BoundaryDelta: Spline-Integrated Average-Velocity Prediction for Efficient Diffusion Sampling

Anonymous authors
Paper under double-blind review

Abstract

Cache-based acceleration for diffusion and flow-matching models typically reuses or extrapolates an instantaneous prediction, even though each solver step depends on velocity accumulated over an interval. We introduce BoundaryDelta, a training-free framework that instead estimates the interval-average velocity. At each cached step, BoundaryDelta constructs a causal, boundary-constrained cubic Hermite velocity surrogate from completed model evaluations, then integrates it analytically to advance without an additional transformer call. Because approximation quality depends on where genuine evaluations occur, we couple this controller with budget-specific anchor schedules found offline. Across Qwen-Image and Stable Diffusion 3.5 Medium, BoundaryDelta delivers the strongest overall FULL-reference fidelity among equal-NFE baselines across aggressive and moderate call budgets. In a separate image-conditioned 3D study, BoundaryDelta retains FULL-50-level held-out ground-truth geometry for Hunyuan3D-Shape-v2.1 Small using 17 of 50 transformer evaluations. On Qwen-Image, BoundaryDelta reduces transformer evaluations by 50–80% at the primary 10-, 17-, and 25-call settings and achieves up to $4.72\times$ paired steady-state speedup.

1 Introduction

Diffusion and flow-matching models generate structured outputs by transforming noise through a sequence of progressively refined states (Ho et al., 2020; Lipman et al., 2023). Modern systems often use diffusion transformers (DiTs), whose capacity and scalable token processing have made them powerful backbones for high-resolution generation (Peebles & Xie, 2023; Wu et al., 2025). Sampling, however, is inherently iterative: the model is invoked at successive times to predict the velocity or denoising direction used by a numerical solver. The computation of a large transformer therefore accumulates over the sampling trajectory, and its sequential nature limits both latency and throughput. Reducing this repeated model cost without sacrificing the quality of the generated trajectory is consequently a central problem for efficient deployment.

Several lines of work reduce this cost in different ways. Higher-order or problem-specific solvers seek accurate trajectories with fewer numerical steps (Lu et al., 2022), while progressive distillation and consistency training learn models that can generate in a much shorter sequence (Salimans & Ho, 2022; Song et al., 2023). Quantization and pruning instead make each network evaluation cheaper by reducing numerical precision or model structure (Li et al., 2023; Fang et al., 2023). These approaches have enabled substantial acceleration, but they can require retraining, calibration, changes to the sampling procedure, or a different balance between per-step cost and the number of sequential steps. This motivates training-free methods that leave the model weights and solver grid intact while avoiding computation that is redundant along a single sampling trajectory.

Caching is attractive in this setting because model predictions at nearby sampling states often change smoothly enough for earlier outputs to remain useful. Existing methods reuse intermediate features, decide adaptively when a refresh is needed, or forecast a future feature from recent history (Ma et al., 2024; Liu et al., 2025a; Ma et al., 2025; Liu et al., 2025c); Cache-DiT provides block-level residual reuse in a broader inference system (Cache-DiT). Yet a numerical update is determined by the velocity accumulated across an interval, whereas direct reuse and point forecasting naturally provide

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.

information associated with one location on the trajectory. MeanCache highlights the importance of average velocity and estimates it with cached Jacobian–vector products (Gao et al., 2026a). A second issue arises from the placement of full model evaluations: two schedules can contain the same number of evaluations while leaving very different gaps between them. Accurate caching therefore depends both on what is predicted during a skipped step and on where the full evaluations that support those predictions are placed.

We introduce BoundaryDelta, a training-free method that addresses these two choices together. At a cached step, recent full evaluations and the current latent state define the boundary information for a local Hermite velocity interpolant. Analytically integrating this interpolant estimates the average velocity over the current solver interval, so the sampler can advance without another transformer evaluation. The construction is causal: it uses only full evaluations completed earlier in the sampling trajectory. Because the reliability of this local approximation changes with the distances between anchors, we select a separate fixed-size anchor schedule for each call budget using an offline self-adjusting search. The resulting method couples an interval-aware prediction rule with the evaluation locations that determine the intervals it must approximate. The same principle applies when the evolving state decodes to a 3D shape rather than an image: the solver still advances through intervals whose average model output must be estimated.

This work makes four contributions:

- We formulate cached flow updates around the interval-average velocity accumulated by a solver step, directly connecting continuous flow dynamics to training-free reuse.
- We derive a causal BoundaryDelta update by integrating a boundary-constrained Hermite interpolant; its fixed weights follow analytically rather than from training or tuning.
- We develop a reproducible fixed-call search that exchanges anchor locations while preserving the number of transformer evaluations, jointly adapting the prediction rule's supporting schedule.
- We show consistent fidelity gains over matched equal-NFE baselines on two text-to-image backbones across aggressive and moderate call budgets, and transfer the same controller and schedule principle to image-conditioned 3D shape generation.

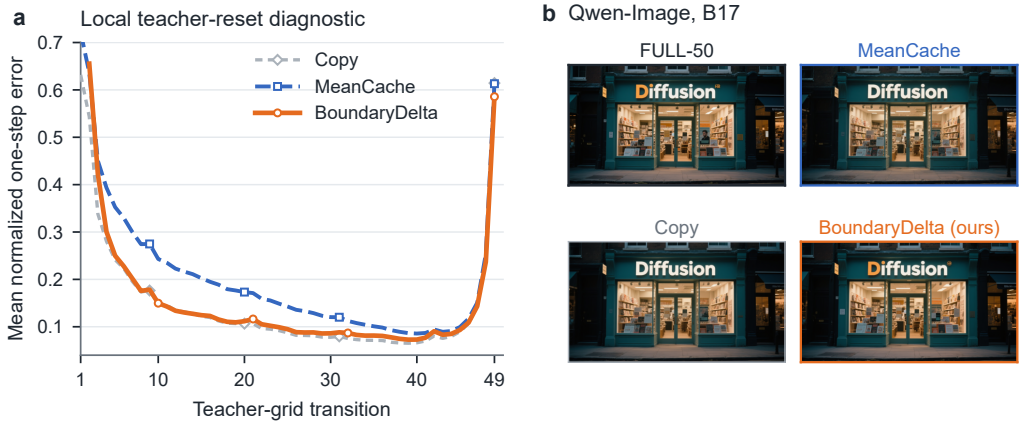


Figure 1: (a) Teacher-reset one-step normalized errors of average-velocity estimates for Copy, MeanCache, and BoundaryDelta; lower is better, and this panel isolates local prediction error rather than a free-running recurrent rollout. (b) FULL-50 and three exact-call Qwen-Image methods at B17 under full sampling.

2 Related Work

Sampling and model-level acceleration. Efficient generation can be approached either by shortening the sampling trajectory or by reducing the cost of each model evaluation. Numerical solvers such as DPM-Solver exploit the structure of diffusion dynamics to improve accuracy at small step counts (Lu et al., 2022). Distillation and consistency objectives move more of the work into training, producing models designed for few-step generation (Salimans & Ho, 2022; Song et al., 2023). Quantization and pruning retain the iterative procedure but lower the arithmetic or parameter cost of

its network calls (Li et al., 2023; Fang et al., 2023). These strategies address the repeated-evaluation bottleneck from different directions, often with changes to training, model representation, or discretization. Training-free caching instead asks whether computation already performed during the same trajectory can replace selected future evaluations.

Caching, reuse, and forecasting. Caching methods exploit temporal redundancy at several levels of the denoising network. DeepCache reuses high-level U-Net features while refreshing lower-level computation (Ma et al., 2024); TeaCache and MagCache use trajectory-dependent signals to decide when reuse remains reliable (Liu et al., 2025a; Ma et al., 2025); and Cache-DiT exposes block-level residual reuse in a broader inference system (Cache-DiT). Forecasting methods enrich the replacement value rather than copying it directly: TaylorSeer uses finite-difference extrapolation, while HiCache, FreqCa, TC-Padé, and Spectrum study Hermite, frequency-aware, rational, or spectral descriptions of feature evolution (Liu et al., 2025c; Feng et al., 2025; Liu et al., 2025b; He et al., 2026; Han et al., 2026). MeanCache takes a complementary average-velocity view, using cached Jacobian–vector products to estimate the quantity accumulated by a flow step (Gao et al., 2026a). Together, these works show that the replacement used at a skipped evaluation matters. BoundaryDelta couples an interval-integrated velocity prediction with the locations of the evaluations that support it.

Evaluation-step and anchor-schedule selection. Once model evaluations are limited, their placement becomes a discrete design problem rather than a consequence of the original solver grid. MeanCache constructs an offline multigraph whose path determines where evaluations occur (Gao et al., 2026a), and LeMiCa selects an exact-budget path through a lexicographic minimax objective (Gao et al., 2025). BuDCache searches masks with a fixed number of evaluations using simulated annealing and local refinement, while DPCache uses calibrated path costs and dynamic programming (Lei et al., 2026; Cui et al., 2026). OTCache further studies how schedules at selected budgets can guide schedules at intermediate budgets through a transport-based construction (Gao et al., 2026b). These approaches demonstrate that equally sized schedules need not be equally effective, but the best placement can depend on the approximation used between evaluations. Our setting therefore treats interval-average prediction and anchor placement as one coupled sampling problem.

3 Problem Formulation

Flow Matching. Let $x(t)$ denote the latent state of a conditional generative flow and let $v_\theta(x(t), t, y)$ be its learned velocity field, with y denoting the conditioning input (Lipman et al., 2023; Karras et al., 2022). Sampling follows this field from noise toward a generated output along a decreasing numerical time grid,

$$\frac{\mathrm{d}x}{\mathrm{d}t} = v_\theta(x(t), t, y), \qquad t_0 > \cdots > t_N, \qquad \Delta t_n := t_{n+1} - t_n < 0. \tag{1}$$

The differential equation describes the continuous path, while a sampler advances through discrete states. With $V_n := v_\theta(x_n, t_n, y)$, an Euler-like step uses one model prediction as its velocity estimate:

$$x_{n+1} = x_n + \Delta t_n V_n. \tag{2}$$

Thus, a conventional N-step sampler evaluates the transformer once at each step before updating the latent.

Caching for Diffusion Models. A caching method replaces some of these transformer calls with a prediction computed from information already available along the trajectory. Let S be the set of steps that perform a full evaluation and let $\widehat{v}_n^{\text{cache}}$ be a generic cached prediction. The value applied by the sampler is

$$w_n = \begin{cases} V_n, & n \in S \quad (\textsc{Full}), \\ \widehat{v}_n^{\text{cache}}, & n \notin S \quad (\textsc{Cache}), \end{cases} \qquad x_{n+1} = x_n + \Delta t_n w_n. \tag{3}$$

At a cached step, the sampler may reuse a stored value or form a new prediction from its history, but it does not invoke the transformer again. With a budget of B full evaluations, the schedule selects exactly B transition indices and includes the two startup evaluations required below:

$$S \subseteq \{0, \ldots, N-1\}, \qquad |S| = B, \qquad \{0, 1\} \subseteq S. \tag{4}$$

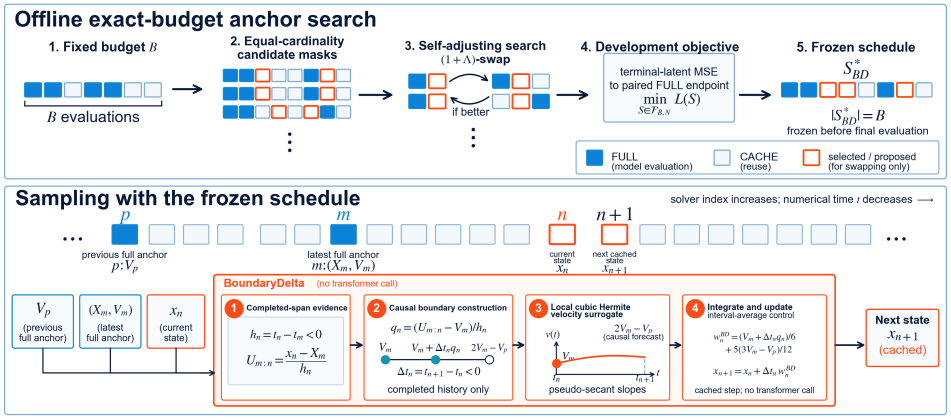


Figure 2: Top: an offline exact-budget swap search freezes the full-evaluation schedule before final evaluation. Bottom: two completed full evaluations and the current latent define a local Hermite interpolant whose integral supplies one cached-step average-velocity control. Equal-budget schedules can leave different cached spans, motivating controller-specific anchor placement.

The eligible indices are the N solver steps $0, \ldots, N-1$. The startup evaluations supply the completed model predictions needed by the causal construction developed below.

We use FULL-N for the N-step numerical reference that calls the transformer at every solver step. A cached run follows the same N-step grid but makes full transformer evaluations at only the B selected steps; the other steps use Eq. (3). It therefore removes a fraction $1 - B/N$ of the full model calls.

From Instantaneous to Average Velocity. The value required by a solver interval is more naturally seen by integrating the flow equation. Define its exact average velocity as

$$\bar{V}^*_n = \frac{1}{\Delta t_n} \int_{t_n}^{t_{n+1}} v_\theta(x(t), t, y)\, \mathrm{d}t, \qquad x(t_{n+1}) = x(t_n) + \Delta t_n \bar{V}^*_n. \tag{5}$$

This identity shows that the state change depends on velocity over the whole interval, not only at one endpoint. Reusing a previous prediction can work when the trajectory varies slowly, but it cannot represent how that velocity changes across a longer skipped span. Full model evaluations provide reliable points on the sampled trajectory; we refer to these points as anchors. Assuming that velocity evolves smoothly enough over the local span between recent anchors, we estimate the interval average from completed evaluations rather than hold the latest anchor velocity constant throughout the next cached interval. Section 4 turns this idea into a causal cached-step update.

