RESPECT: RESIDUAL SPECTRAL PREDICTION FOR TRAINING-FREE ACCELERATION OF DIFFUSION TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion transformers underlie current image and video generation models, but incur substantial sampling costs due to iterative denoising. Training-free caching accelerates sampling through two coupled decisions: which steps to evaluate, and what to substitute for the skipped features. Many existing schedules gate on raw distances between adjacent features, a drift signal with poor signal-to-noise ratio, where stochastic variation masks meaningful drift, while ignoring how the noise level itself evolves along the sampling trajectory. Many existing reconstructors, in turn, rely on local approximation, copying the most recently computed features or extrapolating from a short local history. This approach causes approximation errors to increase with the skip length and sample quality to degrade at high speedups. We propose Residual Spectral Prediction for Efficient Caching of Transformers **(ReSPect)**, a training-free cache that jointly addresses these limitations through spectrally aligned reuse decisions and residual trajectory prediction. Scheduling decisions are based on a spectrally aligned representation obtained from a timestep-dependent filter derived from the linear minimum mean square error (linear-MMSE) denoising formulation, which preserves content-relevant components while suppressing noise. Skipped residuals, rather than hidden states, are forecast by a weighted sum of ridge-regularized Chebyshev regression over the accumulated residual history, which captures the global trajectory, and a first-order discrete Taylor correction, which captures local dynamics. Analysis shows the forecast error admits a skip-length-independent bound, whereas the corresponding bounds for copy-reuse and Taylor extrapolation grow with skip length. Across extensive experiments on FLUX.1-dev, Wan2.1-1.3B, and HunyuanVideo, ReSPect achieves speedups of $3.30\times$, $2.63\times$, and $2.77\times$ over full denoiser evaluation while achieving lower output error than existing caching baselines at matched or lower compute.

1 INTRODUCTION

Diffusion transformers (Peebles and Xie, 2023) form the backbone of many text-to-image and text-to-video models (Black Forest Labs, 2024; Wan et al., 2025; Kong et al., 2024), and sampling them requires tens of sequential evaluations of a large transformer network per output (Ho et al., 2020; Song et al., 2021). Feature caching reduces this cost without retraining by exploiting redundancy along the sampling trajectory.

A cache consists of two decisions at every step: a *schedule* determines whether the network is evaluated, and a *reconstructor* determines what is written to the features when it is not. The two are coupled through the skip length. The schedule decides *where* the model can skip without changing the generated content; the reconstructor decides *how far* a skip can extend before the substituted features depart from the uncached trajectory. Improving either in isolation leaves the other as the bottleneck: a content-aware gate cannot preserve quality on long skips if the skipped features remain a stale copy, and a long-horizon predictor cannot be applied aggressively if the gate responds to noise rather than to content.

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



Figure 1: Qualitative visual results of ReSPect on FLUX.1-dev, maintaining high image fidelity at a $2.43\times$ acceleration.

Many existing schedules gate on the raw distance between consecutive features (Liu et al., 2025a; Bu et al., 2025), which mixes meaningful evolution with high-frequency noise; others use motion cues (Kahatapitiya et al., 2025), frequency bands (Liu et al., 2025c), finer granularity (Chen et al., 2024; Zou et al., 2025), or spectrally filtered distances (Chung et al., 2026). The relevant insight is that a timestep-dependent spectral filter isolates content-sensitive drift, so the gate responds to changes in the generated content rather than to stochastic variation (Sec. 3.2). Many existing reconstructors rely on local temporal information, such as feature reuse (Selvaraju et al., 2024; Zhao et al., 2024) or finite-difference extrapolation (Liu et al., 2025b), while global trajectory fitting methods still predict the full hidden state (Han et al., 2026). Rather than forecasting the hidden state itself, we forecast only the transformer residual, because the block input at skipped steps is known exactly. This removes the predictable component from the regression target and yields a smaller, centered trajectory (Sec. 3.5).

We present **ReSPect** (Residual Spectral Prediction for Efficient Caching of Transformers), a training-free cache that models cache decisions and skipped-step dynamics from spectrally filtered drift and residual trajectory history (Fig. 2). The resulting predictor models the residual trajectory from the full observed history, combining a global Chebyshev estimate with a local Taylor correction. Both the blend and the warm-up are analytically determined (Sec. 3.6).

Our contributions are as follows.

- We formulate cache reconstruction as residual-trajectory prediction and rank reconstructors by their error bounds linear in the skip length for copy-reuse, order $P{+}1$ for ideal order-P Taylor, skip-length independent for a ridge-regularized Chebyshev fit (Han et al., 2026) and refine the Taylor case for the non-uniform evaluation nodes a dynamic gate produces (Sec. 3.3).
- We show that state and residual forecasts share the same error identity on a skipped step, but that residuals are substantially smaller in norm, which tightens the constant B in the ridge analysis and motivates residual-space prediction (Sec. 3.5).
- We derive the blend weight and the warm-up length in closed form from bias–variance and degrees-of-freedom arguments, eliminating two hyperparameter searches (Sec. 3.6).

