CachedSearch: Training-Free Cached Exploration for Test-Time Search in Video Diffusion

Anonymous authors
Paper under double-blind review

Abstract

Test-time search reliably improves a pretrained video diffusion model, sample N candidates, score them with a verifier, and keep the best. But it is wasteful as every candidate is fully denoised, while all but one are discarded. We observe that search uses only the verifier's ranking of discarded candidates, not their pixels. A cheap, lossy rollout that preserves this ranking can therefore replace full generation during search. Training-free feature caching provides such a rollout. On 7,552 seed-matched cached and full-compute Wan2.1-T2V-1.3B videos on VBench, caching that halves candidate cost preserves verifier rankings with a median per-prompt Spearman correlation of 0.905. Its errors occur mostly among nearly tied candidates, yielding zero median regret. Under a Gaussian-copula model, the fraction of search gain preserved under ranking noise equals the latent rank correlation for every N, so cheaper candidates can widen search without compounding error. CachedSearch explores all candidates with aggressive caching, then regenerates only the winner at full compute from its seed, always returning an exact full-compute sample. At $N = 8$, it retains 94% of best-of-8's reward gain at 63% of its cost. For under 2% more cost than best-of-4, it searches six candidates and achieves 11% more gain on VBench; with candidate pruning, it explores candidates $3.12\times$ more cheaply than best-of-8. CachedSearch requires no training and applies to any sample-and-rank search with a training-free cache and verifier. A single threshold, calibrated on 50 prompts, retains at least 85% of the gain with 1.8–$2.4\times$ cheaper candidates on five of six models across four families, ranging from 1.3B to 14B parameters. Code will be released.

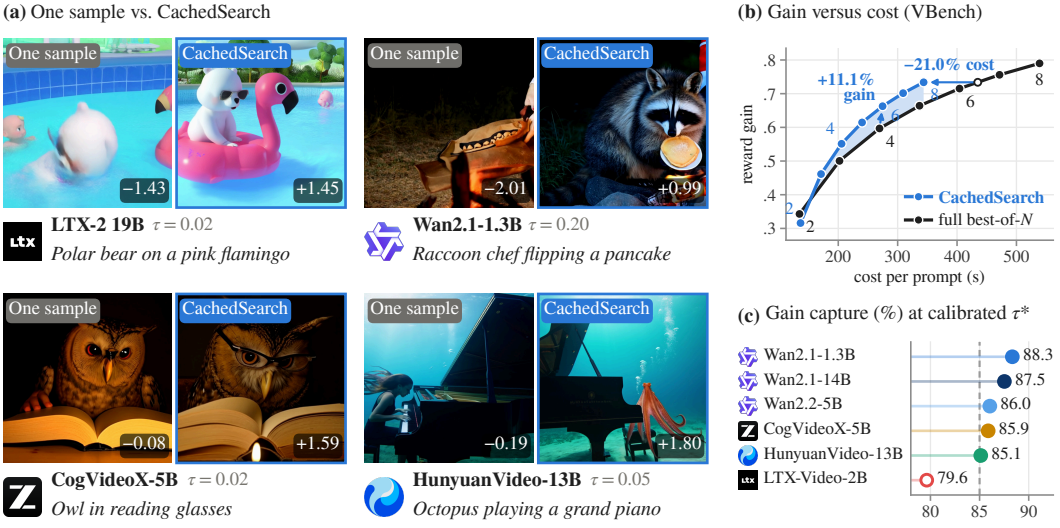


Figure 1: (a) Left: One sample from seed of 8. Right: full-compute video of the seed that CachedSearch selects. (b) Gain over one sample vs. cost on the VBench suite (Wan2.1-T2V-1.3B, τ=0.10; labels give N; axes start at N=2). (c) Mean per-prompt gain capture at each model's τ^* (dashed: 85%).

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.

1 Introduction

Test-time search has become a standard way to improve a pretrained diffusion model, sample N candidates from different seeds, score them with a verifier, and return the best (Ma et al., 2025; Liu et al., 2025b; He et al., 2025). With a modest budget, search can let a small model match a much larger one (Ma et al., 2025; He et al., 2025). For video, however, the cost is steep. One 81-frame, 50-step rollout of Wan2.1-T2V-1.3B (Wan Team, 2025a) takes 68 s on a GH200 GPU, so best-of-8 spends nine minutes per prompt and discards seven of eight generated videos; even pruning-based video searches cost 2–10× a single sample (Liu et al., 2025b; Zhao et al., 2026). Training-free feature caching makes each rollout 2–3× cheaper at near-lossless fidelity by reusing transformer computation across adjacent denoising steps (Zhao et al., 2024; Liu et al., 2024; Lv et al., 2024; Zhou et al., 2025; Gao et al., 2025). Yet caches are typically judged by how closely a cached video matches its full-compute reference, which is the wrong criterion for search. Search never uses the pixels of discarded candidates; it uses only the verifier's ranking. A cache may change every video without hurting search if it preserves their order, while small changes that reorder the leaders can erase the gain.