4 BoundaryDelta: Spline-Integrated Interval Prediction

BoundaryDelta couples an interval-average cached-step controller with an offline exact-budget anchor schedule (Figure 2). This section derives the controller. At each cached step, recent full evaluations and the current latent define a local cubic Hermite velocity surrogate. Its integral estimates the interval average required by Eq. (5), rather than an isolated future feature; Section 5 selects the full-evaluation locations that determine the spans over which the controller operates.

A local Hermite velocity interpolant. For the current interval $[t_n, t_{n+1}]$, introduce the normalized coordinate $\tau = (t - t_n)/\Delta t_n \in [0, 1]$. The interpolant is determined by a left endpoint value $\widehat{V}^L_n$, an estimated right endpoint $\widehat{V}_{n+1|n}$, and slopes $a_{L,n}$ and $a_{R,n}$ with respect to numerical time t. Because $\mathrm{d}/\mathrm{d}\tau = \Delta t_n\, \mathrm{d}/\mathrm{d}t$, each time-domain slope is multiplied by the signed step width when it enters the normalized Hermite basis. The resulting segment is (de Boor, 2001)

$$\begin{aligned}\widehat{V}_n(\tau) = {} & h_{00}(\tau)\widehat{V}^L_n + h_{10}(\tau)\Delta t_n a_{L,n} \\ & + h_{01}(\tau)\widehat{V}_{n+1|n} + h_{11}(\tau)\Delta t_n a_{R,n}.\end{aligned} \tag{6}$$

The four cubic Hermite basis functions are listed in Appendix A.2. The polynomial meets its assigned endpoint values and slopes exactly. Treating this constructed segment as a useful approximation to the unknown velocity path is the local modeling assumption.

Causal boundary and slope construction. To construct the interpolant from information available before the cached update, let $p < m < n$ denote the two most recent full evaluations, with stored latent X_m and velocities V_p and V_m. The completed displacement from the latest anchor to the current state defines the observed average rate $U_{m:n}$, whose deviation from V_m defines the time-rate q_n:

$$p < m < n, \qquad h_n := t_n - t_m < 0, \qquad U_{m:n} := \frac{x_n - X_m}{h_n}, \qquad q_n := \frac{U_{m:n} - V_m}{h_n}. \tag{7}$$

Here $U_{m:n}$ is the average velocity already observed between the latest anchor and the current latent. Dividing $U_{m:n} - V_m$ by the signed elapsed time converts the velocity difference into a rate of change. Advancing that rate by the next signed step width gives $V_m + \Delta t_n q_n$, a provisional average for the cached update rather than a new transformer output.

BoundaryDelta uses this provisional average together with a persistent left boundary and a causal right-boundary forecast:

$$\widehat{V}_n^L = V_m, \qquad \widetilde{U}_n = V_m + \Delta t_n q_n, \qquad \widehat{V}_{n+1|n} = 2V_m - V_p. \tag{8}$$

The left value retains the latest completed model prediction, while the right value extrapolates the velocity change between the two completed anchors. Thus X_m, V_p, V_m, the recurrent state x_n, and the time grid determine all three quantities before the cached update begins.

To turn these values into the endpoint slopes required above, we use two pseudo-secants that share the provisional interval value:

$$a_{L,n} = \frac{\widetilde{U}_n - \widehat{V}_n^L}{\Delta t_n}, \qquad a_{R,n} = \frac{\widehat{V}_{n+1|n} - \widetilde{U}_n}{\Delta t_n}. \tag{9}$$

These are causal slope proxies, not independently observed derivatives of the unknown future path. Multiplication by Δt_n in Eq. (6) recovers the corresponding velocity differences in the normalized τ coordinate.

Analytic interval average. The Hermite form can be averaged without sampling the curve: integrating the four basis functions in Appendix A.2 over the unit interval yields $\int_0^1 h_{00}(s)\,\mathrm{d}s = \frac{1}{2}$, $\int_0^1 h_{10}(s)\,\mathrm{d}s = \frac{1}{12}$, $\int_0^1 h_{01}(s)\,\mathrm{d}s = \frac{1}{2}$, and $\int_0^1 h_{11}(s)\,\mathrm{d}s = -\frac{1}{12}$. Applying these weights to Eq. (6), the constructed segment has exact average $\widehat{\widetilde{V}}_n = \frac{1}{2}(\widehat{V}_n^L + \widehat{V}_{n+1|n}) + \frac{\Delta t_n}{12}(a_{L,n} - a_{R,n})$, with the boundary-slope difference correcting the endpoint average for curvature.

Substituting the two pseudo-secants from Eq. (9) and collecting terms gives

$$\begin{aligned} w_n^{\mathrm{BD}} &= \frac{1}{2}(\widehat{V}_n^L + \widehat{V}_{n+1|n}) + \frac{1}{12}\Big[(\widetilde{U}_n - \widehat{V}_n^L) - (\widehat{V}_{n+1|n} - \widetilde{U}_n)\Big] \\ &= \frac{1}{6}\widetilde{U}_n + \frac{5}{12}(\widehat{V}_n^L + \widehat{V}_{n+1|n}). \end{aligned} \tag{10}$$

The coefficients $1/6$ and $5/12$ therefore follow exactly from integrating the constructed Hermite segment; the approximation lies in using that segment as a surrogate for the unknown velocity path.

Executable cached update. Combining the already available quantities in Eq. (8) with Eq. (10) produces

$$\boxed{w_n^{\mathrm{BD}} = \frac{1}{6}\left(V_m + \Delta t_n q_n\right) + \frac{5}{12}\left(3V_m - V_p\right), \qquad x_{n+1} = x_n + \Delta t_n w_n^{\mathrm{BD}}.} \tag{11}$$

In exact arithmetic, $w_n^{\mathrm{BD}} = \alpha_n V_m + (1 - \alpha_n)V_p$, with α_n fixed by the grid, latest full anchor, and cache depth (Appendix A.4). At a cached step, this rule uses the current state and two completed full evaluations, adding no transformer call. The first two full steps guarantee that the required anchors exist before caching starts.

BoundaryDelta keeps transformer-evaluated anchor velocities distinct from cached interval controls. The released MeanCache recurrence appends each cached control to its rolling velocity history, allowing later cached steps to reuse earlier predictions recursively. BoundaryDelta instead retains two completed model evaluations as its anchors: earlier cached controls affect the recurrent state, but never replace either anchor (Appendix A.5).

The quality of Eq. (11) depends on the gaps between its anchor evaluations. A schedule with tightly placed anchors can support easy local predictions in one part of the trajectory while leaving a long, difficult span elsewhere. The next section searches for anchor locations that suit BoundaryDelta under a fixed model-call budget.

5 Exact-Budget Anchor Optimization

Two schedules can use the same number of transformer evaluations yet leave very different gaps between anchors. Because BoundaryDelta predicts each cached interval from recent full evaluations, these gaps determine both recurrence depth and endpoint geometry. For an N-transition sampler and a budget of B full evaluations, we define

$$\mathcal{F}_{B,N} := \{S \subseteq \{0,\dots,N-1\} : |S| = B,\ \{0,1\} \subseteq S\}, \qquad 2 \le B \le N. \tag{12}$$

Every candidate therefore selects exactly B transformer calls and keeps the two startup anchors required by Eq. (11).

For a fixed set $\mathcal{D}_{\text{dev}}$ of development prompt–seed pairs, we score a schedule with the development-only terminal latent error relative to its Full-N endpoint:

$$\mathcal{L}(S) := \widehat{\mathbb{E}}_{(y,r)\in\mathcal{D}_{\text{dev}}}\left[\text{MSE}\left(x_N^{\text{BD},S}(y,r), x_N^{\text{FULL}}(y,r)\right)\right]. \tag{13}$$

Among the candidates evaluated during development, this loss was an empirically useful search signal and was broadly consistent with the resulting FULL-reference fidelity trends; final evaluation remains held out.

Self-adjusting $(1+\Lambda)$-swap search. The search begins from feasible schedules and proposes nearby alternatives while always retaining exactly B anchors. From the current schedule S, it removes K selected mutable anchors R and inserts the same number of previously unselected positions A:

$$R \subseteq S \setminus \{0,1\}, \qquad A \subseteq \{2,\dots,N-1\} \setminus S, \qquad S' = (S \setminus R) \cup A, \qquad |R| = |A| = K. \tag{14}$$

The equal exchange preserves both the call budget and the startup anchors. When $B = 2$ or $B = N$, the feasible set contains one schedule and no search is needed. For $2 < B < N$, the search controls the size of the exchange with a radius ℓ. Writing $K_{\max} = \min(B-2, N-B)$, it samples

$$K \sim \text{Binomial}(K_{\max}, p_\ell) \mid K \ge 1, \qquad p_\ell = \min\left(\frac{\ell}{K_{\max}}, 0.99\right). \tag{15}$$

This positive truncated-binomial law favors small exchanges when ℓ is small and progressively broader exchanges as ℓ grows.

Each batch contains up to $\Lambda = 32$ different proposals. A candidate with lower loss replaces the current schedule,

$$\mathcal{L}(S') < \mathcal{L}(S) \quad \Longrightarrow \quad S \leftarrow S'.$$

After an improving batch, the radius contracts so that the next proposals refine the new schedule; otherwise it expands to search farther away. This multiplicative update is inspired by one-fifth success-based adaptation (Doerr & Doerr, 2015), but here it controls a discrete swap radius: the radius contracts after success and expands after failure. With the fixed factor $F = 1.5$, the update is

$$\ell_{t+1} = \begin{cases} \max(1, \ell_t/F), & \text{if the batch improves } \mathcal{L}, \\ \min(K_{\max}, \ell_t F^{1/4}), & \text{otherwise.} \end{cases} \tag{16}$$

After five consecutive batches without improvement, the search restarts from a new feasible schedule. It evaluates eight initial schedules and stops after at most $Q = 200$ distinct schedule evaluations. The selected schedule is then frozen before final evaluation; deployment uses only this fixed mask and the BoundaryDelta controller. Appendix B gives the exact archive, restart, and deterministic tie-handling procedure.

6 EXPERIMENTS

Models and references. We evaluate Qwen-Image (Wu et al., 2025) at 1664×928 and Stable Diffusion 3.5 Medium (Esser et al., 2024; Stability AI, 2024) at 1024×1024 against model-specific 50-step FULL-50 references; paired images share their prompt and initial noise. Both image studies use all 200 DrawBench prompts (Saharia et al., 2022) with generation seed 42; the complete DrawBench suite is excluded from search and selection. Qwen uses B6–B10, B17, and B25, with B10/B17/B25 primary; SD3.5 uses B10/B17. We also test the 2048-token Small Hunyuan3D-Shape-v2.1 (Team Hunyuan3D et al., 2025) on 50 held-out HY3D-Bench objects (Team Hunyuan3D et al., 2026); five disjoint objects are used only for search.

Baselines, compute, and metrics. Qwen exact-call baselines are MeanCache (Gao et al., 2026a), Copy (at every cached step, it reuses the velocity V_m from the most recent full model evaluation, so $w_n^{\text{Copy}} = V_m$), reduced-step Qwen, and LeMiCa (Gao et al., 2025); adaptive baselines are MagCache (Ma et al., 2025), two DBCache settings (Cache-DiT), DBCache with fourth-order TaylorSeer (Liu et al., 2025c), TeaCache (Liu et al., 2025a), and fourth-order TaylorSeer. The SD3.5 exact-call panel compares BoundaryDelta with Copy, MeanCache, and reduced-step SD3.5; Appendix D reports its adaptive target-regime panel. Exact-call methods report NFE; adaptive methods report realized full-evaluation equivalents (FEC) from measured FLOPs. Qwen B10/B17/B25 speedups are paired; SD3.5 ratios are descriptive across separate timing cohorts, while Hunyuan3D timing uses raw transformer-denoising time. We report paired FULL-reference MSE/L1, LPIPS-Alex (Zhang et al., 2018), SSIM (Wang et al., 2004), and PSNR, plus separate reference-free CLIP-G with OpenCLIP (Radford et al., 2021; Cherti et al., 2023), ImageReward (Xu et al., 2023), and HPSv2.1 (Wu et al., 2023). Hunyuan3D uses ground-truth geometry measures; Appendix C defines preprocessing and compute accounting.

7 RESULTS

At B17, BoundaryDelta lowers terminal MSE and LPIPS by 26.6% and 26.0% relative to MeanCache, uses 66% fewer calls than FULL-50, and reaches $2.841\times$ paired steady-state speedup (Table 1). At B10, the reductions are 46.8%/27.6%, with 80% fewer calls and $4.723\times$ paired speedup. At B25, they are 48.9%/53.8% with 50% fewer calls and $1.977\times$ paired speedup from 168 valid timing pairs; Appendix D reports the full panel.

Table 1: Qwen-Image B10/B17 results. Eval. is exact NFE for exact-call rows and realized FEC for adaptive rows; Lat./Spd. are median seconds/paired steady-state acceleration. Bold/underline mark the best/second-best quality values among the five equal-NFE rows; BoundaryDelta is shaded.