Coupling the residual forecaster with a spectrally aligned, content-aware schedule, we demonstrate improved quality-compute trade-offs against published caching baselines on FLUX.1-dev, Wan2.1-1.3B, and HunyuanVideo, with every comparison made at matched or lower compute (Sec. 4).

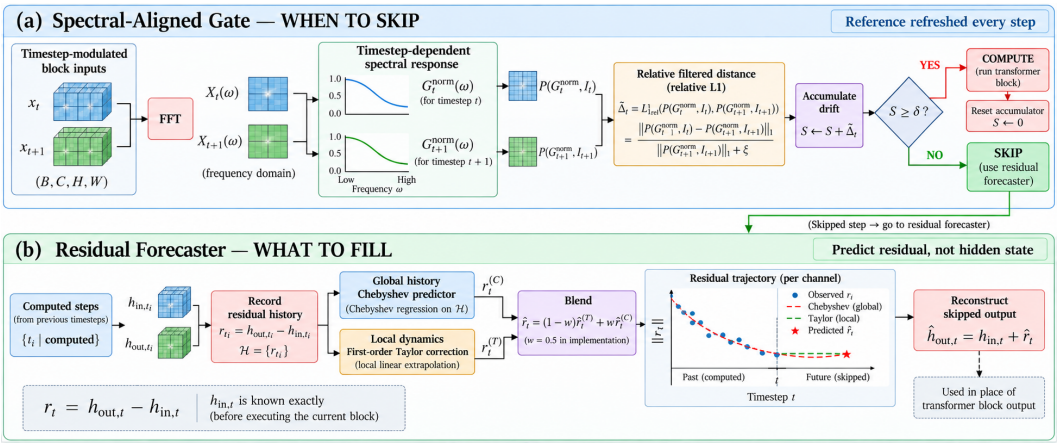

Figure 2: **Overview of ReSPect.** (a) Spectrally aligned gate for adaptive compute decisions. (b) Residual forecaster combining global Chebyshev prediction with a local first-order Taylor correction.

2 Related Work

Accelerating diffusion sampling. Diffusion models (Ho et al., 2020; Song et al., 2021) and latent diffusion models (Rombach et al., 2022), together with transformer-based denoisers (Peebles and Xie, 2023), form the foundation of many modern text-to-image (Black Forest Labs, 2024) and text-to-video models (Wan et al., 2025; Kong et al., 2024). A broad range of approaches has been developed to reduce their inference cost, including distillation (Song et al., 2023; Yin et al., 2024), quantization (Li et al., 2023; Zhao et al., 2024), efficient attention mechanisms, and accelerated numerical solvers (Song et al., 2021; Lu et al., 2022). These approaches typically require additional training, model modification, or changes to the sampling procedure. Feature caching is complementary to these approaches: it reduces redundant network evaluations during sampling without retraining the underlying generative model.

Skip schedules. Early caches evaluate the network at fixed intervals (Selvaraju et al., 2024; Zhao et al., 2024). Dynamic schedules instead gate on a redundancy signal: the distance between consecutive features (Liu et al., 2025a; Bu et al., 2025), motion (Kahatapitiya et al., 2025), or frequency-band energy (Liu et al., 2025c), while block-level (Chen et al., 2024) and token-level (Zou et al., 2025) caches vary the granularity of reuse. SeaCache (Chung et al., 2026) improves the scheduling signal by measuring redundancy after a timestep-aware spectral filter, but retains feature reuse for skipped steps.

Skip reconstructors. Copy-reuse is zero-order prediction; TaylorSeer (Liu et al., 2025b) extrapolates hidden states from finite differences of recent evaluations (local forecasting); Spectrum (Han et al., 2026) regresses hidden-state trajectories on Chebyshev bases (long-range forecasting). SeaCache and Spectrum are the closest prior methods from complementary directions: spectrally aligned scheduling (when to evaluate) and Chebyshev forecasting (what to substitute). ReSPect differs on both axes: it forecasts the transformer residual rather than the full hidden state, blends the global Chebyshev fit with a local Taylor correction, and couples the predictor with a content-aware spectral schedule (Sec. 3).

3 Method

3.1 Problem Setup

We consider a latent generative model whose sampling trajectory $\{x_t\}_{t=0}^{T}$ is produced by a discrete solver. Here $t \in \{0, \ldots, T\}$ denotes the sampling/solver index, and the sequence is traversed in the model's reverse or flow direction, from the initial noisy state toward the clean output. Each step evaluates a transformer network, so sampling costs T forward passes.