We therefore conduct a first ranking fidelity, and find that caching preserves it surprisingly well. We generate each candidate twice from the same seed, once cached and once at full compute, and score both with ImageReward (Xu et al., 2023). Across 7,552 seed-matched pairs on VBench (Huang et al., 2024), a cache that halves candidate cost preserves the within-prompt ranking with median Spearman $\rho = 0.905$ and selects the true winner on 72% of prompts (Section 4.2). Its mistakes are also mild, they concentrate on prompts with nearly tied candidates, yielding zero median regret. A simple result explains why this suffices. If cached and full scores share a Gaussian copula with correlation r, then the fraction of search gain preserved by cached selection equals r for every N (Theorem 1). Thus ranking noise removes a fixed fraction of the gain rather than compounding as search widens.

This observation leads to CachedSearch, which explores cheaply and commits at full compute (Figure 3). All N candidates are generated with aggressive caching and scored, after which only the winning seed is regenerated at full compute. Because generation is deterministic given the seed, the delivered video is always an unmodified full-compute sample, and exactly matches full-compute search whenever both select the same winner. At N=8, CachedSearch reduces an eight-candidate search from nine minutes per prompt to under six, or 63% of the cost, while retaining 94.0% of best-of-8's reward gain (Figure 1b). The saved compute can instead widen search: for under 2% more than full best-of-4, CachedSearch searches six candidates and delivers 11% more gain on VBench, close to the 12% predicted by theory. Combined with candidate pruning, it explores 3.12× more cheaply than best-of-8 (Section 5).

CachedSearch is directly deployable, it changes only candidate rollout, so it works with any sample-and-rank search, training-free cache, and verifier, without modifying model weights or the sampler. Adoption requires only a cache wrapper and one threshold calibrated on 50 prompts (Section 3.2). The absolute saving grows with model cost, from 6 s per candidate on LTX-Video-2B to 174 s on Wan2.1-14B (Section 4.4). More broadly, ranking fidelity suggests a different target for search-time acceleration: approximations need to preserve candidate order, not candidate pixels. Section 6 contrasts concurrent image work on cheap drafts (Rawal et al., 2026). Contributions:

- We identify ranking fidelity as the key requirement for cheap exploration and measure it with a seed-matched protocol using rank correlation, top-1 agreement, regret, and gain capture on over 15,000 paired rollouts from three prompt suites. At the same 2× saving, caching preserves the ranking while step truncation and reduced resolution do not (Sections 2 and 4.2).
- We prove that under a Gaussian copula, gain capture equals the latent rank correlation for every N, with closed forms for regret, break-even width, and when cheaper, wider search wins at equal cost. The measured correlation predicts the observed capture without fitting (Section 3.3).
- We propose CachedSearch, a training-free, drop-in replacement for best-of-N rollout that always returns an exact full-compute sample, retains 94% of best-of-8's gain at 63% of its cost, beats full search at near-equal cost, and composes with pruning (Sections 4.3 and 5).
- We characterize its operating range: four published caches serve as interchangeable exploration engines, all cheaper than full best-of-8, while one threshold per model retains at least 85% of the gain on five of six models across four families; LTX-Video marks where the method stops paying off (Section 4.4).

Table 1: Three cheap rollouts as exploration on Wan2.1-T2V-1.3B, each scored against the same 480×832, T=50 full-compute references (50 prompts, 8 seeds); capture is the mean per-prompt ratio. Shaded: our default.

exploration rollout	speedup ↑	med ρ ↑	p10 ρ ↑	top-1 ↑	regret ↓	capture ↑
caching (τ=0.10)	1.97×	**0.905**	**0.614**	**64%**	**0.039**	**90.1%**
step truncation (T=25)	1.98×	0.512	0.024	44%	0.226	72.6%
reduced resolution (352×608)	2.33×	0.060	−0.410	18%	0.627	−0.8%

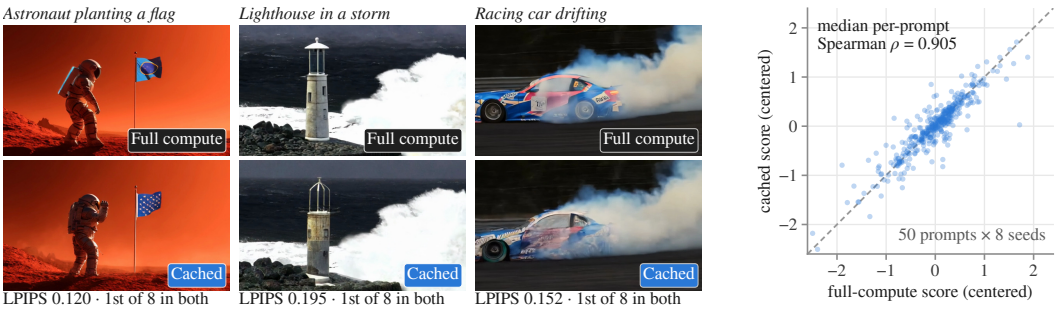


Figure 2: Left: the same Wan2.1-T2V-1.3B seed at full compute and under caching (τ=0.10), same frame and crop; below each pair, its video LPIPS and its rank among the prompt's 8 seeds under full and cached ImageReward. Right: cached against full-compute ImageReward, centered per prompt.