B	Method	Eval.	Lat.↓	Spd.↑	MSE↓	L1↓	LPIPS↓	SSIM↑	PSNR↑	CLIP-G↑	IR↑	HPS↑
	BOUNDARYDELTA (OURS)	10	29.543	4.723×	**0.0284**	**0.1093**	**0.1948**	**0.8294**	**21.65**	0.33436	**1.2075**	**0.2962**
	MeanCache	10	29.588	4.752×	0.0533	0.1471	0.2691	0.7717	18.61	0.33449	1.2049	0.2960
	Copy	10	29.330	4.755×	0.0509	0.1416	0.2702	0.7793	18.73	0.33483	1.1895	0.2946
	Reduced-step Qwen	10	29.310	4.743×	0.2235	0.3647	0.5954	0.5669	12.88	0.32953	0.7645	0.2468
	LeMiCa	10	29.618	4.716×	0.0555	0.1505	0.2887	0.7684	18.34	**0.33514**	1.1801	0.2935
B10	MagCache	15.002	42.880	3.223×	0.0103	0.0625	0.1095	0.9119	26.35	0.33547	1.1931	0.2965
	DBCache r=0.6	14.924	42.957	3.218×	0.0207	0.0983	0.1747	0.8492	22.88	0.33514	1.1823	0.2922
	DBCache r=1.5	12.202	35.313	3.934×	0.0866	0.2325	0.4724	0.6306	15.83	0.31875	0.7316	0.2483
	DBCache+TaylorSeer r=1.5 O4	12.202	35.352	3.903×	8.4319	2.2105	0.9090	0.1527	3.83	0.15446	-2.1200	0.1683
	TeaCache	9.002	26.486	5.239×	0.1877	0.3225	0.5397	0.5737	13.01	0.32655	0.9510	0.2746
	TaylorSeer O4	10.002	29.477	4.696×	0.2699	0.4083	0.7657	0.4152	10.94	0.22715	-0.8681	0.1986
	BOUNDARYDELTA (OURS)	17	48.972	2.841×	**0.0061**	**0.0419**	**0.0599**	**0.9445**	**29.24**	0.33456	1.2130	**0.2997**
	MeanCache	17	49.010	2.860×	0.0084	0.0542	0.0809	0.9287	27.36	0.33287	**1.2245**	0.2990
	Copy	17	48.542	2.864×	0.0080	0.0495	0.0815	0.9309	27.44	0.33461	1.2009	0.2981
	Reduced-step Qwen	17	48.653	2.848×	0.1166	0.2338	0.4242	0.6645	14.91	**0.33718**	1.1135	0.2874
	LeMiCa	17	48.971	2.842×	0.0079	0.0559	0.0962	0.9253	27.78	0.33521	1.1917	0.2967
B17	MagCache	17.002	48.367	2.858×	0.0075	0.0506	0.0828	0.9332	27.93	0.33522	1.2013	0.2977
	DBCache r=0.6	17.206	49.332	2.805×	0.0161	0.0785	0.1262	0.8852	24.17	0.33471	1.2173	0.2978
	DBCache r=1.5	15.802	45.125	3.077×	0.0202	0.0916	0.1515	0.8626	22.98	0.33497	1.2124	0.2983
	DBCache+TaylorSeer r=1.5 O4	15.802	45.096	3.061×	0.4833	0.5462	0.8177	0.2944	7.28	0.22017	-0.7364	0.2173
	TeaCache	18.002	51.222	2.708×	0.0772	0.1742	0.3176	0.7355	16.95	0.33481	1.1843	0.2964
	TaylorSeer O4	17.002	48.788	2.838×	0.1312	0.2652	0.4640	0.6334	14.66	0.32424	1.0414	0.2867

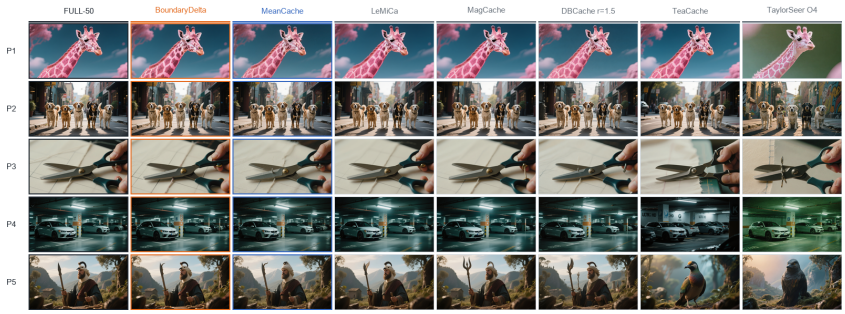


Figure 3: Paired Qwen-Image B17 outputs for five fixed DrawBench prompts. Columns show FULL-50, BoundaryDelta, and six baselines; every row shares prompt, seed, and initial latent, and the prompts were fixed without selection for BoundaryDelta advantage. Appendix D gives all eight prompts and the key.

Table 2: SD3.5 Medium exact-budget results. Eval. is exact NFE; Lat. is median wall time and Spd. is the descriptive FULL-50 median divided by the row median, not a paired speedup. Bold/underline mark the best/second-best quality values within each budget.

B	Method	Eval.	Lat.↓	Spd.↑	MSE↓	L1↓	LPIPS↓	SSIM↑	PSNR↑	CLIP-G↑	IR↑	HPS↑
B10	**BoundaryDelta (ours)**	10	4.606	4.414×	0.0676	**0.1402**	**0.1976**	**0.8197**	**21.66**	**0.33162**	**0.8064**	**0.2659**
	Copy	10	4.536	4.482×	**0.0674**	0.1472	0.2580	0.7843	21.51	0.33122	0.7228	0.2582
	MeanCache	10	4.632	4.389×	0.1766	0.3198	0.4742	0.6845	19.59	0.31758	0.4778	0.2388
	Reduced-step SD3.5	10	4.406	4.614×	0.3880	0.4379	0.4257	0.6747	14.29	0.33039	0.6710	0.2584
B17	**BoundaryDelta (ours)**	17	7.251	2.804×	**0.0156**	**0.0619**	**0.0701**	**0.9376**	**28.57**	0.33033	**0.8972**	**0.2766**
	Copy	17	7.192	2.827×	0.0209	0.0789	0.1212	0.9001	26.96	0.33166	0.8618	0.2727
	MeanCache	17	7.462	2.725×	0.0600	0.1770	0.2431	0.8369	23.70	0.32798	0.7786	0.2634
	Reduced-step SD3.5	17	7.187	2.829×	0.3109	0.3647	0.3323	0.7366	15.50	**0.33499**	0.8862	0.2742

At equal NFE, BoundaryDelta improves all five FULL-reference fidelity metrics over MeanCache, Copy, reduced-step Qwen, and LeMiCa at B10/B17/B25. At B10, its ten evaluations also improve all eight point estimates over TeaCache (9.002 FEC) and TaylorSeer (10.002 FEC). In the mixed-compute target-budget roster, it leads all five B17/B25 fidelity metrics across 11 methods. Figure 3 shows paired B17 outputs with shared prompt, seed, and initial latent.

Stable Diffusion 3.5 Medium tests whether the exact-budget gains transfer to a second image backbone at B10/B17 (Table 2). BoundaryDelta leads seven of eight metrics at both budgets and all five B17 FULL-reference fidelity metrics: MSE 0.0156, L1 0.0619, LPIPS 0.0701, SSIM 0.9376, and PSNR 28.57. Reduced-step SD3.5 has the highest CLIP-G, while BoundaryDelta leads ImageReward and HPSv2.1.

Image-conditioned 3D transfer. To test transfer beyond image generation, we apply the unchanged BoundaryDelta update to Hunyuan3D-Shape-v2.1 Small (Team Hunyuan3D et al., 2025) on 50 held-out HY3D-Bench objects (Team Hunyuan3D et al., 2026). At B17 it uses 17 evaluations (66% fewer calls) while keeping geometry close to FULL-50. Raw transformer-denoising time falls from 5.507 to 1.854 seconds (2.970×). Against uniform B17, the searched schedule lowers terminal MSE by 14.9% and improves all three reported F-score/normal measures. Figure 4 shows fixed paired cases under shared conditioning and renderer view; Appendix Table 13 gives the paired values.

Anchor-schedule optimization. The schedule analysis separates offline selection from final evaluation: each non-nested, budget-specific deployed mask is frozen from development-only terminal-MSE search, without final DrawBench or held-out 3D data (Figure 5; Appendix B).

Prediction-rule and schedule ablation. The crossed ablation asks how the prediction rule and anchor schedule contribute under a fixed budget, and the answer is budget dependent (Appendix Table 11). At B10, MeanCache's rule wins under both schedules, but full BoundaryDelta beats released MeanCache on all five fidelity point estimates. With S^*_{BD}, BoundaryDelta leads all five metrics at B17 (Table 3) and four at B25, tying displayed MSE. Relative to the uniform-index control, the optimized schedule improves all five metrics at every budget; controller effects are therefore complementary to schedule placement but budget dependent. The full rule is most consistent across slot variants; Appendix D gives their definitions and results.

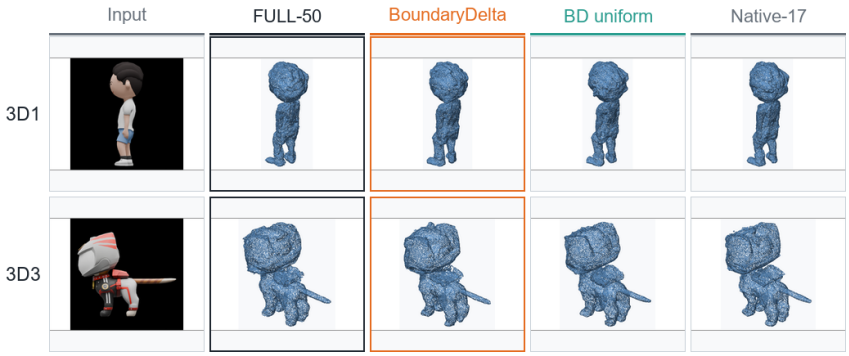


Figure 4: Image-conditioned 3D transfer at B17 on the fixed held-out 3D1 and 3D3 cases. Columns share the conditioning image and renderer view; Appendix D gives the full table, protocol, and additional cases including 3D2.

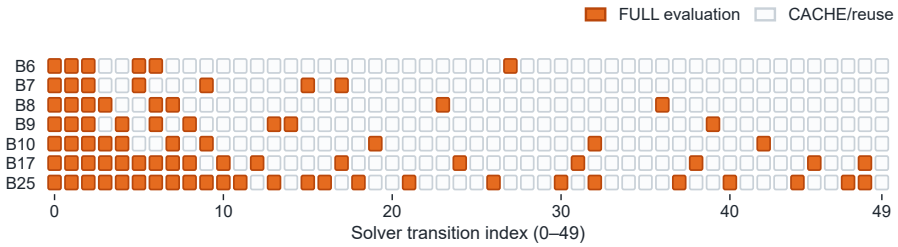


Figure 5: Qwen-Image schedules (orange: FULL; light: CACHE/reuse); Appendix B gives search convergence.

Table 3: B17 prediction-rule/schedule ablation. S_{MC}, S^{*}_{BD}, and S_{uniform} denote the MeanCache, optimized BoundaryDelta, and uniform-index schedules; bold/underline mark the best/second-best values.

B	Metric	MeanCache $+S_{\mathrm{MC}}$	MeanCache $+S^{*}_{\mathrm{BD}}$	BoundaryDelta $+S_{\mathrm{MC}}$	BoundaryDelta $+S^{*}_{\mathrm{BD}}$	BoundaryDelta $+S_{\mathrm{uniform}}$
B17	MSE↓	0.0084	<u>0.0062</u>	0.0085	**0.0061**	0.0791
B17	L1↓	0.0542	<u>0.0458</u>	0.0499	**0.0419**	0.1705
B17	LPIPS↓	0.0809	<u>0.0640</u>	0.0769	**0.0599**	0.2939
B17	SSIM↑	0.9287	<u>0.9427</u>	0.9301	**0.9445**	0.7466
B17	PSNR↑	27.36	<u>28.96</u>	27.40	**29.24**	17.40

8 Discussion and Conclusion

BoundaryDelta reframes diffusion caching as controlling a continuous trajectory from sparse, expensive observations. Skipping a transformer evaluation does not remove the solver interval that must still be traversed; what is missing is the model output averaged over that interval. A causal spline supplies this interval estimate, while anchor placement determines where the trajectory is observed. Their complementary effects support a broader design principle: the prediction rule and the computation schedule should be designed together, rather than treating caching as passive reuse at predetermined steps.

Across the evaluated image generators, this perspective preserves the reference trajectory more faithfully than matched caching alternatives while reducing transformer evaluations; transfer to image-conditioned 3D provides initial evidence beyond a single output modality. The evidence nevertheless remains bounded to the tested systems. BoundaryDelta relies on sufficient local regularity between anchors, and its fixed schedules remain dependent on the development distribution used to select them. Transferable schedules, online adaptation, and richer interval models are therefore important directions. More broadly, BoundaryDelta suggests viewing efficient generative inference as sparse numerical control: spend expensive model evaluations where they are most useful for maintaining the trajectory, and use structure, rather than repetition, to bridge the intervals between them.

AI USE STATEMENT

AI-assisted tools supported literature organization, drafting/language refinement, code/LaTeX checks, and consistency audits. Authors retained responsibility for scientific decisions, experimental design, result selection, source verification, and the final manuscript; citations and numbers were checked against primary sources/project data.

REPRODUCIBILITY STATEMENT

Sections 4–6 define the method, search, and NFE/FEC accounting; Appendices A–D provide derivations, search details, experimental settings, and complete results. Qwen-Image and Stable Diffusion 3.5 Medium use all 200 DrawBench prompts with paired initial latents, model-specific FULL-50 references, and seed 42; Hunyuan3D-Shape-v2.1 Small uses 50 held-out objects, disjoint search objects, and the same seed. We do not claim that the project code or supplementary package has already been publicly released.