At each solver step t, the current latent x_t is mapped to the input $I_t = \phi(x_t, t)$ of a transformer block, whose output $h_{\text{out}}(t)$ is produced from input $h_{\text{in}}(t)$. The block residual

$$r(t) = h_{\text{out}}(t) - h_{\text{in}}(t) \tag{1}$$

induces a second trajectory $\{r(t)\}$ alongside $\{x_t\}$. Adjacent solver states often exhibit substantial temporal redundancy, and this redundancy propagates to intermediate transformer features.

To derive the spectral gate we model the latent at step t as the linear mixture

$$x_t = a_t x_0 + b_t \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, I), \tag{2}$$

where x_0 is the clean latent, (a_t, b_t) are the signal and noise coefficients at that solver step, and ε is white and independent of x_0. This is a signal/noise model for the gate, not a claim that every evaluated architecture follows this exact probabilistic forward process; flow schedules enter by reading (a_t, b_t) from the solver (Appendix A).

A cache consists of two decisions at every step. A *schedule* determines whether the network is evaluated at step t, and a *reconstructor* determines what is written to the features when it is not. On an evaluated step the identity $h_{\text{out}}(t) = h_{\text{in}}(t) + r(t)$ holds by definition. On a skipped step the input $h_{\text{in}}(t)$ is known, and the reconstructor supplies $\hat{r}(t)$, yielding

$$\hat{h}_{\text{out}}(t) = h_{\text{in}}(t) + \hat{r}(t). \tag{3}$$

The reconstruction error therefore coincides with the residual error,

$$\hat{h}_{\text{out}}(t) - h_{\text{out}}(t) = \hat{r}(t) - r(t). \tag{4}$$

ReSPect specifies both decisions: a spectrally aligned gate (Sec. 3.2) and a residual forecaster (Secs. 3.3–3.6); Fig. 2 gives an overview.

3.2 Spectrally aligned gate: when to compute

A dynamic gate skips the network while the filtered representation changes slowly and triggers a full evaluation once the accumulated drift exceeds a threshold δ. Because the gate is evaluated at every solver step, it must incur negligible overhead. We therefore compute it from the first-block input $I_t = \phi(x_t, t)$, which is available before executing the transformer blocks, and measure the relative ℓ_1 distance between consecutive steps, as in prior distance-gated caches (Liu et al., 2025a):

$$\Delta_t = \mathrm{L1}_{\text{rel}}(I_t, I_{t+1}) = \frac{\|I_t - I_{t+1}\|_1}{\|I_{t+1}\|_1 + \xi}. \tag{5}$$

This raw distance is a noise-contaminated drift: under the mixture above, the noise contribution can dominate the raw distance at early steps, so Δ_t responds to stochastic variation on which no cache decision should depend. We therefore pass from the raw representation to a spectrally filtered drift, measuring distance on the component of I_t that is aligned with the clean signal. The linear-MMSE estimator of x_0 from x_t,

$$h_t^\star = \arg\min_h \|h * x_t - x_0\|_2^2, \tag{6}$$

is convolutional, so its solution is diagonal in the Fourier domain and yields the Wiener frequency response

$$G_t(f) = \frac{a_t S_x(f)}{a_t^2 S_x(f) + b_t^2}, \tag{7}$$

where S_x is the power spectrum of the clean latent. Under a power-law spectral prior $S_x(f) \propto 1/|f|^p$, G_t passes only low frequencies at early steps (small a_t) and admits progressively higher frequencies as sampling proceeds toward the clean output, reproducing the observed spectral evolution along the sampling trajectory, in which coarse structure is formed first and fine detail later (Fig. A1; Appendix A). Applying G_t to I_t in the Fourier domain yields $\mathcal{P}(G_t, I_t) = \mathcal{F}^{-1}\big[G_t \odot \mathcal{F}(I_t)\big]$, a representation whose drift reflects content change with the noise component suppressed.

The raw responses G_t differ in overall gain across t, which would allow high-gain steps to dominate the accumulator. We therefore normalize each response to unit mean gain over radial frequency bins,

$$G_t^{\text{norm}}(f) = \nu_t G_t(f), \quad \nu_t^{-1} = \frac{1}{L} \sum_\ell G_t(f_\ell), \tag{8}$$

so that the accumulator measures changes in spectral shape rather than changes in overall filter gain. Because the optimal spectral response is timestep-dependent, each adjacent pair is compared using its corresponding normalized response G_t^{norm} and G_{t+1}^{norm}:

$$\tilde{\Delta}_t = \mathrm{L1}_{\text{rel}}\big(\mathcal{P}(G_t^{\text{norm}}, I_t), \mathcal{P}(G_{t+1}^{\text{norm}}, I_{t+1})\big) \tag{9}$$

The gate accumulates $\tilde{\Delta}_t$ and evaluates the transformer network when the sum exceeds δ, resetting the accumulator; the first and last steps are always evaluated. Crucially, the reference is refreshed at every step, including skipped steps, so each $\tilde{\Delta}_t$ measures only the incremental drift between consecutive solver states (Sec. 4.4 quantifies the difference). The gate decides *which* steps are evaluated; the remainder of this section specifies *what* is written on the others.