2 Background and Motivation

A latent video diffusion transformer v_θ trained with flow matching (Wan Team, 2025a) generates a video by integrating a deterministic T-step probability-flow ODE under classifier-free guidance, requiring $2T$ transformer calls. A rollout $G(c, s)$ maps prompt c and seed s to a decoded video; fixing the seed fixes the initial noise, making the rollout deterministic (Appendix D). Best-of-N draws seeds $s_1, \ldots, s_N$, scores the full-compute candidates $S_i = V(G(c, s_i), c)$ with verifier V, and returns $i^\star = \arg\max_i S_i$ (Ma et al., 2025). We use ImageReward (Xu et al., 2023), averaged over 8 uniformly spaced frames. If one full rollout costs C_f, best-of-N costs NC_f; verification is shared and negligible, so we omit it.

Now let a cheaper rollout $\tilde{G}(c, s)$ cost $C_c = \gamma C_f$, with $\gamma < 1$. Search scores $\hat{S}_i = V(\tilde{G}(c, s_i), c)$ and selects $\hat{\imath} = \arg\max_i \hat{S}_i$. Its usefulness depends only on how well $(\hat{S}_i)$ preserves the within-prompt ordering of (S_i). We measure this with four quantities: per-prompt Spearman correlation ρ; top-1 agreement, $\Pr[\hat{\imath} = i^\star]$; regret, $S_{i^\star} - S_{\hat{\imath}}$; and *gain capture*,

$$\frac{\mathbb{E}[S_{\hat{\imath}}] - \mathbb{E}[S_1]}{\mathbb{E}[S_{i^\star}] - \mathbb{E}[S_1]},$$

the fraction of best-of-N's gain over one sample preserved by cheap selection. Regret and capture use the delivered full-compute value. Theorem 1 predicts this ratio of mean gains; tables that average per-prompt ratios are slightly lower (90.1% versus 94.0% for our default).

This criterion sharply separates three standard ways to reduce candidate cost (Table 1). Truncating to T=25 steps and caching at τ=0.10 (Section 3.1) both save 2×, yet caching retains 90.1% of the search gain per prompt while truncation retains only 72.6%, incurs six times the regret, and has worst-decile rank correlation 0.02, effectively chance. Reduced resolution (352×608) is cheapest, but selecting from these drafts performs no better than random: its −0.8% capture is consistent with zero. Truncation re-discretizes the trajectory, while changing resolution makes the same seed index a different sample; both therefore rank videos different from the 50-step, full-resolution samples eventually delivered. Caching instead preserves the seed and 50-step trajectory, perturbing the state only locally. Pixels may change visibly, yet the verifier ordering survives (Figure 2). Pixel fidelity and ranking fidelity are distinct, and search requires only the latter.

3 Method

Test-time search uses one rollout for two jobs. Exploration generates candidates for the verifier to rank; delivery generates the video returned to the user. Existing methods pay full compute for both,

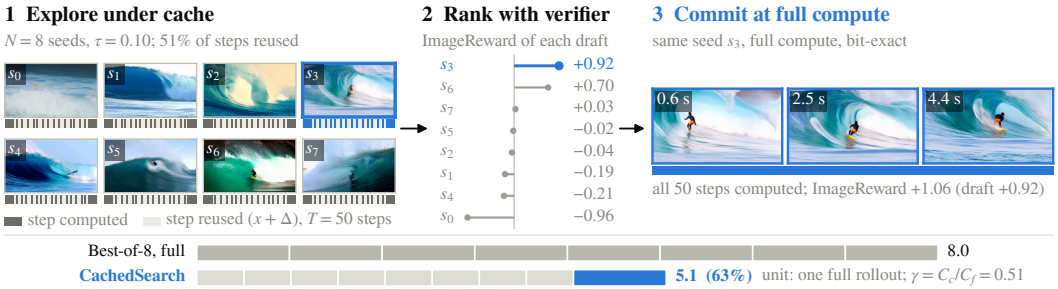


Figure 3: CachedSearch on one prompt (Wan2.1-T2V-1.3B, "A surfer riding inside a large barrel wave", N=8, τ=0.10). (1) Every seed is rolled out under caching; strips show each draft's measured computed and reused steps (positions schematic). (2) ImageReward ranks the drafts. (3) The chosen seed s_3, here also the full-compute winner, is regenerated at full compute. Bottom: total cost against full best-of-8.

although exploration needs only the ranking (Section 2). CachedSearch separates them: exploration uses an approximation chosen for ranking fidelity, while delivery regenerates only the selected seed at full compute (Figure 3).