REFERENCES

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2818–2829, 2023. doi: 10.1109/CVPR52729.2023.00276. URL https://openaccess.thecvf.com/content/CVPR2023/html/Cherti_Reproducible_Scaling_Laws_for_Contrastive_Language-Image_Learning_CVPR_2023_paper.html. Official repository: https://github.com/mlfoundations/open_clip; authoritative ViT-g-14 checkpoint record: https://huggingface.co/laion/CLIP-ViT-g-14-laion2B-s34B-b88K.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 43632–43642, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Cui_Denoising_as_Path_Planning_Training-Free_Acceleration_of_Diffusion_Models_with_CVPR_2026_paper.html.

Carl de Boor. *A Practical Guide to Splines*, volume 27 of *Applied Mathematical Sciences*. Springer, New York, revised edition, 2001. ISBN 978-0-387-95366-3. URL https://link.springer.com/book/9780387953663.

DefTruth, vipshop.com, etc. Cache-DiT: A pytorch-native inference engine with cache, parallelism, quantization and cpu offload for DiTs. Open-source software, 2025. URL https://github.com/vipshop/cache-dit. Official repository and design documentation; the project's own collective author field does not enumerate every individual contributor.

Benjamin Doerr and Carola Doerr. Optimal parameter choices through self-adjustment: Applying the 1/5-th rule in discrete settings. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO '15, pp. 1335–1342. ACM, 2015. doi: 10.1145/2739480.2754684. URL https://doi.org/10.1145/2739480.2754684.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR, 2024. URL https://proceedings.mlr.press/v235/esser24a.html.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Structural pruning for diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 16716–16728, 2023. doi: 10.52202/075280-0731. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/35c1d69d23bb5dd6b9abcd68be005d5c-Abstract-Conference.html.

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. *arXiv preprint arXiv:2508.16984*, 2025. doi: 10.48550/arXiv.2508.16984. URL https://arxiv.org/abs/2508.16984. Current record: arXiv:2508.16984v2, 26 January 2026; official repository: https://github.com/fenglang918/HiCache.

Huanlin Gao, Ping Chen, Fuyuan Shi, Chao Tan, Zhaoxiang Liu, Fang Zhao, Kai Wang, and Shiguo Lian. LeMiCa: Lexicographic minimax path caching for efficient diffusion-based video generation. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.48550/arXiv.2511.00090. URL https://arxiv.org/abs/2511.00090. NeurIPS 2025 Spotlight; OpenReview forum QIXdI207nq; the DOI is the arXiv-issued DataCite DOI.

Huanlin Gao, Ping Chen, Fuyuan Shi, Ruijia Wu, Yantao Li, Qiang Hui, Yuren You, Ting Lu, Chao Tan, Shaoan Zhao, Zhaoxiang Liu, Fang Zhao, Kai Wang, and Shiguo Lian. MeanCache: From instantaneous to average velocity for accelerating flow matching inference. In *International Conference on Learning Representations*, 2026a. doi: 10.48550/arXiv.2601.19961. URL https://openreview.net/forum?id=GMCyL7Xs9R. The DOI is the arXiv-issued DataCite DOI; the paper is published at ICLR 2026.

Huanlin Gao, Fang Zhao, Qiang Hui, Fuyuan Shi, Shaoan Zhao, Yantao Li, Chao Tan, Ting Lu, Yuren You, Kai Wang, and Shiguo Lian. OTCache: Optimal transport for geometry-aware caching in diffusion models. In *European Conference on Computer Vision*, 2026b. doi: 10.48550/arXiv.2606.31026. URL https://arxiv.org/abs/2606.31026. Accepted at ECCV 2026; the DOI is the arXiv-issued DataCite DOI.

Jiaqi Han, Juntong Shi, Puheng Li, Haotian Ye, Qiushan Guo, and Stefano Ermon. Adaptive spectral feature forecasting for diffusion sampling acceleration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 43320–43330, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Han_Adaptive_Spectral_Feature_Forecasting_for_Diffusion_Sampling_Acceleration_CVPR_2026_paper.html.

Shaoxuan He, Benlei Cui, Bukun Huang, Zhizeng Ye, Yunyun Sun, Longtao Huang, Hui Xue, Yang Yang, Haiwen Hong, Jingqun Tang, and Zhou Zhao. TC-Padé: Trajectory-consistent Padé approximation for diffusion acceleration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35768–35778, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/He_TC-Pade_Trajectory-Consistent_Pade_Approximation_for_Diffusion_Acceleration_CVPR_2026_paper.html.

Jonathan Ho, Ajay N. Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 26565–26577, 2022. doi: 10.52202/068431-1926. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/a98846e9d9cc01cfb87eb694d946ce6b-Abstract.html.

Mingkun Lei, Tong Zhao, Liangyu Yuan, and Chi Zhang. Budget-constrained step-level diffusion caching. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, Seoul, South Korea, 2026. PMLR. doi: 10.48550/arXiv.2606.13496. URL https://arxiv.org/abs/2606.13496. Accepted at ICML 2026; the DOI is the arXiv-issued DataCite DOI.

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*, pp. 17535–17545, 2023. doi: 10.1109/ICCV51070.2023.01608. URL https://openaccess.thecvf.com/content/ICCV2023/html/Li_Q-Diffusion_Quantizing_Diffusion_Models_ICCV_2023_paper.html.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. doi: 10.48550/arXiv.2210.02747. URL https://openreview.net/forum?id=PqvMRDCJT9t. The DOI is the arXiv-issued DataCite DOI; the ICLR record has no proceedings DOI.

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*, pp. 7353–7363, June 2025a. doi: 10.1109/CVPR52734.2025.00689. URL https://openaccess.thecvf.com/content/CVPR2025/html/Liu_Timestep_Embedding_Tells_Its_Time_to_Cache_for_Video_Diffusion_CVPR_2025_paper.html. Official repository: https://github.com/ali-vilab/TeaCache.

Jiacheng Liu, Peiliang Cai, Qinming Zhou, Yuqi Lin, Deyang Kong, Benhao Huang, Yupei Pan, Haowen Xu, Chang Zou, Junshu Tang, Shikang Zheng, and Linfeng Zhang. FreqCa: Accelerating diffusion models via frequency-aware caching. *arXiv preprint arXiv:2510.08669*, 2025b. doi: 10.48550/arXiv.2510.08669. URL https://arxiv.org/abs/2510.08669. Current record: arXiv:2510.08669v1, 9 October 2025.

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*, pp. 15853–15863, October 2025c. doi: 10.1109/ICCV51701.2025.01471. URL https://openaccess.thecvf.com/content/ICCV2025/html/Liu_From_Reusing_to_Forecasting_Accelerating_Diffusion_Models_with_TaylorSeers_ICCV_2025_paper.html.

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*, volume 35, pp. 5775–5787, 2022. doi: 10.52202/068431-0418. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/260a14acce2a89dad36adc8eefe7c59e-Abstract-Conference.html.

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*, pp. 15762–15772, 2024. doi: 10.1109/CVPR52733.2024.01492. URL https://openaccess.thecvf.com/content/CVPR2024/html/Ma_DeepCache_Accelerating_Diffusion_Models_for_Free_CVPR_2024_paper.html.

Zehong Ma, Longhui Wei, Feng Wang, Shiliang Zhang, and Qi Tian. MagCache: Fast video generation with magnitude-aware cache. In *Advances in Neural Information Processing Systems*, volume 38, pp. 38592–38624, 2025. doi: 10.52202/085713-1153. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/311207bb626e36a8f1d3eb92aa67af22-Abstract-Conference.html. OpenReview forum KZn7TDOL4J; official repository: https://github.com/Zehong-Ma/MagCache.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023. doi: 10.1109/ICCV51070.2023.00387. URL https://openaccess.thecvf.com/content/ICCV2023/html/Peebles_Scalable_Diffusion_Models_with_Transformers_ICCV_2023_paper.html.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language

understanding. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-2643. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/ec795aeadae0b7d230fa35cbaf04c041-Abstract.html.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 2023. URL https://proceedings.mlr.press/v202/song23a.html.

Stability AI. Stable Diffusion 3.5 Medium. Hugging Face model card, October 2024. URL https://huggingface.co/stabilityai/stable-diffusion-3.5-medium. Official model repository.

Team Hunyuan3D, Shuhui Yang, Mingxin Yang, Yifei Feng, Xin Huang, Sheng Zhang, Zebin He, Di Luo, Haolin Liu, Yunfei Zhao, Qingxiang Lin, Zeqiang Lai, Xianghui Yang, Huiwen Shi, Zibo Zhao, Bowen Zhang, Hongyu Yan, Lifu Wang, Sicong Liu, Jihong Zhang, Meng Chen, Liang Dong, Yiwen Jia, Yulin Cai, Jiaao Yu, Yixuan Tang, Dongyuan Guo, Junlin Yu, Hao Zhang, Zheng Ye, Peng He, Runzhou Wu, Shida Wei, Chao Zhang, Yonghao Tan, Yifu Sun, Lin Niu, Shirui Huang, Bojian Zheng, Shu Liu, Shilin Chen, Xiang Yuan, Xiaofeng Yang, Kai Liu, Jianchen Zhu, Peng Chen, Tian Liu, Di Wang, Yuhong Liu, Linus, Jie Jiang, Jingwei Huang, and Chunchao Guo. Hunyuan3D 2.1: From images to high-fidelity 3D assets with production-ready PBR material, 2025. URL https://arxiv.org/abs/2506.15442.

Team Hunyuan3D, Bowen Zhang, Chunchao Guo, Dongyuan Guo, Haolin Liu, Hongyu Yan, Huiwen Shi, Jiaao Yu, Jiachen Xu, Jingwei Huang, Kunhong Li, Lifu Wang, Linus, Penghao Wang, Qingxiang Lin, Ruining Tang, Xianghui Yang, Yang Li, Yirui Guan, Yunfei Zhao, Yunhan Yang, Zeqiang Lai, Zhihao Liang, and Zibo Zhao. HY3D-Bench: Generation of 3D assets, 2026. URL https://arxiv.org/abs/2602.03907.

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, April 2004. doi: 10.1109/TIP.2003.819861. URL https://doi.org/10.1109/TIP.2003.819861.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-Image technical report, 2025. URL https://arxiv.org/abs/2508.02324. Official repository: https://github.com/QwenLM/Qwen-Image; official model record: https://huggingface.co/Qwen/Qwen-Image.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023. doi: 10.48550/arXiv.2306.09341. URL https://arxiv.org/abs/2306.09341. Official repository: https://github.com/tgxs002/HPSv2; official v2.1 checkpoint record: https://huggingface.co/xswu/HPSv2/tree/main.

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*, volume 36, pp. 15903–15935, 2023. doi: 10.52202/075280-0700. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/33646ef0ed554145eab65f6250fab0c9-Abstract-Conference.html. Official repository: https://github.com/zai-org/ImageReward.

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/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_The_Unreasonable_Effectiveness_CVPR_2018_paper.html. Official implementation: https://github.com/richzhang/PerceptualSimilarity.

A Derivation and Causal Construction

A.1 Integrated flow update

The continuous flow in Eq. (1) satisfies the exact integral identity

$$x(t_{n+1}) = x(t_n) + \int_{t_n}^{t_{n+1}} v_\theta(x(t), t, y)\,\mathrm{d}t. \tag{17}$$

Dividing the integral by Δt_n gives the interval-average velocity in Eq. (5). This identity motivates the prediction target; Full-N is the numerical reference produced by the finite N-transition sampler rather than the exact continuous solution of Eq. (17). The experiments instantiate it as FULL-50.

A.2 Cubic Hermite average

For $\tau = (t - t_n)/\Delta t_n$, the cubic Hermite basis used in Eq. (6) is

$$\begin{aligned} h_{00}(\tau) &= 2\tau^3 - 3\tau^2 + 1, \quad & h_{10}(\tau) &= \tau^3 - 2\tau^2 + \tau, \\ h_{01}(\tau) &= -2\tau^3 + 3\tau^2, & h_{11}(\tau) &= \tau^3 - \tau^2. \end{aligned} \tag{18}$$

The value basis functions select the left and right endpoints, while the derivative basis functions select their normalized slopes:

$$\begin{aligned} (h_{00}, h_{10}, h_{01}, h_{11})(0) &= (1,0,0,0), \quad & (h_{00}, h_{10}, h_{01}, h_{11})(1) &= (0,0,1,0), \\ (h'_{00}, h'_{10}, h'_{01}, h'_{11})(0) &= (0,1,0,0), & (h'_{00}, h'_{10}, h'_{01}, h'_{11})(1) &= (0,0,0,1). \end{aligned} \tag{19}$$

Consequently, Eq. (6) gives $\widehat{V}_n(0) = \widehat{V}_n^L$ and $\widehat{V}_n(1) = \widehat{V}_{n+1|n}$. Since $\mathrm{d}/\mathrm{d}t = (1/\Delta t_n)\mathrm{d}/\mathrm{d}\tau$, its endpoint derivatives are $a_{L,n}$ and $a_{R,n}$. The four basis functions integrate over $[0,1]$ to $1/2$, $1/12$, $1/2$, and $-1/12$, so the average of the local polynomial is

$$\frac{1}{2}(\widehat{V}_n^L + \widehat{V}_{n+1|n}) + \frac{\Delta t_n}{12}(a_{L,n} - a_{R,n}).$$

Inserting the two pseudo-secant slope proxies gives

$$\begin{aligned} \frac{1}{2}(\widehat{V}_n^L + \widehat{V}_{n+1|n}) + \frac{1}{12}\left[(\widetilde{U}_n - \widehat{V}_n^L) - (\widehat{V}_{n+1|n} - \widetilde{U}_n)\right] \\ = \frac{1}{6}\widetilde{U}_n + \frac{5}{12}(\widehat{V}_n^L + \widehat{V}_{n+1|n}). \end{aligned}$$

The endpoint constraints, basis integrals, and final coefficients are exact algebraic properties of the constructed polynomial. The scientific approximation is the use of this local interpolant and its secant slopes to represent the velocity over a cached solver interval.