3.3 Predictors on the sampling trajectory

The reconstruction problem is therefore reduced to forecasting the residual trajectory at unevaluated steps. We state the analysis for one residual channel; the predictor is applied independently across channels. Consider a skip from the last evaluated step t_k to the current step t_j. Every reconstructor is a predictor $\hat{r}(t_j)$ built from the residuals recorded at evaluated steps; the candidates differ in the smoothness they assume of r.

Order zero (copy-reuse). $\hat{r} = r(t_k)$. If r is L-Lipschitz, Lipschitz continuity gives

$$|r(t_j) - r(t_k)| \le L\,|t_j - t_k|, \tag{10}$$

which is tight in general and yields a linear error bound in the skip length (App. B). No smoothness assumption beyond Lipschitz continuity can improve the bound, since copy-reuse exploits no higher-order structure.

Order P (Taylor extrapolation). The ideal degree-P Taylor expansion of r about t_k has a Lagrange remainder of order $|t_j - t_k|^{P+1}$ (Han et al., 2026, Thm. 3.1); see App. B. A higher order thus sharpens the bound for short skips — but only as an oracle reference, since the exact derivatives are unavailable in a training-free cache. The implemented predictor interpolates recorded residuals instead, and recovers the $(P{+}1)$-order rate only under comparable node spacing, which the dynamic gate does not guarantee (App. B). Taylor extrapolation is therefore accurate when the skip is short but its remainder grows with the extrapolation distance; it also discards older trajectory information. These bounds motivate a global predictor that can exploit the full observed trajectory rather than only local finite differences; Sec. 3.4 develops such a predictor using a ridge-regularized Chebyshev basis.

3.4 Residual forecasting with Chebyshev bases

To predict $r(t_j)$, we fit the accumulated residual trajectory. Map the solver-step index affinely from the fixed solver-step range to $\tau \in [-1, 1]$, and expand each residual channel (of feature dimension F) in the Chebyshev basis $T_0, \ldots, T_M$ of degree M. Stack the K recorded residuals row-wise into the observation matrix $\mathbf{H} \in \mathbb{R}^{K\times F}$, let $\mathbf{\Phi} \in \mathbb{R}^{K\times(M+1)}$ evaluate the basis at the recorded steps, and regress the coefficient matrix $\mathbf{C} \in \mathbb{R}^{(M+1)\times F}$ with ridge penalty λ:

$$\mathbf{C}^\star = \arg\min_{\mathbf{C}} \|\mathbf{\Phi}\mathbf{C} - \mathbf{H}\|_F^2 + \lambda\|\mathbf{C}\|_F^2 = (\mathbf{\Phi}^\top\mathbf{\Phi} + \lambda I)^{-1}\mathbf{\Phi}^\top\mathbf{H}. \tag{11}$$

Each row of $\mathbf{\Phi}$ evaluates the basis at one recorded step, and the corresponding row of $\mathbf{H}$ holds the recorded residual. The fit is re-solved whenever a new residual is computed; the window retains the last K observations, which exceeds the number of solver steps in all reported runs. The forecast at a skipped step evaluates the fitted expansion at its coordinate, $\hat{r}(t_j) = \sum_{m=0}^{M} \hat{C}_m\, T_m(\tau_j)$. Let $p_M^\star$ denote the exact degree-M Chebyshev truncation of r. Under the standard analyticity assumption that r extends analytically to a Bernstein ellipse of parameter $\rho > 1$ on which $|r| \le B$, classical Chebyshev approximation gives (Han et al., 2026, Thm. 3.2)

$$\|r - p_M^\star\|_\infty \le \frac{2B}{\rho - 1}\,\rho^{-M}, \tag{12}$$

uniformly in the evaluation point (App. B): this truncation term decays with the degree M and is independent of the skip length. Because the coefficients are estimated from finitely many observations, we also control the deviation of the estimator from that ideal truncation.

Lemma 1 (Ridge stability, restated from Han et al. 2026, Thm. 3.3)**.** *Let $\hat{r}$ be the Chebyshev component from equation 11 and $p^\star_M$ the exact degree-M truncation. With $\sigma_{\min} = \sigma_{\min}(\mathbf{\Phi})$, equal to zero whenever $K < M{+}1$, at any forecast coordinate τ_j,*

$$|\hat{r}(\tau_j) - p^\star_M(\tau_j)| \le \varepsilon_M \frac{(M{+}1)K}{\sigma^2_{\min} + \lambda} + \frac{\lambda\sqrt{M{+}1}}{\sigma^2_{\min} + \lambda} \cdot \frac{2B}{\sqrt{1-\rho^{-2}}}, \tag{13}$$

for ε_M the right-hand side of equation 12; with that equation and the triangle inequality this bounds the total error $|r(\tau_j) - \hat{r}(\tau_j)|$ (App. C).