3.1 Exploration with feature caching

We explore with training-free feature caching using the transformation-vector form of EasyCache (Zhou et al., 2025) (Algorithm 1, lines 2 to 12). A wrapper intercepts each transformer call. After computing at input $\boldsymbol{x}_{\text{ref}}$, it stores $\Delta = v_\theta(\boldsymbol{x}_{\text{ref}}) - \boldsymbol{x}_{\text{ref}}$; a skipped call at $\boldsymbol{x}$ returns $\boldsymbol{x} + \Delta$, assuming the residual $v_\theta(\boldsymbol{x}) - \boldsymbol{x}$ changes slowly across adjacent steps. An accumulated drift indicator a controls reuse (lines 4 to 10): each step adds the relative change from the last computed input, the call is skipped while $a \le \tau$, and a computed call refreshes $(\Delta, \boldsymbol{x}_{\text{ref}})$ and resets a. The first and last five of the T=50 steps are always computed, each guidance branch keeps separate state $(\Delta, \boldsymbol{x}_{\text{ref}}, a)$, and the cache resets for every candidate. Thus the cached rollout $G_\tau(c, s)$ remains deterministic in the seed (Appendix C).

Algorithm 1 CachedSearch: cached exploration, full-compute commit

Require: prompt c; seeds $s_{1:N}$; threshold τ; verifier V; mode $\in$ {commit, keep}
Ensure: one video; under commit, the exact full-compute sample of the selected seed

1: **for** $i = 1, \dots, N$ **do** ▷ explore: NC_c
2: $\quad \boldsymbol{x} \leftarrow \text{noise}(s_i)$; $\boldsymbol{x}_{\text{ref}} \leftarrow \boldsymbol{x}$; $a \leftarrow 0$ ▷ fresh cache
3: $\quad$ **for** $t = 1, \dots, T$ **do** ▷ state per CFG branch
4: $\quad\quad a \leftarrow a + \|\boldsymbol{x} - \boldsymbol{x}_{\text{ref}}\|_F / \|\boldsymbol{x}_{\text{ref}}\|_F$
5: $\quad\quad$ **if** $5 < t \le T-5$ **and** $a \le \tau$ **then**
6: $\quad\quad\quad \hat{v} \leftarrow \boldsymbol{x} + \Delta$ ▷ skip the call
7: $\quad\quad$ **else**
8: $\quad\quad\quad \hat{v} \leftarrow v_\theta(\boldsymbol{x}, t, c)$ ▷ compute
9: $\quad\quad\quad \Delta \leftarrow \hat{v} - \boldsymbol{x}$; $\boldsymbol{x}_{\text{ref}} \leftarrow \boldsymbol{x}$; $a \leftarrow 0$
10: $\quad\quad$ **end if**
11: $\quad\quad \boldsymbol{x} \leftarrow$ ODE step along the guided $\hat{v}$
12: $\quad$ **end for**
13: $\quad y_i \leftarrow \text{decode}(\boldsymbol{x})$; $\hat{S}_i \leftarrow V(y_i, c)$ ▷ score
14: **end for**
15: $\hat{\imath} \leftarrow \arg\max_i \hat{S}_i$ ▷ uses the ranking only
16: **if** mode = commit **then**
17: $\quad$ **return** $G(c, s_{\hat{\imath}})$ ▷ full compute: C_f
18: **else**
19: $\quad$ **return** $y_{\hat{\imath}}$ ▷ keep: cached video
20: **end if**

At τ=0.10 on Wan2.1-T2V-1.3B, the wrapper reuses 51% of steps and reduces rollout time from C_f=68.3 s to C_c=34.7 s (1.97×). Skipped calls perturb the trajectory locally, so ranking errors concentrate among candidates with small score gaps (Section 5). We therefore tune τ for ranking rather than visual fidelity: for each model, we choose the most aggressive threshold retaining at least 85% of the search gain (Section 4.4), even when cached videos visibly differ from full compute (Figure 2).