A.3 Information available at a cached step

Let $p < m < n$ be the two most recent full evaluations. BoundaryDelta stores the latent X_m and velocities V_p, V_m from those completed steps. At the beginning of cached step n, the current latent x_n is also known, so the method can compute

$$h_n = t_n - t_m, \qquad U_{m:n} = \frac{x_n - X_m}{h_n}, \qquad q_n = \frac{U_{m:n} - V_m}{h_n}.$$

These quantities determine the three values used in the cubic construction:

$$\widehat{V}_n^L = V_m, \qquad \widetilde{U}_n = V_m + \Delta t_n q_n, \qquad \widehat{V}_{n+1|n} = 2V_m - V_p.$$

Substitution recovers the grouped real-arithmetic update

$$w_n^{\mathrm{BD}} = \frac{1}{6}(V_m + \Delta t_n q_n) + \frac{5}{12}(3V_m - V_p), \qquad x_{n+1} = x_n + \Delta t_n w_n^{\mathrm{BD}}.$$

The estimate at the right boundary is extrapolated from completed anchors, and the update therefore remains causal. Startup steps 0 and 1 are always full evaluations, ensuring that p and m exist before the first cached step. After any later full evaluation, the two-anchor history advances and subsequent cached steps use the newly completed anchor pair.

A.4 Equivalent two-anchor form

The executable rule also admits a scalar two-anchor representation. Define the anchor difference $D := V_m - V_p$ and write the recurrent state within one cache run as

$$x_n - X_m = h_n V_m + r_n D, \qquad h_n := t_n - t_m, \qquad r_{m+1} = 0. \tag{20}$$

The initial condition follows because the first transition after the full evaluation at m uses V_m. Equation (20) then gives

$$U_{m:n} = V_m + \frac{r_n}{h_n} D, \qquad q_n = \frac{r_n}{h_n^2} D. \tag{21}$$

Substitution into the existing BoundaryDelta average yields

$$\begin{aligned} w_n^{\mathrm{BD}} &= \frac{1}{6}\left(V_m + \Delta t_n q_n\right) + \frac{5}{12}\left(3V_m - V_p\right) \\ &= V_m + \left(\frac{5}{12} + \frac{\Delta t_n r_n}{6h_n^2}\right)\left(V_m - V_p\right) \\ &= \alpha_n V_m + (1-\alpha_n) V_p, \qquad \alpha_n = \frac{17}{12} + \frac{\Delta t_n r_n}{6h_n^2}. \end{aligned} \tag{22}$$

Advancing x_n with this control and collecting the coefficient of D gives the scalar recurrence

$$r_{n+1} = \left(1 + \frac{\Delta t_n^2}{6h_n^2}\right) r_n + \frac{5}{12}\Delta t_n, \qquad r_{m+1} = 0. \tag{23}$$

Hence the first cached step after every full evaluation has $\alpha_{m+1} = 17/12$, and all later coefficients are fixed analytically by the signed numerical-time grid and the depth of the current cache run. If $V_p = V_m$, both forms reduce to $w_n^{\mathrm{BD}} = V_m$. Equations (22) and (23) are exact-arithmetic identities for the deterministic Euler recurrence; different floating-point evaluation orders may differ by roundoff, and no empirical equivalence is asserted. The Hermite construction remains the primary definition of BoundaryDelta.

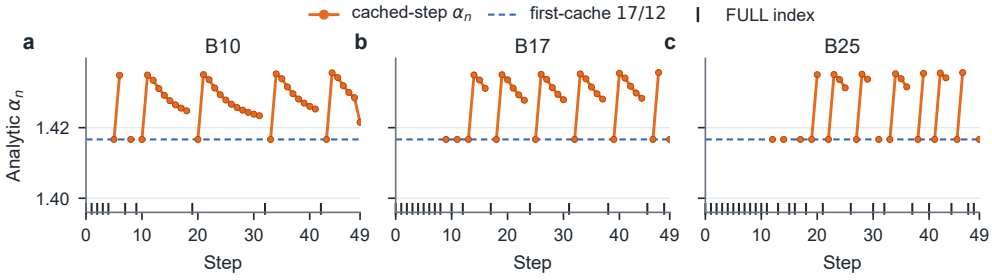


Figure 6: Exact-arithmetic equivalent coefficient schedules for the deployed Qwen-Image B10, B17, and B25 masks on the authenticated nonuniform numerical-time grid. Orange points show α_n only at cached steps; gray ticks and vertical rules mark full evaluations and cache-run boundaries. The dashed line marks the first cached-step value $17/12$. This is a deterministic analytic visualization, not a quality or runtime measurement.

A.5 Comparison with the MeanCache rolling recurrence

The released MeanCache update is an affine combination of controls in its selected rolling history. Because a cached prediction is appended as $c_k = w_k^{\mathrm{MC}}$, subsequent cached steps may recursively use earlier predicted controls. BoundaryDelta instead retains the exact form $w_n^{\mathrm{BD}} = \alpha_n V_m + (1-\alpha_n) V_p$, where both vector-valued anchors come from completed transformer evaluations. Earlier cached updates affect its recurrent state and analytic coefficient α_n, but they do not replace either model-evaluated anchor. This distinction characterizes the two recurrences; it is not, by itself, a proof of performance superiority.

B Anchor-Schedule Search

B.1 Candidate distribution

The feasible set and development objective are given by Eqs. (12) and (13). Since steps 0 and 1 are mandatory, the mutable domain and largest legal exchange are

$$\mathcal{U}_N = \{2, \dots, N-1\}, \qquad K_{\max} = \min(B-2, N-B). \tag{24}$$

At $B = 2$ or $B = N$, the feasible set contains a single schedule, so no mutation or search is needed. The nontrivial search below therefore assumes $2 < B < N$, for which $K_{\max} \geq 1$. For the $N = 50$ experiments, $K_{\max}$ is 8, 15, and 23 at B10, B17, and B25.

At radius ℓ, let $p_\ell = \min(\ell / K_{\max}, 0.99)$. The number of exchanged indices follows

$$\Pr(K = k \mid \ell) = \frac{\binom{K_{\max}}{k} p_\ell^k (1-p_\ell)^{K_{\max}-k}}{1-(1-p_\ell)^{K_{\max}}}, \quad k = 1, \dots, K_{\max}, \qquad \ell_0 = 1. \tag{25}$$

Thus K is positive. Given K, the search draws selected indices R and unselected indices A uniformly and applies

$$R \subseteq S \cap \mathcal{U}_N, \quad A \subseteq \mathcal{U}_N \setminus S, \quad |R| = |A| = K, \qquad S' = (S \setminus R) \cup A. \tag{26}$$

The equal exchange keeps exactly B evaluations and never removes the two startup anchors.

For finite losses x and y, define

$$\mathrm{tol}(x, y) = 10^{-12} + 10^{-9} \max(|x|, |y|), \qquad \mathrm{strict}(x, y) \iff x < y - \mathrm{tol}(x, y). \tag{27}$$

A finite candidate precedes every non-finite candidate. Finite values within tolerance are ordered by mask hash, but such a replacement is not a strict success. With $F = 1.5$, the radius update is

$$\ell_{t+1} = \begin{cases} \max(1, \ell_t / F), & \text{strict success}, \\ \min(K_{\max}, \ell_t F^{1/4}), & \text{otherwise}. \end{cases} \tag{28}$$

The first branch narrows the next search around an improved parent; the second broadens the exchange after a batch that does not strictly improve it.

B.2 Full search procedure

The eight distinct starts are drawn from a released cache schedule, structured placements, and random feasible sets, and every objective query evaluates the complete development set. The procedure below makes the charged-query, archive, and restart state explicit. $\mathrm{Best}_{\mathrm{tol,hash}}$ selects the raw minimum finite loss and then the smallest mask hash in its tolerance tier; if a set contains no finite loss, it selects the smallest hash.

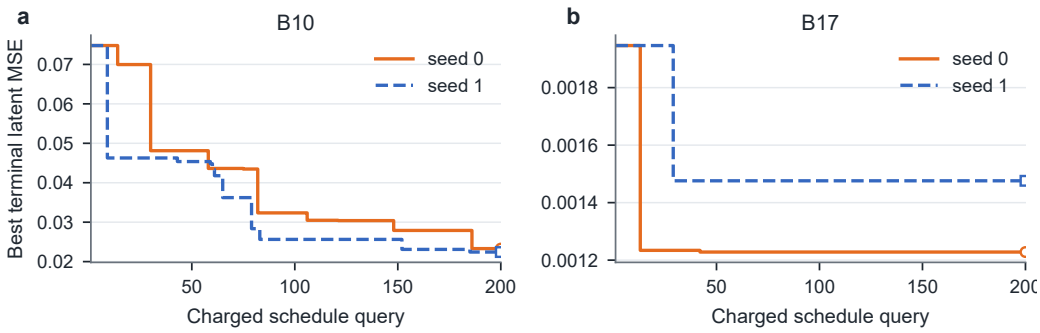


Figure 7: Qwen-Image anchor-search convergence. The panels plot best-so-far terminal latent MSE for both optimizer seeds at B10 and B17 against charged schedule queries. These optimization trajectories are distinct from the fixed deployment schedules in Figure 5.

Algorithm 1: Self-adjusting $(1 + \Lambda)$ fixed-call anchor search

Input: N, B with $2 < B < N$; $\mathcal{F}_{B,N}, \mathcal{L}$; eight distinct starts $\mathcal{I}$; $Q = 200$; $\Lambda = 32$; $F = 1.5$; stagnation limit 5.
Evaluate $\mathcal{I}$; set archive $\mathcal{A} \leftarrow \mathcal{I}$, $q \leftarrow 8$, incumbent $S \leftarrow \mathrm{Best}_{\mathrm{tol,hash}}(\mathcal{I})$, radius $\ell \leftarrow 1$, and stagnation count $s \leftarrow 0$.
while $q < Q$ **do**
  if $s = 5$ **then** evaluate unseen feasible restart schedules, charging each to q, until a finite restart S_r is found or $q = Q$; if none is found, **break**; otherwise set $S \leftarrow S_r$, $\ell \leftarrow 1$, $s \leftarrow 0$, and **continue**.
  $M \leftarrow \min(\Lambda, Q - q)$; $\mathcal{C} \leftarrow \varnothing$; record the pre-batch archive best $S_\star$.
  Draw unseen feasible proposals by Eqs. (25)–(26) until $\mathcal{C}$ contains M unique candidates; evaluate $\mathcal{C}$, set $\mathcal{A} \leftarrow \mathcal{A} \cup \mathcal{C}$, and charge $q \leftarrow q + M$.
  $S_c \leftarrow \mathrm{Best}_{\mathrm{tol,hash}}(\mathcal{C})$; store $S_{\mathrm{old}} \leftarrow S$ and replace S when S_c precedes S_{old} under finite-first tolerance/hash order.
  Set u to true exactly when $\mathcal{L}(S_c)$ is finite and either $\mathcal{L}(S_{\mathrm{old}})$ is non-finite or strict$(\mathcal{L}(S_c), \mathcal{L}(S_{\mathrm{old}}))$; use u to choose the corresponding branch of Eq. (28).
  Let $S'_\star \leftarrow \mathrm{Best}_{\mathrm{tol,hash}}(\mathcal{A})$; set $s \leftarrow 0$ when $S'_\star$ strictly improves $S_\star$ under the same finite-first rule, and otherwise set $s \leftarrow s + 1$.
end while
Output: finite $\mathrm{Best}_{\mathrm{tol,hash}}(\mathcal{A})$, or no schedule if no archived loss is finite.

The reported lanes stop when the $Q = 200$ unique-query budget is exhausted. For the reported searches, the two B10 seeds reduce best-so-far development loss from 0.07478 to 0.02330 and 0.02243; the B17 seeds reduce it from 0.001946 to 0.001228 and 0.001476.

B.3 Development data and mask selection

The image campaigns reserve a disjoint 50-prompt, five-category T2V-CompBench development pool: 10 search prompts with one seed per prompt, 15 selection prompts with two seeds, and 25 confirmation prompts with five seeds. Qwen-Image searches B6–B10/B17/B25 at 1664×928; Stable Diffusion 3.5 Medium searches B10/B17 at 1024×1024. Each budget is searched independently on the corresponding 50-step model configuration. These are development data: the 200 DrawBench prompts used for final image evaluation are excluded.

Every optimizer query is scored by mean terminal-latent MSE to the paired FULL-50 endpoint, with prompt and seed averaging as in Eq. (13); decoded final-evaluation scores are not optimizer feedback. The registered Qwen development rule ranks retained candidates by lower development terminal MSE and uses development LPIPS only as an exact-primary-tie fallback; the SD3.5 rule is terminal-MSE-only. Within a search lane, finite losses precede non-finite losses, and values inside the tolerance in Eq. (27) are ordered by mask hash. Hunyuan3D instead searches B17 on five HY3D-Bench validation objects at generation seed 42, again using mean terminal-latent MSE to paired FULL-50; its 50 held-out evaluation objects, decoded meshes, and ground-truth geometry do not enter selection. Thus campaign-specific development panels differ, but final DrawBench prompts and held-out 3D objects remain separated from mask search.

B.4 Query accounting and selected schedules

One query evaluates one previously unseen schedule on every development prompt–seed pair. The accounting identity is

$$q = N_{\text{start}} + N_{\text{offspring}} + N_{\text{restart}}, \qquad N_{\text{start}} = 8, \qquad q \le Q = 200. \tag{29}$$

Starts, unique children, and unique restarts consume queries; duplicate proposals do not.

Table 4 lists the Qwen-Image schedules deployed in the final experiments. The indices identify full transformer evaluations on the common 50-step grid; all other steps use BoundaryDelta.

Table 4: Frozen Qwen-Image full-evaluation indices for BoundaryDelta.

