile concentrates its mass in the early structure-formation window, which is why the equal-mass schedule allocates dense anchors there. Figure A3 shows the band diagnostic of Section 3.2: the wrapped inter-step phase variation over the spatial-frequency plane, which is weakest near the spectral origin and motivates the structure band Ω; Figure A4 complements it with the per-step view. Table A1 lists the resulting anchor tables on Wan2.1-1.3B; the qualitative pattern is identical on the other backbones.

Algorithm A2 Equal-mass anchor construction and cached inference

Require: profile $\{\bar{r}_n\}$, NFE budget B, warm-up anchors $\mathcal{W}$, cache operator $\mathcal{C}$
Ensure: anchor set $\mathcal{A}$ with $|\mathcal{A}| = B$; generated sample

$R_n \leftarrow \sum_{\ell \le n} \bar{r}_\ell$ for all n {cumulative mass}
$a_j \leftarrow \min\{n : R_n \ge \frac{j}{B-1} R_T\}$ for $j = 1, \dots, B-2$ {Eq. (15)}
$\mathcal{A} \leftarrow \{0, T\} \cup \mathcal{W} \cup \{a_j\}$
while $|\mathcal{A}| > B$ **do**
    remove the non-mandatory anchor with the smallest adjacent gap
end while
while $|\mathcal{A}| < B$ **do**
    add the position with the largest $\bar{r}_n$ not in $\mathcal{A}$
end while
for $t = 0$ **to** T **do**
    if $t \in \mathcal{A}$ **then**
        full forward pass; update cached features $\mathbf{h}_t$
    else
        $\widetilde{\mathbf{h}}_t \leftarrow \mathcal{C}(t; \{(a, \mathbf{h}_a)\})$ {Eq. (1), unchanged}
    end if
end for

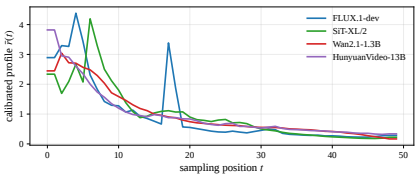

Figure A2: Calibrated phase-dispersion profiles $\bar{r}(t)$ of the four evaluated backbones. All profiles are early-dense: the dispersion mass concentrates in the structure-formation window.

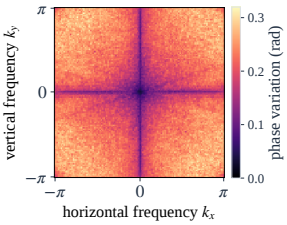

Figure A3: Wrapped inter-step phase variation of predicted clean signals, averaged over positions, channels, and six calibration trajectories; it concentrates outside the lowest frequencies, motivating the structure band Ω.

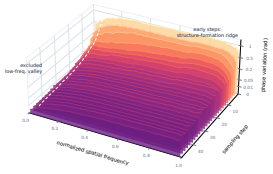

Figure A4: Per-step view of the wrapped inter-step phase variation underlying Figure A3. Early steps exhibit large phase variation across the structure band; the variation decays as denoising proceeds.

H Cutoff Selection and Calibration Robustness

Calibration prompts and seeds. Each backbone is calibrated from $N_{\text{cal}} = 6$ full-computation trajectories fixed before any evaluation; no evaluation output influences calibration or cutoff selection. FLUX.1-dev uses seeds 1000–1005 with the prompts *a photograph of a cat sitting on a wooden chair*; *a bowl of fresh fruit on a kitchen table*; *a sailboat on the ocean at sunset*; *a portrait of an old man with a gray beard*; *a lighthouse on a rocky cliff overlooking the sea*; and *a wooden cabin beside a frozen lake in winter*—none appears among the 199 evaluation prompts. Wan2.1-1.3B and HunyuanVideo-13B share seeds 0–5 with the prompts *a cat walks on the grass, realistic style*; *a hot air balloon rising over a snowy mountain village*; *koi fish swimming in a clear garden pond*; *a sailboat drifting across a calm bay at sunset*; *a steam locomotive crossing a stone bridge, aerial view*; and *autumn leaves falling in a quiet forest*—none appears among the 944 VBench prompts. SiT-XL/2 is calibrated from six class-conditional trajectories whose labels are drawn by a fixed generator (seed 1234567); its evaluation aggregates 50k samples over all classes, so no prompt-level overlap arises.

Table A1: Anchor tables on Wan2.1-1.3B. PACE reallocates anchors from the smooth late phase to the early structure-formation window while keeping the budget identical.