3.2 Algorithm and cost

Because G is deterministic in the seed, the commit $G(c, s_{\hat{\imath}})$ (Algorithm 1, line 17) is the exact full-compute sample of that seed, with no latents stored across candidates. If cached and full rankings choose the same winner, CachedSearch returns exactly the video from full best-of-N; otherwise it returns the full-compute sample of the seed preferred by the cached ranking. Approximation therefore affects only seed selection, not the delivered sample for a chosen seed, so its cost appears entirely

as selection regret. The cheaper KEEP mode returns the cached winner directly; we evaluate it but recommend COMMIT, since cached videos may contain motion artifacts missed by a frame-level verifier (Section 5). Omitting shared verification cost,

$$\text{cost}(\text{KEEP}) = NC_c, \qquad \text{cost}(\text{COMMIT}) = NC_c + C_f, \qquad \text{cost}(\text{best-of-}N) = NC_f, \tag{1}$$

so COMMIT becomes cheaper than full search for $N > N^* = 1/(1-\gamma)$, where $\gamma = C_c/C_f$; both cost ratios approach γ as N grows. On Wan2.1-T2V-1.3B, $\gamma = 0.508$ and $N^* = 2.03$. Caching adds two latent-sized tensors and one scalar per guidance branch, so peak memory remains comparable to a plain rollout.

Adding CACHEDSEARCH to an existing best-of-N pipeline requires three steps, none of which changes the weights or sampler. First, wrap the transformer with any training-free cache; because only rollout changes, the method applies to any sample-and-rank search, including pruned variants (Zhao et al., 2026), with any cache and verifier. Second, calibrate τ once per model and verifier on a pilot set of 50 prompts and 8 seeds, choosing the most aggressive threshold that retains at least 85% of the gain; the same pilot estimates γ and rank correlation. Third, allocate the budget: N_f full rollouts buy $\lfloor (N_f - 1)/\gamma \rfloor$ cached candidates plus one commit, and Corollary 1 gives the condition under which this wider search wins.

3.3 GAIN CAPTURE UNDER RANKING NOISE

Fix a prompt, and model each candidate i by its full score S_i and cached score $\hat{S}_i$ (cache threshold τ). Assume (i) the pairs $(S_i, \hat{S}_i)$ are i.i.d. over the N seeds, (ii) $S_i \sim \mathcal{N}(\mu, \sigma^2)$ and the pair has a Gaussian copula with latent correlation $r \in [0, 1]$ (only the ranks of $\hat{S}$ enter selection, so its marginal is irrelevant; r is calibrated from the per-prompt Spearman ρ by $r = 2\sin(\pi\rho/6)$, so the median $\rho = 0.905$ gives $r = 0.913$), and (iii) the verifier score is the value of a candidate. Let $e_N = \mathbb{E}[\max_{i \le N} Z_i]$ for i.i.d. standard normals ($e_2 = 0.56$, $e_4 = 1.03$, $e_8 = 1.42$, $e_{16} = 1.77$).

Theorem 1 (Selection under rank noise). *Under (i)–(iii), CACHEDSEARCH (commit) with $\hat{\imath} = \arg\max_i \hat{S}_i$ delivers $\mathbb{E}[S_{\hat{\imath}}] = \mu + \sigma r e_N$, while full-compute best-of-N delivers $\mathbb{E}[\max_i S_i] = \mu + \sigma e_N$. Hence the expected regret is $\mathcal{R}(N, r) = \sigma(1 - r)\, e_N$, and the gain capture $(\mathbb{E}[S_{\hat{\imath}}] - \mu)/(\mathbb{E}[\max_i S_i] - \mu)$ equals r for every $N \ge 2$.*

Proof sketch. Standardize and write $Z_i = r\hat{Z}_i + \sqrt{1 - r^2}\,\eta_i$ with η_i independent of every $\hat{Z}_j$. The index $\hat{\imath}$ is a function of $(\hat{Z}_j)_{j \le N}$ alone, so $\mathbb{E}[\eta_{\hat{\imath}}] = 0$ and $\mathbb{E}[Z_{\hat{\imath}}] = r\,\mathbb{E}[\max_j \hat{Z}_j] = r\, e_N$. Appendix A gives the full proof, the top-1 agreement law, the prompt-heterogeneous version (capture is the spread-weighted mean of r), and the bounded-score extension. □

Corollary 1 (Break-even and iso-cost widening). *Let $\gamma = C_c/C_f$ be the cached-to-full cost ratio. Commit is cheaper than full best-of-N iff $N > 1/(1-\gamma)$. A budget that buys N_f full candidates buys $N_c = \lfloor (N_f - 1)/\gamma \rfloor$ cached candidates plus the commit, and under (i)–(iii) commit attains the higher expected value at that budget iff $r\, e_{N_c} > e_{N_f}$.*

Ranking noise removes the constant fraction $1 - r$ of the search gain and does not compound with width, so widening with cheap candidates can more than repay it. At the measured $\gamma = 0.508$ and $r = 0.913$, N_f=3 buys only N_c=3 and full search wins, while N_f=4 buys N_c=5 and cached search wins ($re_5 = 1.06 > e_4 = 1.03$); the advantage peaks near 7% at N_f=6 to 8 and then shrinks, because $e_{N_c}/e_{N_f} \to 1$ (Appendix A). For bounded scores capture tends to one as N grows (Proposition 4), so the Gaussian case is conservative in the wide-search limit. Section 4.3 tests both predictions.