Budget	Full-evaluation diffusion-step indices
B6	0, 1, 2, 5, 6, 27
B7	0, 1, 2, 5, 9, 15, 17
B8	0, 1, 2, 3, 6, 7, 23, 36
B9	0, 1, 2, 4, 6, 8, 13, 14, 39
B10	0, 1, 2, 3, 4, 7, 9, 19, 32, 42
B17	0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 17, 24, 31, 38, 45, 48
B25	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 15, 16, 18, 21, 26, 30, 32, 37, 40, 44, 47, 48

For the uniform-index controls, the Qwen-Image full-evaluation indices are B10 $\{0, 1, 6, 11, 16, 21, 26, 32, 38, 44\}$, B17 $\{0, 1, 4, 7, 10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, 43, 46\}$, and B25 $\{0, 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47\}$. The Hunyuan3D uniform B17 control uses $\{0, 1, 2, 5, 9, 12, 15, 18, 22, 25, 28, 32, 35, 38, 41, 45, 48\}$.

C Experimental Details

C.1 Generation settings

Each campaign materializes one initial latent for every DrawBench prompt and reuses it across paired methods together with the same model, sampler, conditioning, and decoder. FULL-50 is generated once for that model–prompt case and supplies the finite 50-transition numerical reference. The complete 200-prompt benchmark is excluded from schedule search.

Table 5: Frozen model and generation configurations. The citations support model identity; inference settings are project-artifact fields.

Field	Qwen-Image	Stable Diffusion 3.5 Medium
Model	Qwen-Image (Wu et al., 2025)	`stabilityai/stable-diffusion-3.5-medium` (Esser et al., 2024; Stability AI, 2024)
Sampler	FlowMatch Euler; 50 reference steps	FlowMatch Euler; 50 reference steps
Image size	1664×928	1024×1024
Numerics	bfloat16; true CFG 4.0; guidance 1.0	bfloat16; CFG batch order [negative, positive]; guidance 4.5; maximum sequence length 256
Prompt conditioning	suffix "Ultra HD, 4K, cinematic composition."; empty negative prompt	no positive suffix; empty negative prompt
Benchmark	Complete 200-prompt DrawBench suite (Saharia et al., 2022); one generation seed, 42	
Exact budgets	BoundaryDelta B6–B10/B17/B25; matched panels B10/B17/B25	B10 and B17

BoundaryDelta stores its anchor velocities and performs the controller arithmetic in FP32, then casts the emitted control to the runtime tensor dtype: bf16 for Qwen-Image and SD3.5, fp16 for Hunyuan3D.

C.2 Image-conditioned 3D protocol

The Hunyuan3D study uses the 2048-token Small configuration of Hunyuan3D-Shape-v2.1 (Team Hunyuan3D et al., 2025) for single-image-to-3D shape generation; texture generation is excluded.

Five HY3D-Bench (Team Hunyuan3D et al., 2026) validation objects are used only for B17 schedule search, and 50 disjoint test objects form the held-out evaluation. Every comparison uses the same conditioning image and seed 42. FULL-50 follows the official 50-call grid, while BoundaryDelta evaluates indices $\{0, 1, 3, 5, 6, 8, 10, 13, 14, 16, 17, 22, 25, 28, 32, 39, 43\}$. The compared rows are FULL-50, BoundaryDelta, Copy, BoundaryDelta with a uniform B17 schedule, and native 17-step sampling.

For each generated and ground-truth mesh, the analysis draws 100,000 deterministic area-weighted surface samples. With point sets C and R and $d(x, R) = \min_{y \in R} \|x - y\|_2$, the reported Chamfer measures are

$$\mathrm{CD}_{L1} = \tfrac{1}{2}(\mathbb{E}_{c \in C} d(c, R) + \mathbb{E}_{r \in R} d(r, C)), \qquad \mathrm{CD}_{L2} = \tfrac{1}{2}\left(\mathbb{E}_{c \in C} d(c, R)^2 + \mathbb{E}_{r \in R} d(r, C)^2\right).$$

F-score is the harmonic mean of bidirectional precision and recall at thresholds 0.01 and 0.02; normal consistency averages the absolute dot product of paired nearest-neighbor face normals in both directions. Metrics prefixed by GT use the official ground-truth mesh; the schedule effects reported in Section 7 use the paired FULL-50 generated mesh. The held-out object is the statistical unit. Uncertainty uses 20,000 deterministic object-resampling percentile-bootstrap draws and the empirical 2.5th and 97.5th percentiles.

C.3 Methods and compute

Exact NFE counts complete transformer evaluations. For adaptive methods, measured denoising FLOPs are divided by the FLOPs of one complete transformer evaluation to obtain realized FEC, which may be non-integer and may differ from the named target regime.

One exact NFE denotes one solver position at which the complete guided model prediction is evaluated. For Qwen-Image true CFG, this includes the conditional and unconditional transformer branches, which are combined before the resulting guided velocity is stored or used by the cache controller. For SD3.5, the negative and positive CFG branches are evaluated in one batch and likewise combined before caching. Timing and full-evaluation FLOP accounting include the complete CFG computation associated with that solver position.

The SD3.5 reduced-step row runs the FlowMatch scheduler on a 10- or 17-step grid. We implement the published MeanCache recurrence and construct its schedule specifically for SD3.5.

C.4 Metrics and aggregation

Every metric is computed per prompt and then averaged with equal prompt weight. Fidelity metrics compare an accelerated output with its paired FULL-50 output. Semantic and preference metrics score the generated image against the prompt and are not combined with fidelity into a scalar.

D Consolidated Results and Qualitative Evidence

Table 1 contains the primary Qwen B10/B17 comparison, Table 8 contains the corresponding B25 panel, and Table 2 contains the SD3.5 exact-budget panel. The material below adds the dense Qwen frontier, compute details, external SD3.5 results, mechanism studies, and qualitative evidence. Exact NFE and measured FEC remain distinct throughout.

Table 6: Baseline configurations and compute accounting. Qwen compute entries are ordered B10/B17/B25. Exact NFE counts complete transformer evaluations; adaptive rows report realized mean FEC from measured denoising FLOPs. Configuration values absent from the routed manuscript evidence are identified rather than inferred.

Method	Evaluated configuration	Provenance/status	Target and realized compute
BoundaryDelta	Frozen deployment masks; B6–B10, B17, and B25 are available, with primary comparisons at B10/B17/B25.	Project method; exact whole-step cache	Exact NFE $10/17/25$
MeanCache (Gao et al., 2026a) / Copy	Released MeanCache masks; Copy reuses the latest full prediction on the same masks.	Released method / project exact control	Exact NFE $10/17/25$
Reduced-step Qwen	Native solver rerun on B numerical steps, with every step evaluated; not a 50-step cache mask.	Project exact control	Exact NFE $10/17/25$
LeMiCa (Gao et al., 2025)	Authenticated released B10/B17/B25 schedules.	Released, paper-faithful Qwen implementation	Exact NFE $10/17/25$; measured FEC $10.002/17.002/25.001$
MagCache (Ma et al., 2025)	Official Qwen adapter; retention ratio 0.2 and strict accumulated-error test. Budget-specific thresholds and maximum reuse intervals were frozen by development-only cost matching but are not repeated in the routed manuscript evidence.	Paper-faithful reproduction	Realized FEC $15.002/17.002/25.001$
TeaCache (Liu et al., 2025a)	Qwen adaptation with a degree-4 fitted rescaling polynomial and paired CFG decision. Budget-specific thresholds, warm-up anchors, and reuse limits are not repeated in the routed manuscript evidence.	Project Qwen adaptation	Realized FEC $9.002/18.002/26.001$
DBCache $r = 0.6$ (Cache-DiT)	$F_n = 2$, $B_n = 4$, dynamic block caching, maximum warm-up 8, separate CFG handling.	Paper-faithful reproduction	Realized FEC $14.924/17.206/25.877$
DBCache $r = 1.5$ (Cache-DiT)	$F_n = 2$, $B_n = 4$, dynamic block caching, maximum warm-up 8, separate CFG handling.	Paper-faithful reproduction	Realized FEC $12.202/15.802/24.801$
DBCache+ TaylorSeer	DBCache $r = 1.5$ with fourth-order TaylorSeer feature forecasting.	Paper-faithful composed baseline (Cache-DiT)(Liu et al., 2025c)	Realized FEC $12.202/15.802/24.801$
TaylorSeer (Liu et al., 2025c)	Fourth-order forecasting with separate CFG histories, six mandatory startup evaluations, a final evaluation, and frozen budget-specific refresh indices.	Project Qwen adaptation	Realized FEC $10.002/17.002/25.001$
SD3.5 exact panel	BoundaryDelta, Copy, MeanCache (Gao et al., 2026a), and reduced-step SD3.5 at B10/B17.	Separate frozen SD3.5 protocol	Exact NFE $10/17$
SD3.5 measured panel	LeMiCa transfer, TaylorSeer O4, TeaCache, MagCache, two DBCache settings, and DBCache+TaylorSeer.	Separate target-regime panel	Frozen target and realized mean FEC are reported in Appendix D

Table 7: Metric definitions, directions, and preprocessing shared by the reported studies.

Family	Metric	Direction	Backend and preprocessing
Terminal fidelity	MSE	↓	Mean squared difference between matched finite terminal latent arrays, accumulated in float64
Terminal fidelity	L1	↓	Mean absolute difference between the same matched terminal arrays, accumulated in float64
Image fidelity	LPIPS-Alex (Zhang et al., 2018)	↓	Official AlexNet-backed model; RGB inputs converted to NCHW float32 in $[-1, 1]$
Image fidelity	SSIM (Wang et al., 2004)	↑	RGB images at generated resolution with channel-aware evaluation and data range 255
Image fidelity	PSNR	↑	$10 \log_{10}(255^2 / \mathrm{MSE}_{\mathrm{RGB}})$ using channel-inclusive float64 MSE
Semantic	CLIP-G (Radford et al., 2021; Cherti et al., 2023)	↑	OpenCLIP ViT-g/14 with LAION-2B weights; normalized image/text embeddings and cosine similarity
Preference	ImageReward (Xu et al., 2023)	↑	Official ImageReward v1.0 scoring interface
Preference	HPSv2.1 checkpoint	↑	Official HPSv2.1 project checkpoint associated with HPSv2 (Wu et al., 2023); ViT-H/14 scoring model and validation transform

D.1 Complete Qwen-Image B25 panel

Table 8: Qwen-Image B25 quality and efficiency. Eval. gives the recorded evaluation amount, latency is the median in seconds, and reported speedups are paired. Bold and underline mark the best and second-best quality point estimates among the five equal-budget rows. The highlighted orange row is BoundaryDelta.

B	Method	Eval.	Lat.↓	Spd.↑	L1↓	LPIPS↓	SSIM↑	PSNR↑	CLIP-G↑	IR↑	HPS↑
	BoundaryDelta (ours)	25	70.826	1.977×	**0.0225**	**0.0250**	**0.9754**	**34.88**	0.33430	1.2187	**0.3004**
	MeanCache	25	71.207	1.961×	0.0411	0.0540	0.9554	30.72	0.33230	**1.2203**	0.3000
	Copy	25	70.528	1.967×	0.0321	0.0441	0.9619	31.38	**0.33436**	1.2159	0.2993
	Reduced-step Qwen	25	70.974	1.956×	0.1487	0.2700	0.7701	18.06	0.33377	1.2003	0.2991
	LeMiCa	25	71.187	1.953×	0.0422	0.0622	0.9529	31.51	0.33435	1.1979	0.2979
B25	MagCache	25.001	70.353	1.965×	0.0261	0.0336	0.9701	32.90	0.33395	1.2148	0.2999
	DBCache r=0.6	25.877	72.994	1.895×	0.0397	0.0578	0.9451	28.87	0.33377	1.2175	0.3002
	DBCache r=1.5	24.801	69.946	1.986×	0.0417	0.0598	0.9431	28.71	0.33402	1.2188	0.3007
	DBCache+TaylorSeer r=1.5 O4	24.801	69.699	1.979×	0.0603	0.0937	0.9138	25.72	0.33325	1.2326	0.3024
	TeaCache	26.001	73.267	1.893×	0.1101	0.1956	0.8307	20.61	0.33428	1.2155	0.2993
	TaylorSeer O4	25.001	70.854	1.953×	0.0603	0.0972	0.9114	25.38	0.33304	1.2175	0.3011

D.2 Complete Qwen-Image BoundaryDelta frontier

Table 9: Complete Qwen-Image BoundaryDelta budget frontier. The upper panel gives 200-prompt means. The lower panel separates exact NFE from median normalized end-to-end latency and speedup.

B	MSE↓	L1↓	LPIPS↓	SSIM↑	PSNR↑	CLIP-G↑	IR↑	HPS↑
B6	0.0960	0.2258	0.4555	0.6535	16.34	0.32579	0.7695	0.2587
B7	0.0699	0.1808	0.3791	0.7051	17.65	0.33383	0.9579	0.2751
B8	0.0532	0.1606	0.3029	0.7452	18.82	0.33362	1.1532	0.2878
B9	0.0415	0.1363	0.2674	0.7807	20.17	0.33482	1.1509	0.2874
B10	0.0284	0.1093	0.1948	0.8294	21.65	0.33436	1.2075	0.2962
B17	0.0061	0.0419	0.0599	0.9445	29.24	0.33456	1.2130	0.2997
B25	0.0021	0.0225	0.0250	0.9754	34.88	0.33430	1.2187	0.3004

Budget	Exact NFE	Median latency (s)↓	Speedup↑
B6	6	18.333	8.231×
B7	7	21.370	6.565×
B8	8	24.125	5.804×
B9	9	26.904	5.183×
B10	10	29.543	4.723×
B17	17	48.972	2.841×
B25	25	70.826	1.977×

BoundaryDelta also operates at the more aggressive Qwen-Image B6–B9 budgets. All five fidelity metrics improve monotonically from B6 to B10. The lowest budgets already preserve broad scene content, while additional calls progressively recover texture, text, and other fine details. Semantic and preference scores stabilize earlier than FULL-reference fidelity: B9 still has 46.2% higher terminal MSE and 37.3% higher LPIPS than B10. Figure 8 shows these trends across the full budget range.