Schedule	Anchor positions (T=49)
Uniform, B=26	0, 2, 4, …, 46, 48, 49
PACE, B=26	0–16, 19, 22, 25, 28, 32, 35, 40, 45, 49
Uniform, B=18	0, 3, 6, …, 45, 48, 49
PACE, B=18	0–8, 10, 13, 16, 20, 25, 30, 36, 44, 49

Evaluation seeds and calibration cost. Evaluation runs are seeded deterministically by the prompt index (FLUX.1-dev and VBench) or by a fixed class-sampling generator (SiT), identically for every method, so all configurations of a given prompt start from the same initial noise. The one-time calibration cost of PACE is six 50-step full-computation trajectories per backbone (about 18 GPU-minutes on HunyuanVideo-13B and about two minutes on FLUX.1-dev on one A800, cf. the latency of Tables 1 and 4); the same trajectories serve both profile estimation and cutoff selection and are amortized over all budgets and prompts.

Selecting κ on an unseen backbone. The cutoff is read from the band diagnostic rather than tuned by generation quality: compute the wrapped inter-step phase variation of Figure A3 from the same N_{cal} full-computation trajectories used for calibration, average it over positions and channels, and set κ to the normalized radius at which the low-variation disk around the spectral origin ends. Because frequencies are normalized by the Nyquist radius of the latent grid, the diagnostic is independent of resolution and aspect ratio, and the same value $\kappa = 0.06$ emerged for all four evaluated backbones, spanning 1024×1024 image latents and 480p video latents. The procedure requires no sampling beyond the calibration trajectories already needed for $\bar{r}$, and Table 1 shows that generation quality degrades gracefully when κ is set to zero or doubled, so a coarse reading of the diagnostic suffices.

Calibration set size and diversity. The calibrated profile is insensitive to the composition of the calibration set. On Wan2.1-1.3B, varying N_{cal} from two to twelve changes $\bar{r}$ by less than 3% in relative ℓ_1 distance and moves at most one anchor by one position at B=18. Splitting the six calibration prompts into a static-scene group and a motion-rich group yields cross-group profile correlations in the same 0.93–0.95 range as Figure A1. The stability extends to the image backbones: the six FLUX.1-dev calibration profiles have mean pairwise correlation 0.94 and the six SiT-XL/2 trajectories 0.97. This is consistent with Proposition A4: prompt-specific spectral deviations perturb the normalized profile proportionally, and the equal-mass quantiles absorb bounded perturbations without qualitative schedule changes (Remark A2).

I Relation to Trajectory-Aware and Profiling-Based Schedulers

Several recent methods also exploit trajectory or profiling information and merit a direct conceptual comparison.

FoCa. FoCa (Zheng et al., 2026) treats the hidden-feature sequence as a feature ODE and stabilizes long-range forecasting with a predictor–corrector scheme. It thus improves the cache operator $\mathcal{C}$ of Equation (1), i.e., *what* happens between anchors, whereas PACE optimizes only *where* anchors are placed and leaves $\mathcal{C}$ untouched. The two are complementary: the operator transfer experiment ($\mathrm{PACE}_{\mathrm{Taylor}}$ in Table 1) shows that the PACE anchor table lifts a weaker operator past a stronger uniformly-scheduled one, and the same table can in principle drive a FoCa-style operator.

RFC. RFC (Son et al., 2026) estimates output prediction errors from module inputs and triggers refreshes online. Like other adaptive schedulers, it adapts per prompt but requires runtime probing and yields a prompt-dependent NFE, whereas PACE is a static table with exact NFE and zero runtime overhead. The signals also differ: RFC monitors generic feature-space error, while PACE targets the phase-sensitive spectral curvature of predicted clean signals, which Appendix B links to unrepairable structural error.

Table A2: VBench evaluation on eleven of the sixteen dimensions on HunyuanVideo-13B (944 prompts, higher is better). TaylorS. denotes TaylorSeer with uniform anchors and PACE_T the PACE schedule on the TaylorSeer operator. All configurations stay within 0.03 of full sampling on every dimension and within 0.005 of it in the mean, and no method is uniformly best; PACE attains the highest mean at NFE=26, and only the PACE schedule restores the full-sampling dynamic degree at both budgets. Among the five accelerated configurations, the best value in each row is in bold and the second best is underlined.