4 EXPERIMENTS

4.1 SETUP

We generate every candidate twice from the same seed, once cached and once at full compute, so each comparison uses identical noise. The *gate grid* (50 prompts, 8 seeds, Wan2.1-T2V-1.3B at 480×832, 81 frames, T=50, guidance 5.0, one GH200 per rollout) is our configuration set: we fixed τ=0.10, COMMIT, and N=8 on it before evaluating the official VBench (7,552 pairs) and harder VBench-2.0

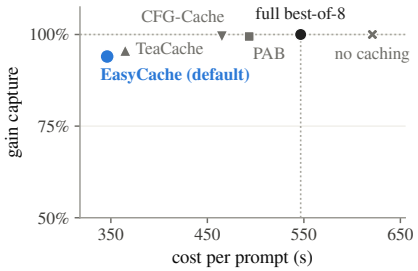


Figure 4: Four published caches as exploration engines for best-of-8 commit on Wan2.1-T2V-1.3B: gain capture against cost per prompt.

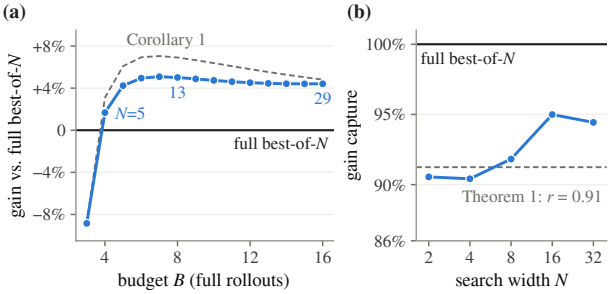


Figure 5: (a) Gain of CachedSearch over full best-of-B at a budget of B full rollouts, with $N{=}\lfloor(B{-}1)/\gamma\rfloor$ cached candidates (labels); dashed: Corollary 1. (b) Gain capture against width; dashed: Theorem 1. Gate grid, all seed subsets.

Table 2: Cached ($\tau{=}0.10$) versus full-compute rankings over 8 seeds on Wan2.1-T2V-1.3B, on the gate grid and two full prompt suites; capture is the mean per-prompt fraction of best-of-8 gain kept by cached selection.

suite	prompts	pairs	median $\rho\uparrow$	p10 $\rho\uparrow$	top-1 $\uparrow$	capture $\uparrow$	speedup $\uparrow$
gate grid	50	400	0.905	0.61	64%	90.1%	1.97×
VBench	944	7,552	0.905	0.69	72%	90.2%	1.95×
VBench-2.0	1,013	8,104	0.881	0.69	69%	89.3%	1.98×

(8,104 pairs) (Huang et al., 2024; Zheng et al., 2025). Unless noted, we evaluate strategies from measured scores and latencies over all $\binom{8}{N}$ seed subsets, without extrapolation, and measure gain relative to the mean full-compute score of one sample. Brackets denote prompt-bootstrap intervals. A no-cache control reproduces full-compute videos bit-exactly (Appendix D).

4.2 Ranking fidelity at scale

Caching halves candidate cost while largely preserving rank (Table 2). On VBench, median ρ is 0.905, the cached ranking selects the true winner on 72% of prompts, and mean regret is 0.056 versus 0.79 for random selection, retaining 90.2% of the gain per prompt. On VBench-2.0, median ρ is 0.881 with 89.3% capture. Both lie within one point of the gate-grid capture (90.1%), where all settings were fixed, and only 11% of VBench prompts fall below $\rho{=}0.7$ (Appendix Figure 20). Errors concentrate on nearly tied candidates, giving zero median regret (Section 5).

4.3 Cost and gain

CachedSearch retains 94.0% of best-of-8's gain for 346 s per prompt instead of 547 s, or 63% of the cost (Table 3). Commit breaks even at $N{=}2$, consistent with $N^*{=}2.03$, and saves more as search widens. The same saving can instead buy wider search. For 277 s, only 1.2% more than full best-of-4 (1.9% on VBench), CachedSearch searches six candidates and yields 6.6% more gain on the gate grid ($[0.7, 12.0]\%$) and 11.1% more on VBench ($[9.6, 12.4]\%$), close to the 12% predicted by Theorem 1 for this pairing (re_6/e_4). At exactly equal cost, mixing five- and six-candidate searches still gives +5.8% on the gate grid and +9.8% on VBench. At equal gain, the saving is 21%: on VBench, CachedSearch with $N{=}8$ reaches in 344 s the gain requiring 435 s from full search, a mixture of best-of-6 and best-of-7 (Figure 1b).