Each observation adds a positive semi-definite rank-one term to $\mathbf{\Phi}^\top\mathbf{\Phi}$, so $\sigma_{\min}$ is non-decreasing in K and the factor $K/(\sigma^2_{\min} + \lambda)$ stays bounded rather than growing; before K reaches $M{+}1$ the design is rank-deficient and λ alone controls the bound, at the price of bias. Sec. 3.6 turns this stability condition into an explicit warm-up threshold.

3.5 Residual space

On a skipped step, the block input h_{in} is known exactly from the current sampling state and the preceding computation; it therefore does not need to be forecast. For a state forecast $\hat{h}_{\text{out}}$ and a residual forecast $\hat{r}$,

$$\hat{h}_{\text{out}} - h_{\text{out}} = (\hat{r} + h_{\text{in}}) - (r + h_{\text{in}}) = \hat{r} - r. \tag{14}$$

Thus, once the predictor is fixed, state and residual formulations have identical reconstruction error; they differ only in their regression target. The full hidden state carries a large component that is already determined by h_{in}, so fitting it forces the regression to reproduce variation the sampler already knows and inflates the target scale. The residual trajectory is substantially smaller in norm, so comparable relative fit quality yields smaller absolute error and tighter constants in the ridge analysis (Lemma 1), whose constant B scales with the target norm. This comparison takes the residual trajectory to satisfy the same analyticity condition with comparable ρ; consistent with this, the M-sweep in Tab. 3 saturates at the same small M used throughout. The skipped-step update is simply $h \leftarrow h + \hat{r}$: the forecast supplies the missing transformer increment, and no full network evaluation is needed.

3.6 Analytic blend and warm-up

Blend. For the risk model, we assign the Taylor predictor's systematic error to bias b^2_T and the Chebyshev estimator's finite-history uncertainty to variance σ^2_C, and omitting w-independent terms, the risk of the convex combination $\hat{r}_w = (1-w)\hat{r}_T + w\hat{r}_C$ is

$$R(w) = (1-w)^2 b^2_T + w^2\sigma^2_C, \tag{15}$$

minimized at $w^\star = b^2_T/(b^2_T + \sigma^2_C)$. Early in sampling the drift is fast (b_T large) and the fit has seen few points (σ_C large); later both decrease as the trajectory settles and observations accumulate. Thus Taylor is relatively more reliable during the early sparse-history regime, while the global Chebyshev predictor becomes increasingly reliable as observations accumulate. Because the bias and variance terms are not observable at inference time, we cannot evaluate $w^\star$ online and instead use the symmetric fixed weight $w{=}0.5$, which introduces no additional inference-time adaptation or hyperparameter search. The ablations in App. H show that this choice remains robust across models and budgets, staying close to the stronger component on each model, whereas $w{=}1$ removes the low-bias local term precisely when the polynomial fit is least determined.

Warm-up. With $n < M{+}1$ observations, equation 11 is rank-deficient ($\sigma_{\min} = 0$) and the penalty dominates the data, so ReSPect uses the discrete Taylor step alone: exact on the most recent points by construction, it reduces to copy-reuse at $n = 1$ and extrapolates linearly from $n = 2$ onward. In practice the regularized fit is determined one observation early, so blending starts at $n \ge 4$ (with $M{=}4$) rather than $M{+}1$.

Compute and memory. Skip decisions accumulate per sample, so prompts in a batch never share a schedule. Each sample stores one $(M{+}1) \times F$ coefficient matrix and K recorded residuals, i.e. $\mathcal{O}((M{+}1)F + KF)$ memory and $\mathcal{O}(K(M{+}1)F)$ fitting cost with $M \ll K \ll F$, which is negligible relative to a transformer block ($\sim$0.09 s per image in our runs).

Table 1: FLUX.1-dev, DrawBench 200, 1024×1024, 50 steps. Speedup is Original latency (5.31 s) over row latency. ReSPect spends equal or less compute than every baseline in its block.

Method	PSNR↑	SSIM↑	LPIPS↓	Lat. (s)	Speedup
Original (50 steps)	–	–	–	5.31	1.00×
Vanilla 25 steps	15.55	0.668	0.409	2.72	1.95×
TeaCache (δ=0.3)	20.76	0.810	0.211	2.85	1.86×
TaylorSeer ($\mathcal{S}$=3)	22.78	0.828	0.163	2.52	2.11×
ReSPect (δ=0.37)	**26.24**	**0.895**	**0.093**	**2.19**	2.43×
Vanilla 15 steps	17.84	0.740	0.305	1.68	3.16×
ToCa	18.40	0.700	0.324	2.25	2.36×
TeaCache (δ=0.6)	17.21	0.714	0.348	1.72	3.09×
TaylorSeer ($\mathcal{S}$=5)	19.97	0.762	0.236	1.95	2.72×
ReSPect (δ=0.6)	**21.63**	**0.823**	**0.178**	**1.61**	3.30×

4 Experiments

4.1 Settings

We evaluate FLUX.1-dev (Black Forest Labs, 2024) on 200 DrawBench prompts (Saharia et al., 2022) (1024×1024, 50 steps) and HunyuanVideo (Kong et al., 2024) and Wan2.1-1.3B (Wan et al., 2025) on 946 VBench prompts (Huang et al., 2024) (480p, 65 frames, 50 steps). Metrics are PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018) against the uncached same-seed reference. All qualities and latencies are measured on a single NVIDIA B200 (bf16). Each ReSPect operating point uses equal or less compute than the baseline row it is compared against. Protocol details are in Appendix G.