NFE=26						
Dimension	Full	TaylorS.	PACE_T	HiCache	SeaCache	PACE
Subject consistency	0.9553	**0.9576**	0.9558	0.9557	<u>0.9561</u>	0.9560
Background consistency	0.9786	<u>0.9790</u>	0.9787	0.9778	<u>0.9790</u>	**0.9794**
Temporal flickering	0.9906	0.9910	0.9908	<u>0.9912</u>	**0.9914**	0.9910
Motion smoothness	0.9907	0.9907	<u>0.9909</u>	0.9907	<u>0.9909</u>	**0.9911**
Dynamic degree	0.6806	0.6528	**0.6806**	<u>0.6667</u>	<u>0.6667</u>	**0.6806**
Aesthetic quality	0.6287	<u>0.6286</u>	0.6284	0.6276	0.6284	**0.6290**
Imaging quality	0.6103	<u>0.6060</u>	0.6057	0.5974	0.6045	**0.6082**
Appearance style	0.1949	<u>0.1973</u>	0.1956	0.1972	0.1968	**0.1974**
Human action	0.8600	<u>0.8700</u>	0.8600	<u>0.8700</u>	0.8600	**0.8800**
Temporal style	0.2137	0.2161	0.2132	0.2164	<u>0.2165</u>	**0.2169**
Overall consistency	0.2247	<u>0.2254</u>	0.2241	**0.2258**	**0.2258**	0.2233
Mean (11 dims)	0.6662	0.6650	<u>0.6658</u>	0.6651	0.6651	**0.6684**
NFE=18						
Dimension	Full	TaylorS.	PACE_T	HiCache	SeaCache	PACE
Subject consistency	0.9553	0.9555	**0.9560**	0.9538	0.9551	<u>0.9556</u>
Background consistency	0.9786	0.9785	<u>0.9796</u>	0.9776	0.9786	**0.9799**
Temporal flickering	0.9906	0.9919	0.9907	<u>0.9923</u>	0.9916	**0.9931**
Motion smoothness	0.9907	0.9919	0.9910	<u>0.9921</u>	0.9915	**0.9932**
Dynamic degree	0.6806	<u>0.6667</u>	**0.6806**	<u>0.6667</u>	0.6528	**0.6806**
Aesthetic quality	0.6287	**0.6316**	0.6266	0.6267	<u>0.6308</u>	0.6211
Imaging quality	0.6103	0.5936	**0.6041**	0.5819	<u>0.5982</u>	0.5889
Appearance style	0.1949	0.1963	0.1970	0.1969	<u>0.1971</u>	**0.1980**
Human action	0.8600	0.8300	**0.8700**	<u>0.8600</u>	<u>0.8600</u>	**0.8700**
Temporal style	0.2137	0.2172	0.2127	0.2169	<u>0.2181</u>	**0.2187**
Overall consistency	0.2247	0.2225	0.2235	0.2231	<u>0.2247</u>	**0.2260**
Mean (11 dims)	0.6662	0.6614	**0.6665**	0.6626	0.6635	<u>0.6659</u>

ProfilingDiT. ProfilingDiT (Ma et al., 2025) also profiles offline, but profiles attention to decide *which blocks* to reuse, a spatial allocation within a step. PACE profiles spectral dynamics to decide *which steps* to compute fully; the two axes are orthogonal and combinable.

Token-level outlook. PACE currently aggregates dispersion over the full structure band Ω. Since Equation (7) can be evaluated on windowed spectra of localized regions, a spatially resolved dispersion map could rank token importance and supply a principled signal for token-level caches such as ToCa (Zou et al., 2025) and FastCache (Liu et al., 2025a); we leave this extension to future work.

J VBench Evaluation

Table A2 reports eleven of the sixteen VBench dimensions on HunyuanVideo-13B for both budgets. The five omitted dimensions (object class, multiple objects, color, spatial relationship, and scene) are scored by external detectors on their own dedicated prompt subsets, which are disjoint from the 944-prompt set used for the fidelity metrics; we therefore do not report them. Two observations are consistent with the main text. First, these dimensions are reference-free and nearly saturated: all accelerated configurations score within 0.03 of full sampling on every dimension and within 0.005 of it in the eleven-dimension mean, and no schedule is uniformly best. PACE obtains the highest mean at NFE=26 (0.668) and the best or second-best value on most dimensions, but it is



Figure A5: Additional qualitative comparison on Wan2.1-1.3B at NFE=26 (middle frame shown). From top to bottom: a running dog, a surfer, a steam train, and a waterfall.

marginally lower than SeaCache on aesthetic and imaging quality at NFE=18, and marginally higher on background consistency and appearance style. These metrics measure generic video quality rather than fidelity to the full-computation reference, which is why the main text relies on PSNR, LPIPS, and SSIM for comparison. Second, the dimension most sensitive to scheduling is *dynamic degree*: uniform and adaptive schedules reduce motion dynamics relative to full sampling (0.653–0.667 vs. 0.681), whereas PACE restores the full-sampling value (0.681) at both budgets and with both cache operators.

Per-prompt robustness. The video PSNR margins are consistent at the per-prompt level rather than driven by outliers: on Wan2.1-1.3B at NFE=26, PACE attains higher PSNR than SeaCache on 98.4% of the 944 VBench prompts, and a paired bootstrap over prompts (10^4 resamples) yields a 95% confidence interval of $[13.1, 14.1]$ dB for the mean improvement; the corresponding win rates at NFE=18 and on HunyuanVideo-13B all exceed 95%.

K Additional Qualitative Results

Figure A5 provides additional motion-centric comparisons on Wan2.1-1.3B at NFE=26 beyond the structure-oriented VBench prompts of the main text. The pattern of the main-text qualitative figures persists: uniform HiCache and SeaCache alter scene composition, color balance, or object identity within a single early cache interval, whereas PACE remains nearly indistinguishable from the 50-step reference on every prompt.