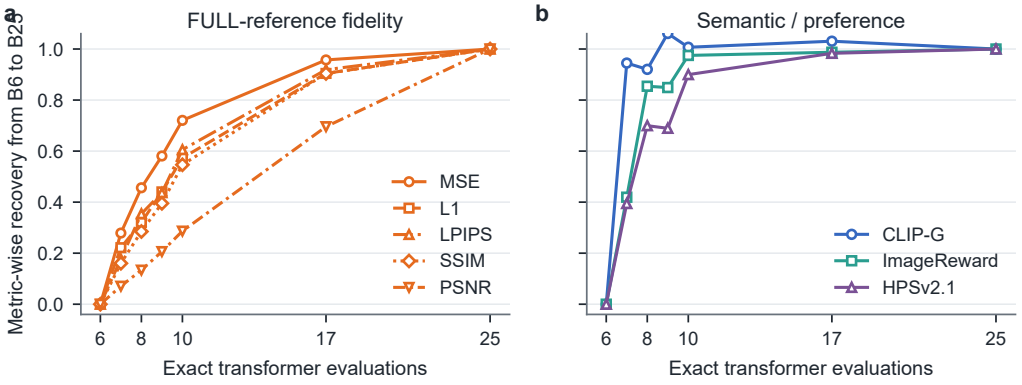


Figure 8: BoundaryDelta's Qwen-Image budget frontier. The left panel reports normalized FULL-reference fidelity, for which higher is better after orienting each metric; the right panel reports the three semantic/preference metrics separately. B6–B10 form the dense sweep, with B17 and B25 showing more moderate call budgets. Each metric is normalized only for display.

D.3 Stable Diffusion 3.5 Medium external panel

Table 10: Stable Diffusion 3.5 Medium quality at the B10 and B17 target regimes. BoundaryDelta, MeanCache transfer, and MeanCache native report exact transformer-evaluation counts; the seven external rows report realized mean full-equivalent compute. MeanCache transfer uses the released Qwen schedules, while MeanCache native uses SD3.5-specific searched schedules from our paper-faithful reimplementation. Quality values are 200-prompt means; Lat. is median wall time and Spd. is the descriptive FULL-50 median divided by the row median across separate timing cohorts.

Target	Method	Eval.	Lat.↓	Spd.↑	MSE↓	L1↓	LPIPS↓	SSIM↑	PSNR↑	CLIP-G↑	IR↑	HPS↑
B10	**BoundaryDelta (ours)**	10	4.606	4.414×	0.0676	0.1402	0.1976	0.8197	21.66	0.33162	0.8064	0.2659
	MeanCache transfer	10	4.467	4.551×	0.1691	0.2519	0.2928	0.7615	17.52	0.32987	0.7543	0.2630
	MeanCache native	10	4.632	4.389×	0.1766	0.3198	0.4742	0.6845	19.59	0.31758	0.4778	0.2388
	LeMiCa transfer	10.021	4.513	4.505×	0.1616	0.2312	0.3214	0.7408	17.52	0.33133	0.7384	0.2615
	TaylorSeer O4	10.021	4.571	4.448×	1.0513	0.7783	0.7479	0.4022	9.67	0.21823	-1.1523	0.1771
	TeaCache	11.021	4.972	4.089×	0.3810	0.4308	0.4086	0.6853	14.37	0.32868	0.7096	0.2628
	MagCache	14.019	6.093	3.337×	0.0428	0.1161	0.1867	0.8407	23.50	0.33170	0.7971	0.2664
	DBCache (r=0.6)	18.795	7.796	2.608×	0.1180	0.2047	0.3020	0.7748	18.79	0.30660	0.1809	0.2218
	DBCache (r=1.5)	17.635	7.235	2.810×	0.1387	0.2321	0.3436	0.7397	18.23	0.29717	-0.0856	0.2071
	DBCache+TaylorSeer (r=1.5, O4)	17.636	7.346	2.768×	8.4797	2.2478	0.9137	0.1338	5.94	0.15884	-1.9328	0.1444
B17	**BoundaryDelta (ours)**	17	7.251	2.804×	0.0156	0.0619	0.0701	0.9376	28.57	0.33033	0.8972	0.2766
	MeanCache transfer	17	7.244	2.806×	0.0331	0.1133	0.1133	0.9013	25.38	0.33214	0.8677	0.2720
	MeanCache native	17	7.462	2.725×	0.0600	0.1770	0.2431	0.8369	23.70	0.32798	0.7786	0.2634
	LeMiCa transfer	17.017	7.274	2.795×	0.0263	0.0922	0.1418	0.8719	25.96	0.33022	0.8370	0.2712
	TaylorSeer O4	17.018	7.335	2.772×	0.2300	0.2891	0.2977	0.7422	16.24	0.33053	0.7258	0.2648
	TeaCache	18.017	7.744	2.625×	0.3088	0.3627	0.3270	0.7425	15.57	0.33548	0.8878	0.2747
	MagCache	18.017	7.663	2.653×	0.0298	0.0861	0.1194	0.9027	25.19	0.33185	0.8450	0.2723
	DBCache (r=0.6)	18.795	7.789	2.610×	0.1180	0.2047	0.3020	0.7748	18.79	0.30660	0.1809	0.2218
	DBCache (r=1.5)	17.635	7.229	2.813×	0.1387	0.2321	0.3436	0.7397	18.23	0.29717	-0.0856	0.2071
	DBCache+TaylorSeer (r=1.5, O4)	17.636	7.344	2.768×	8.4797	2.2478	0.9137	0.1338	5.94	0.15884	-1.9328	0.1444

MeanCache transfer applies the original recurrence with released Qwen schedules, whereas MeanCache native uses the same recurrence with SD3.5-specific searched schedules from our paper-faithful reimplementation. Transfer is stronger on seven of eight B10 and all eight B17 reported means, yet BoundaryDelta improves all five fidelity means over transfer at both budgets.



Figure 9: Stable Diffusion 3.5 Medium exact-budget comparisons at B10 and B17. Columns show FULL-50, BoundaryDelta, Copy, MeanCache transfer, MeanCache native, and reduced-step SD3.5. Rows are the registered worst, median, and best prompts by BoundaryDelta's standardized fidelity effect relative to MeanCache transfer.

D.4 Prediction rule and anchor placement

The crossed experiment isolates prediction-rule and schedule effects by evaluating the MeanCache and BoundaryDelta rules with both the released MeanCache schedule S_{MC} and the selected BoundaryDelta schedule S^*_{BD}. Table 11 gives all four crossed cell means and adds the BoundaryDelta uniform-index control. Across budgets, S^*_{BD} improves both rules, whereas the relative controller effect changes with budget; the point estimates therefore support complementary but budget-dependent contributions.

Table 11: Crossed prediction-rule and schedule experiment with a BoundaryDelta uniform-index schedule control. Each row gives the four crossed cell means and the uniform-index mean. Bold and underlining mark the best and second-best point estimates within each budget–metric row.

B	Metric	MeanCache +S_{MC}	MeanCache +S^{*}_{BD}	BoundaryDelta +S_{MC}	BoundaryDelta +S^{*}_{BD}	BoundaryDelta +S_{uniform}
B17	MSE	0.0084	<u>0.0062</u>	0.0085	**0.0061**	0.0791
B17	L1	0.0542	<u>0.0458</u>	0.0499	**0.0419**	0.1705
B17	LPIPS	0.0809	<u>0.0640</u>	0.0769	**0.0599**	0.2939
B17	SSIM	0.9287	<u>0.9427</u>	0.9301	**0.9445**	0.7466
B17	PSNR	27.36	<u>28.96</u>	27.40	**29.24**	17.40
B10	MSE	0.0533	**0.0234**	0.0742	<u>0.0284</u>	0.1600
B10	L1	0.1471	**0.0994**	0.1812	<u>0.1093</u>	0.2738
B10	LPIPS	0.2691	**0.1828**	0.3261	<u>0.1948</u>	0.4488
B10	SSIM	0.7717	**0.8469**	0.7281	<u>0.8294</u>	0.6343
B10	PSNR	18.61	**22.71**	17.11	<u>21.65</u>	13.51
B25	MSE	0.0041	<u>0.0021</u>	0.0035	**0.0021**	0.0444
B25	L1	0.0411	<u>0.0239</u>	0.0314	**0.0225**	0.1148
B25	LPIPS	0.0540	<u>0.0259</u>	0.0416	**0.0250**	0.1984
B25	SSIM	0.9554	<u>0.9749</u>	0.9623	**0.9754**	0.8253
B25	PSNR	30.72	<u>34.32</u>	31.57	**34.88**	20.76

D.5 BoundaryDelta slot ablation

The B17 slot study holds the fixed evaluation schedule, prompt, seed, and initial latent constant while replacing one term in the integrated update. Its master expression and BoundaryDelta mapping are

$$w_n = \frac{b_n + e_n}{2} + \frac{2u_n - b_n - e_n}{12}, \qquad b_n = V_m, \quad u_n = V_m + \Delta t_n q_n, \quad e_n = 2V_m - V_p.$$

Here b_n is the persistent left boundary, u_n is the provisional interval average, and e_n is the predicted right endpoint. Each ablation replaces one slot while holding the other two at these BoundaryDelta values. In the original slot-study bookkeeping, $q_L = q_n$; we use the main-text symbol q_n below. The exact replacements are

$$\begin{aligned}
b &: V_m + \frac{h_n(V_m - V_p)}{t_m - t_p}, \quad \frac{1}{2}\left\{V_m + V_m + \frac{(h_n + \Delta t_n)(V_m - V_p)}{t_m - t_p}\right\}, \quad w_{n-1}, \\
u &: V_m, \quad U_{m:n}, \quad V_m - \Delta t_n q_n, \\
e &: V_m, \quad V_m + \frac{(h_n + \Delta t_n)(V_m - V_p)}{t_m - t_p}, \quad 2V_m - 2U_{p:m} + V_p,
\end{aligned}$$

where $h_n = t_n - t_m$ and $U_{p:m} = (X_m - X_p)/(t_m - t_p)$. The additional symmetric provider averages the left rate $q_L = q_n$ with the constant right rate $-q_n$, so $u_n = V_m + \Delta t_n(q_n - q_n)/2 = V_m$. It therefore reduces algebraically to anchor persistence and is an identity control, not independent evidence.

Table 12: B17 slot replacements relative to BoundaryDelta. Entries are oriented mean changes (%): positive favors the replacement and negative favors BoundaryDelta. †The symmetric-q_L/q_R row is numerically identical to anchor persistence and is collapsed here as an identity control.

Replacement	MSE	L1	LPIPS	SSIM	PSNR
b: current linear	-37.44	-23.94	-26.25	-1.53	-5.50
b: midpoint	-14.95	-9.73	-10.62	-0.65	-2.64
b: previous control	-18.65	-13.31	-13.00	-0.77	-2.95
u: anchor persistence†	+0.22	-0.02	-0.67	-0.01	+0.07
u: past chord	-1.45	-1.15	-1.02	-0.08	-0.27
u: constant q_R	+0.61	+0.17	-0.19	-0.00	-0.06
e: anchor persistence	+2.75	-2.62	-11.93	-0.17	-0.50
e: velocity-linear	-10.89	-6.28	-9.70	-0.49	-1.92
e: physical residual	-7.91	-9.77	-11.77	-0.45	-1.45

D.6 Image-conditioned 3D protocol and results

Appendix C defines the Hunyuan3D protocol and geometry metrics. The study uses 50 held-out objects, five complete method rows, and deterministic 100,000-point surface sampling per mesh. Its

B17 search set contains five disjoint validation objects. Neither ground-truth meshes, decoded held-out meshes, nor visualization renders enter the optimizer.

Table 13: Paired B17 fidelity to the generated FULL-50 mesh/reference on 50 held-out HY3D-Bench objects. The searched and uniform rows use the same BoundaryDelta controller and differ only in anchor placement. Positive oriented differences favor the searched schedule. These FULL-reference measures are distinct from the ground-truth (GT) HY3D metrics in Table 14.

Metric	Searched B17	Uniform B17	Rel. improvement	Oriented Δ [95% CI]
Terminal MSE↓	0.1935	0.2275	14.9%	+0.0339 [+0.0234, +0.0459]
F@.01↑	0.5239	0.4512	16.1%	+0.0727 [+0.0305, +0.1156]
F@.02↑	0.7465	0.6693	11.5%	+0.0773 [+0.0328, +0.1229]
Normal↑	0.7865	0.7595	3.6%	+0.0270 [+0.0124, +0.0434]

Table 14: Hunyuan3D-Shape-v2.1 Small on 50 held-out HY3D-Bench objects. Geometry metrics compare generated shapes with ground truth. Latency is mean raw transformer-denoising time; acceleration is the descriptive FULL-50 mean divided by the method mean and excludes mesh decoding. It is neither paired nor end-to-end.

Method	Calls	Ch.-L1↓	Ch.-L2↓	F@.01↑	F@.02↑	Normal↑	Denoise (s)↓	Raw accel.↑
FULL-50	50	0.1704	0.0665	0.0883	0.1817	0.5530	5.507	1.000×
BoundaryDelta	17	0.1682	0.0650	0.0899	0.1825	0.5546	1.854	2.970×
Copy	17	0.1690	0.0654	0.0935	0.1845	0.5525	1.835	3.002×
BoundaryDelta uniform	17	0.1717	0.0673	0.0840	0.1732	0.5520	1.846	2.982×
Native-17	17	0.1706	0.0668	0.0860	0.1770	0.5537	1.839	2.994×

BoundaryDelta obtains geometry scores close to FULL-50 while reducing mean raw transformer-denoising time from 5.507 to 1.854 seconds, corresponding to 2.970× raw denoising acceleration.