4.2 Text-to-image

On FLUX (Tab. 1), ReSPect attains 26.24 dB at δ=0.37, +3.46 dB over TaylorSeer at lower latency (2.19 s vs. 2.52 s), and 21.63 dB at δ=0.6, +1.66 dB while faster (1.61 s vs. 1.95 s). Both points dominate the published quality–compute frontier (Fig. 4). Fig. 3 shows two DrawBench prompts at the aggressive budget.

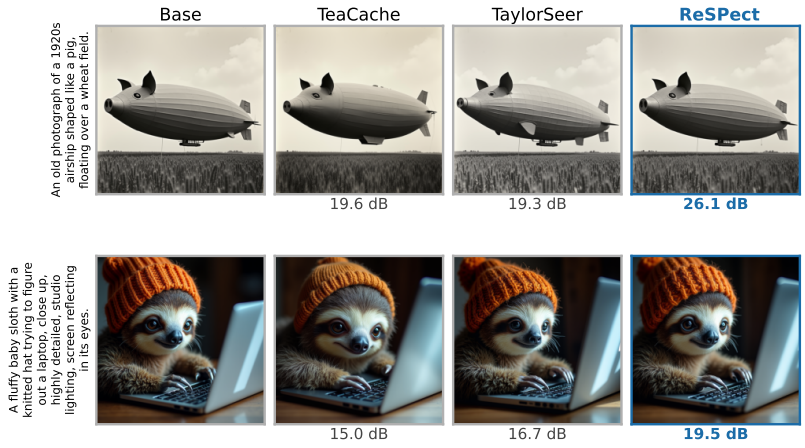

Figure 3: FLUX.1-dev, δ=0.6. PSNR vs. the uncached image under each panel. TeaCache and TaylorSeer use equal or more compute.

4.3 Text-to-video

Gains are largest on video (Tab. 2). On Wan, ReSPect reaches 32.21 dB at δ=0.17 (+11.4 dB over TeaCache) and 27.20 dB at δ=0.31 (+8.3 dB); LPIPS is less than a third of any baseline. On

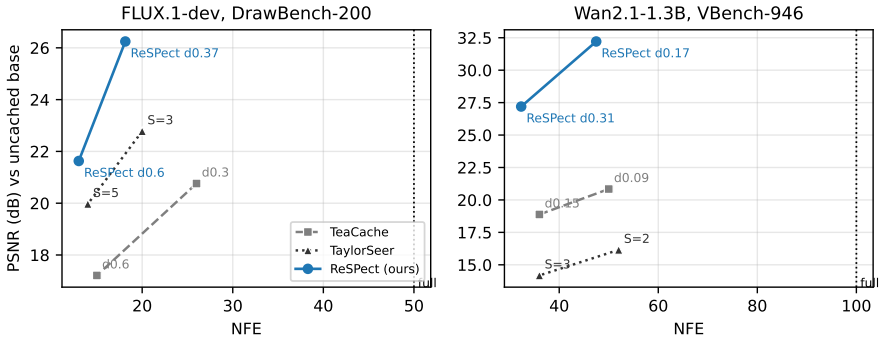

Figure 4: PSNR versus compute. ReSPect is compared with published TeaCache and TaylorSeer points; dotted line: uncached compute. The Wan aggressive-budget gain is $+8.32$ dB.