The advantage holds at every measured budget above three full rollouts (Figure 5a). A budget of B full rollouts buys either full best-of-B or, by Corollary 1, $\lfloor(B{-}1)/\gamma\rfloor$ cached candidates plus the commit. Over all seed subsets of the gate grid's 32-seed pool, full search wins only at $B{=}3$, where the budget fits three cached candidates (−8.8%). CachedSearch wins by 1.7% at $B{=}4$ and by 4.2–5.1% for every B from 5 to 16 (+5.0% at $B{=}8$). The measured curve matches the Corollary's shape, crossing zero between $B{=}3$ and 4, peaking at $B{=}7$, and remaining within 2.0 points of it.

Capture also remains stable as search widens. From median Spearman alone, with no fitting, Theorem 1 predicts capture $r{=}0.913$ for every N. Over all seed subsets of the gate grid's 16-seed pool (32 seeds for $N{=}32$), measured capture is 90.6%, 90.4%, and 91.8% at $N{=}2$, 4, and 8, then rises to 95.0% and 94.4% at $N{=}16$ and 32 (Figure 5b; paired rise from $N{=}4$ to 16: +4.6 points, $[0.8, 8.0]$).

Table 3: Cost and value per strategy (gate grid, all seed subsets). Gain is over the mean single sample (68 s); capture is the ratio of mean gains; rel. is cost relative to full search at the same N. KEEP delivers cached videos, so its gain is nominal (Section 5).

	full best-of-N		KEEP (cached delivery)			CACHEDSEARCH COMMIT			
N	gain ↑	cost (s) ↓	gain ↑	capture ↑	cost (s) ↓	gain ↑	capture ↑	cost (s) ↓	rel. ↓
2	+0.312	137	+0.294	94.3%	69	+0.281	90.1%	138	100.8%
4	+0.522	273	+0.504	96.6%	139	+0.469	89.8%	207	75.8%
8	+0.657	547	+0.655	99.6%	278	**+0.618**	**94.0%**	**346**	**63.3%**

Matched cost: full best-of-4 gains +0.522 for 273 s; COMMIT with N=6 gains **+0.556** for 277 s.

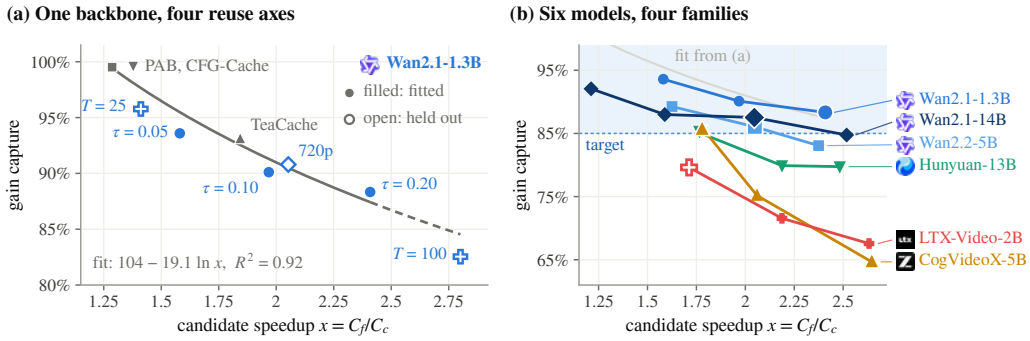


Figure 6: Gain capture (mean per-prompt ratio) against candidate speedup. (a) Wan2.1-T2V-1.3B: the log fit uses only the filled points (threshold sweep in blue; PAB, CFG-Cache and TeaCache in grey); open points are held out (T=25 and T=100 schedules, 720p). (b) Threshold sweeps on six models (τ rises to the right); large markers are the calibrated τ^* (at least 85% capture, shaded), open for LTX-Video, which has none.

The Gaussian model is therefore conservative at large widths, as Proposition 4 predicts for bounded scores, though it does not predict the size of the excess. Table 3's 94.0% at N=8 lies within the pooled interval $[88.8, 94.4]$%.

4.4 OTHER CACHES AND MODELS

Any training-free cache can serve as the exploration engine. We evaluate four published caches under the same protocol (Figure 4; Appendix D.5). All run best-of-8 for less than full search (346–493 s versus 547 s) while retaining at least 94% of its gain. PAB (Zhao et al., 2024) and CFG-Cache (Lv et al., 2024) reuse parts of each transformer call, giving 1.29× and 1.38× cheaper candidates while retaining 99.5% of the gain as a ratio of mean gains. TeaCache (Liu et al., 2024) and EasyCache skip entire calls, reach 1.84× and 1.97×, and retain 95.6% and 94.0%. The engines lie on one capture-cost frontier (Figure 6a), so the cache determines the operating point.