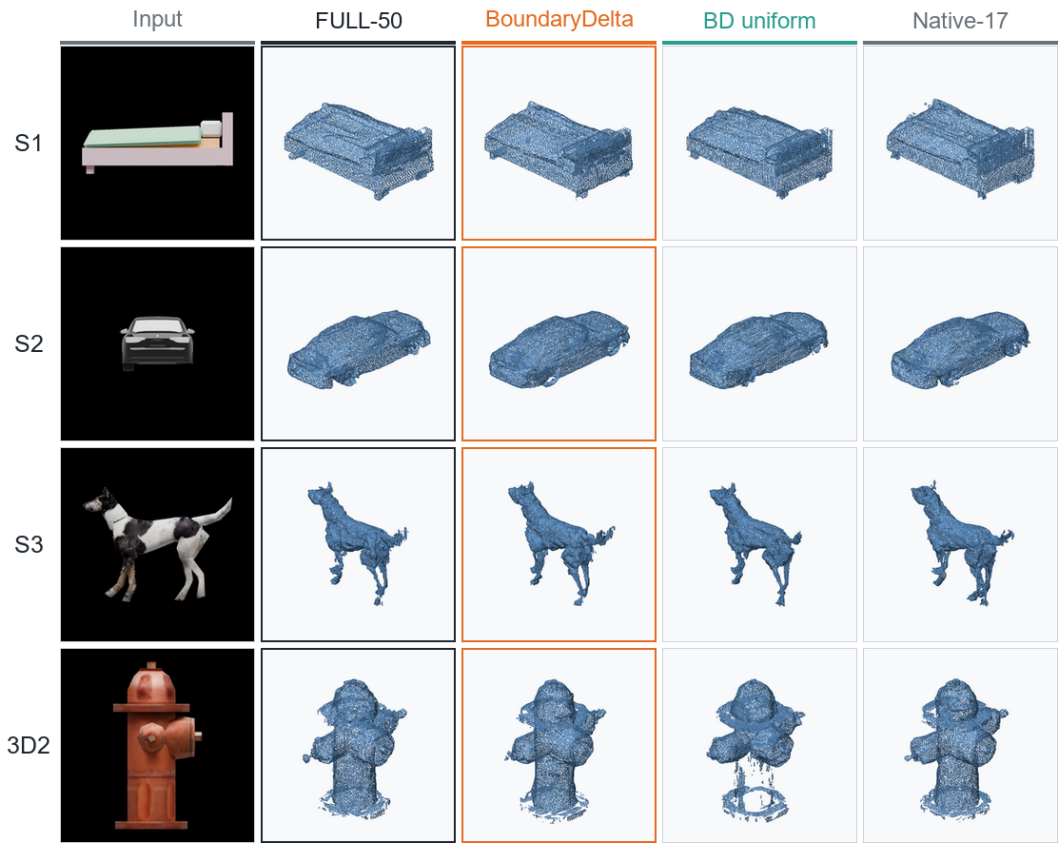


Figure 10: Additional Hunyuan3D schedule comparisons for three recognizable, category-diverse held-out objects (S1 bed, S2 civilian car, and S3 dog) and the main-paper 3D2 Fire Equipment case. Each row uses the same conditioning image and perspective camera across FULL-50, searched-schedule BoundaryDelta, uniform-schedule BoundaryDelta, and Native-17.

D.7 Cross-model schedules

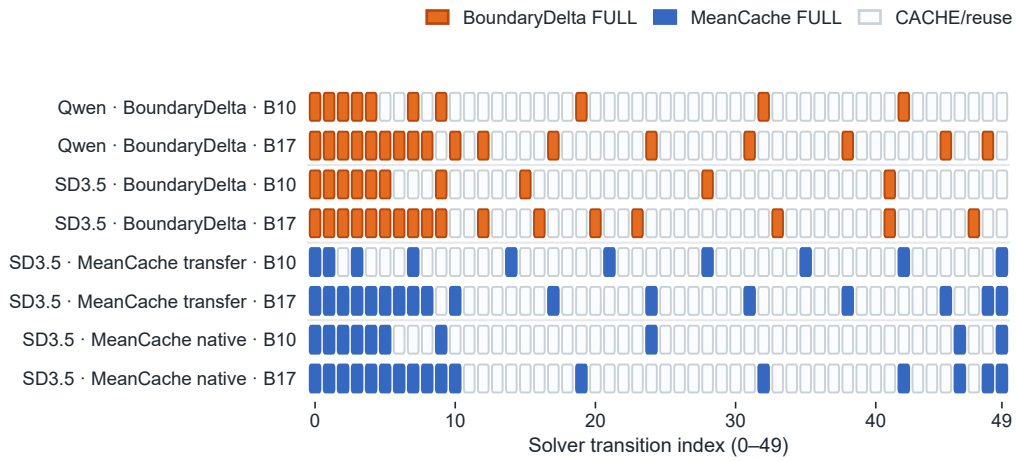


Figure 11: B10/B17 evaluation schedules across the two image models and methods. Each row spans indices 0–49; filled cells are FULL evaluations and open cells are CACHE/reuse. The Qwen rows show BoundaryDelta; the SD3.5 rows distinguish BoundaryDelta, MeanCache transfer with released Qwen schedules, and MeanCache native with SD3.5-specific searched schedules.

D.8 Qualitative examples

Figure 12 retains the paired comparison of the storefront prompt at B17, the parking-machine prompt at B17, and the same parking-machine case at B10. Prompt, generation seed, and initial latent are paired within each row.

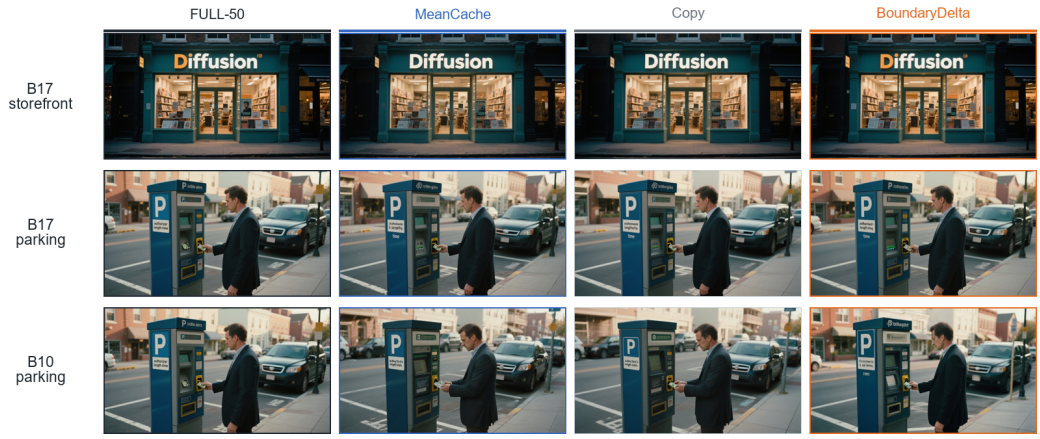


Figure 12: Paired qualitative comparisons. Columns show FULL-50, MeanCache, Copy, and BoundaryDelta. The top row uses the prompt "A storefront with 'Diffusion' written on it." The lower rows use the same parking-machine prompt at B17 and B10.

The next two aligned panels extend the main B17 comparison to eight deterministic category-diverse prompts. Exact-NFE and adaptive-FEC methods remain separated, while P1–P8 refer to the common prompt key in Table 15.

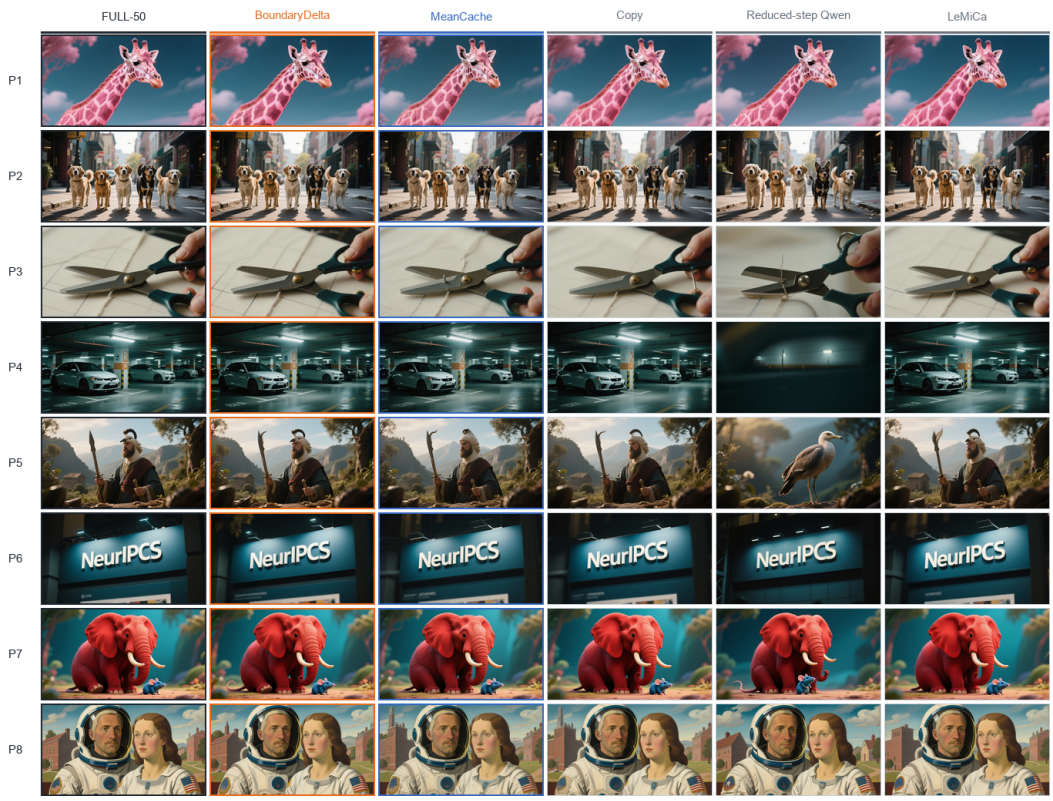


Figure 13: Qwen-Image B17 exact-call qualitative panel for P1–P8. Every row shares its prompt, generation seed 42, and initial latent across FULL-50 and the five exact-NFE methods.

Table 15: Prompt key for the deterministic category-diverse Qwen qualitative panels. All rows use generation seed 42. P1–P5 appear in the main all-method display; P1–P6 appear in the appendix budget-progression display; P1–P8 appear in the appendix all-method displays.

ID	Category	Prompt
P1	Colors	A pink colored giraffe.
P2	Counting	Five dogs on the street.
P3	Descriptions	An instrument used for cutting cloth, paper, and other thin material, consisting of two blades laid one on top of the other and fastened in the middle so as to allow them to be opened and closed by a thumb and finger inserted through rings on the end of their handles.
P4	Misspellings	Pafrking metr.
P5	Rare Words	Peristeronic.
P6	Text	A sign that says 'NeurIPS'.
P7	DALL-E	An illustration of a large red elephant sitting on a small blue mouse.
P8	Reddit	A painting by Grant Wood of an astronaut couple, american gothic style.



Figure 14: Qwen-Image adaptive-method qualitative panel for the same P1–P8 rows. These methods are shown at their realized FEC near the B17 target rather than as exact 17-call runs.

Figure 15 asks a different question by holding each prompt fixed while varying the BoundaryDelta budget. Each column uses the frozen budget-specific mask listed in Appendix B; the masks are independently selected rather than nested prefixes.

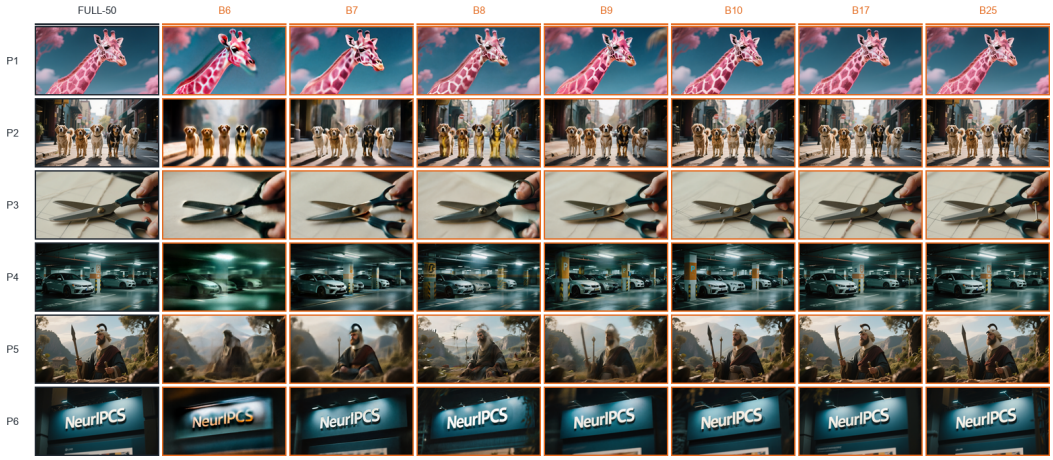


Figure 15: Multi-prompt Qwen-Image BoundaryDelta budget progression. Columns show the paired FULL-50 output followed by B6–B10, B17, and B25 for P1–P6. Each row uses one prompt, seed, and initial latent throughout.