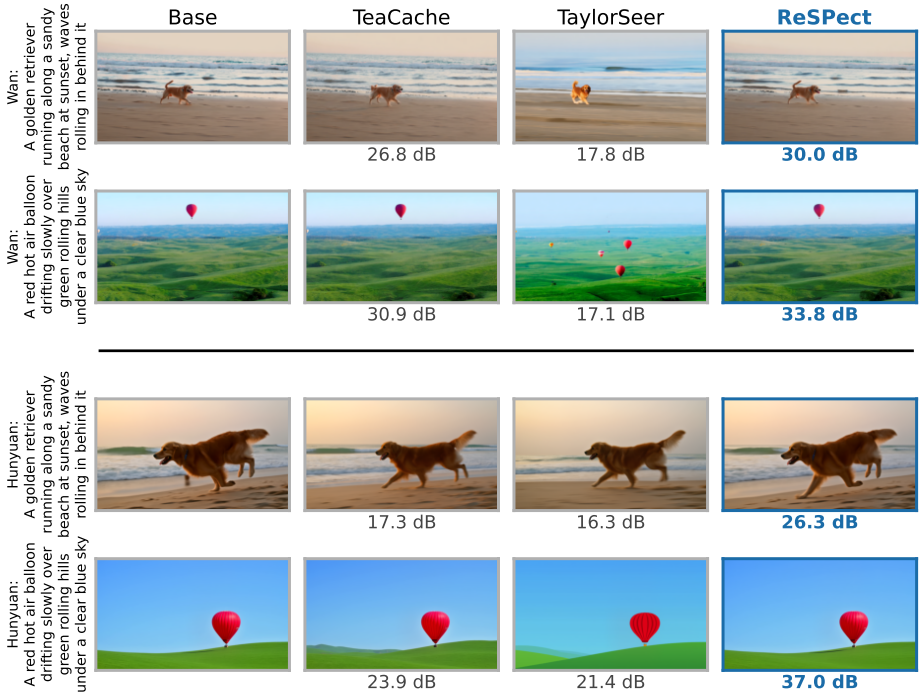

Figure 5: Opening frames on Wan and HunyuanVideo. ReSPect uses δ=0.31/0.48 with less compute. PSNR is shown below each panel; full strips are in App. I.

HunyuanVideo the margins are $+9.9$ dB and $+8.8$ dB at δ=0.27 and 0.48. Fig. 5 shows opening frames; 65-frame strips are in App. I.

4.4 Ablations

On FLUX δ=0.6 (first 20 prompts, fixed across arms with one shared base run), varying M and λ around the defaults yields a flat optimum (Tabs. 3–4): $M \in \{4, 6\}$ and $\lambda \in \{0.01, 0.1\}$ agree to 0.15 dB. Refreshing the gate reference at every step rather than only at evaluated steps contributes $+2.50$ dB on HunyuanVideo at matched compute. Further reconstructor ablations are in App. H.

5 Conclusion

We introduced ReSPect, a training-free cache for diffusion transformers that combines spectrally aligned scheduling with residual-space forecasting. A linear-MMSE-motivated gate selects eval-

Table 2: Text-to-video results on VBench (480p, 65 frames, 50 steps). Latencies are measured on one NVIDIA B200 (bf16); speedup is relative to the Original latency (Hunyuan 77.3 s, Wan 28.8 s).

HunyuanVideo (Orig. 77.3 s)

Method	PSNR↑	SSIM↑	LPIPS↓	Lat. (s) / Speedup
Original	–	–	–	77.3 / 1.00×
Vanilla 25	19.97	0.731	0.263	38.7 / 2.00×
TeaCache 0.12	23.40	0.805	0.133	42.0 / 1.84×
TaylorSeer S2	24.14	0.820	0.152	42.0 / 1.84×
ReSPect 0.27	**34.07**	**0.946**	**0.046**	39.9 / 1.94×
Vanilla 15	17.49	0.662	0.371	23.2 / 3.33×
TeaCache 0.2	20.42	0.734	0.172	30.2 / 2.56×
TaylorSeer S3	20.42	0.733	0.242	30.2 / 2.56×
ReSPect 0.48	**29.20**	**0.891**	**0.100**	27.9 / 2.77×

Wan2.1-1.3B (Orig. 28.8 s)

Method	PSNR↑	SSIM↑	LPIPS↓	Lat. (s) / Speedup
Original	–	–	–	28.8 / 1.00×
TeaCache 0.09	20.84	0.721	0.171	14.9 / 1.93×
TaylorSeer S2	16.15	0.543	0.336	15.3 / 1.88×
ReSPect 0.17	**32.21**	**0.947**	**0.033**	14.8 / 1.95×
TeaCache 0.15	18.88	0.645	0.245	10.4 / 2.77×
TaylorSeer S3	14.18	0.453	0.455	11.2 / 2.57×
ReSPect 0.31	**27.20**	**0.888**	**0.082**	10.9 / 2.63×

Table 3: M sweep on FLUX δ=0.6 (n=20), λ=0.1 fixed. Default is the first row.

Setting	PSNR↑	SSIM↑
Default (M=4)	23.57	0.852
M=2	23.02	0.845
M=6	23.72	0.855

Table 4: λ sweep on FLUX δ=0.6 (n=20), M=4 fixed. Default is the first row.

Setting	PSNR↑	SSIM↑
Default (λ=0.1)	23.57	0.852
λ=0.01	23.56	0.853
λ=1.0	23.38	0.847

uation steps, while skipped residuals are reconstructed using a global Chebyshev fit with a local first-order Taylor correction. The blend and warm-up are motivated by bias-variance and degrees-of-freedom analyses. At matched or lower compute, ReSPect improves over the strongest compared baseline by up to 11.4 dB on Wan2.1, 9.9 dB on HunyuanVideo, and 3.46 dB on FLUX, with the largest gains arising under longer skips, particularly for video generation.

Limitations. Gains are largest when many steps are skipped; under rapid content drift the gate evaluates densely and little residual error remains for any predictor to reduce. The blend weight is fixed at w=0.5 rather than scheduled. Since content-aware gates need not realize equal skip rates at a shared δ, we compare at matched or lower compute.

AI USE STATEMENT

Generative AI was used for coding support, proofreading, and LaTeX formatting. All AI-assisted code, derivations, claims, and experimental results were independently checked by the authors. Generative AI was not used to generate research ideas or fabricate data. The authors take full responsibility for the final work.

ETHICS STATEMENT

This work accelerates inference of publicly available generative models using public benchmarks (DrawBench, VBench) and introduces no new data collection, human subjects, or disallowed content. Faster sampling lowers the cost of generating synthetic media, with the usual dual-use considerations shared by all generative modeling research.

REPRODUCIBILITY STATEMENT

ReSPect is fully specified in Sec. 3, with M=4, K=100, λ=0.1, w=0.5, first-order discrete Taylor prediction, and warm-up $n \geq 4$. Experimental settings are detailed in Sec. 4.1 and App. G, with theoretical proofs in Apps. A-E. Source code and evaluation scripts will be released.

REFERENCES

Black Forest Labs. FLUX.1 [dev]. https://github.com/black-forest-labs/flux, 2024.

Jiazi Bu, Pengyang Ling, Yujie Zhou, et al. DiCache: Let diffusion model determine its own cache. *arXiv:2508.17356*, 2025.

Pengtao Chen, Mingzhu Shen, Peng Ye, et al. Δ-DiT: A training-free acceleration method tailored for diffusion transformers. *arXiv:2406.01125*, 2024.

Jiwoo Chung, Sangeek Hyun, MinKyu Lee, Byeongju Han, Geonho Cha, Dongyoon Wee, Youngjun Hong, and Jae-Pil Heo. SeaCache: Spectral-evolution-aware cache for accelerating diffusion models. In *CVPR*, pages 14283–14294, 2026. *arXiv:2602.18993*.

Jiaqi Han, Juntong Shi, Puheng Li, Haotian Ye, Qiushan Guo, and Stefano Ermon. Spectrum: Adaptive spectral feature forecasting for diffusion sampling acceleration. In *CVPR*, 2026. *arXiv:2603.01623*.

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

Ziqi Huang, Yinan He, Jiashuo Yu, et al. VBench: Comprehensive benchmark suite for video generative models. In *CVPR*, 2024.

Kumara Kahatapitiya, Haozhe Liu, Sen He, et al. Adaptive caching for faster video generation with diffusion transformers. In *ICCV*, 2025.

Weijie Kong, Qi Tian, Zijian Zhang, et al. HunyuanVideo: A systematic framework for large video generative models. *arXiv:2412.03603*, 2024.

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

Feng Liu, Shiwei Zhang, Xiaofeng Wang, et al. Timestep embedding tells: It's time to cache for video diffusion model. In *CVPR*, 2025.

Jiacheng Liu, Chang Zou, Yuanhuiyi Lyu, et al. From reusing to forecasting: Accelerating diffusion models with TaylorSeers. *arXiv:2503.06923*, 2025.

Jiacheng Liu, Peiliang Cai, Qinming Zhou, et al. FreqCa: Accelerating diffusion models via frequency-aware caching. *arXiv:2510.08669*, 2025.

Cheng Lu, Yuhao Zhou, Fan Bao, et al. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *NeurIPS*, 2022.

Z. Ma, L. Wei, F. Wang, S. Zhang, and Q. Tian. MagCache: Fast video generation with magnitude-aware cache. In *NeurIPS*, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, et al. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022.

Pratheba Selvaraju, Tianyu Ding, Tianyi Chen, et al. FORA: Fast-forward caching in diffusion transformer acceleration. *arXiv:2407.01425*, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *ICML*, 2023.

Wan Team. Wan: Open and advanced large-scale video generative models. *arXiv:2503.20314*, 2025.

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

Tianwei Yin, Michaël Gharbi, Richard Zhang, et al. One-step diffusion with distribution matching distillation. In *CVPR*, 2024.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Xuanlei Zhao, Xiaolong Jin, Kai Wang, and Yang You. Real-time video generation with pyramid attention broadcast. *arXiv:2408.12588*, 2024.

Tianchen Zhao, Tongcheng Fang, Haofeng Huang, Enshu Liu, Rui Wan, Widyadewi Soedarmadji, Shiyao Li, Zinan Lin, Guohao Dai, Shengen Yan, et al. ViDiT-Q: Efficient and accurate quantization of diffusion transformers for image and video generation. *arXiv:2406.02540*, 2024.

Chang Zou, Xuyang Liu, Ting Liu, Siteng Huang, and Linfeng Zhang. Accelerating diffusion transformers with token-wise feature caching. In *ICLR*, 2025.