One threshold per model carries the method across architectures: five of six models from four families (Wan Team, 2025a;b; Yang et al., 2024; Kong et al., 2024; HaCohen et al., 2024), spanning 1.3B to 14B parameters, retain at least 85% of the gain with 1.8–2.4× cheaper candidates (Table 4, Figures 1c and 6b). We choose τ^* as the most aggressive measured τ retaining at least 85% mean per-prompt capture on 50 prompts: 0.20 for Wan2.1-1.3B, 0.10 for the other two Wan models, and 0.05 for CogVideoX-5B and HunyuanVideo-13B. Ranking fidelity depends more on architecture than size. At shared τ=0.10, Wan2.1-1.3B and Wan2.1-14B have the same median ρ (0.905), whereas CogVideoX-5B and HunyuanVideo-13B reach 0.762, lose capture at similar speedups, and require smaller thresholds. Absolute savings grow with model cost, from 6 s per candidate on LTX-Video-2B to 174 s on Wan2.1-14B at τ=0.10. LTX-Video marks the boundary: no measured threshold reaches 85% capture.

4.5 QUALITATIVE RESULTS

When cached and full rankings agree, as on 72% of VBench prompts, CACHEDSEARCH returns the full-search video bit-for-bit and captures the entire best-of-8 gain over one sample (Figure 7). The truck stays on the road rather than dissolving into streaks; the panda sits on the requested swing; the robot dances on a crowded street rather than against an empty grey background; and the dog is

Table 4: Six models at the shared threshold τ=0.10 and at the calibrated τ^*, the most aggressive measured threshold with at least 85% mean per-prompt capture (50 prompts, 8 seeds; sweeps in Appendix Tables 15 and 16). Skip: fraction of reused steps. LTX-Video has no τ^* (its best setting, τ=0.02, is shown).

	shared τ=0.10			calibrated τ^*				
model	med ρ ↑	capture ↑	speedup ↑	τ^*	skip	speedup ↑	capture ↑	med ρ ↑
Wan2.1-1.3B	0.905	90.1%	1.97×	0.20	60%	2.41×	88.3%	0.857
Wan2.2-TI2V-5B	0.881	86.0%	2.05×	0.10	55%	2.05×	86.0%	0.881
Wan2.1-14B	0.905	87.5%	2.05×	0.10	52%	2.05×	87.5%	0.905
CogVideoX-5B	0.762	75.2%	2.06×	0.05	44%	1.78×	85.9%	0.810
HunyuanVideo-13B	0.762	79.9%	2.19×	0.05	46%	1.77×	85.1%	0.810
LTX-Video-2B	0.536	67.6%	2.63×	none	43%	1.71×	79.6%	0.786



Figure 7: Delivered videos on four VBench prompts (Wan2.1-T2V-1.3B, τ=0.10) for one sample (seed 0), full best-of-8 and CachedSearch; headers give the mean per-prompt cost (Table 3). Each cell shows frames 20 and 60 of 81; chips give the full-compute ImageReward and the seed.

recognizably a corgi at sunset. Figure 1a extends this behavior to other backbones, up to LTX-2 (19B; HaCohen et al., 2026), with thresholds calibrated per backbone on new prompts. These are selected examples; Section 5 examines cases where the cached ranking fails.

5 Analysis

The cached ranking errs where a wrong pick costs least (Figure 8). On VBench, a prompt's rank correlation rises with the spread of its eight full-compute scores (Spearman +0.31, p=2×10^{-22}). Prompts with $\rho < 0.7$ have a median spread of 0.40 against 0.56 for the rest, and the lower half of the spread distribution holds 78.5% of them but only 56.1% of the regret. This is why a median ρ of 0.905 still yields 90% per-prompt capture and zero median regret, and why an approximation inside search should be judged by its regret, not its rank correlation alone. Spread also explains why the verifier matters. Under the video-native VideoScore (He et al., 2024) on 467 VBench prompts, cached rankings agree less with full ones (median ρ=0.762, top-1 agreement 54%, capture 65.6%), because VideoScore separates the candidates of a prompt by a median of only 0.083 on its $[1, 4]$ scale, against 0.55 for ImageReward. The two verifiers disagree with each other far more than caching perturbs either (within-prompt rank correlation +0.24), so the threshold should be calibrated for the verifier in use (Appendix G.1).

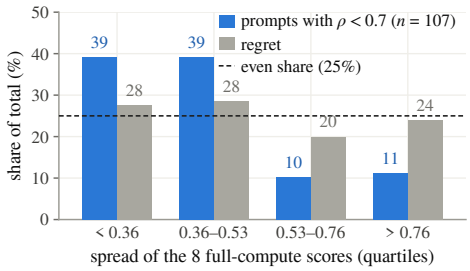


Figure 8: VBench prompts (Wan2.1-T2V-1.3B, τ=0.10) grouped by spread quartile. Bars: each quartile's share of the prompts with $\rho < 0.7$ (blue) and of the total regret (grey).

Committing at full compute costs 68 s per prompt and guarantees that every delivered video is an unmodified full-compute sample. Delivering the cached winner itself (KEEP) nominally retains 99.6% of the gain at 51% of the cost (Table 3), but that number is optimistic twice over. The verifier that

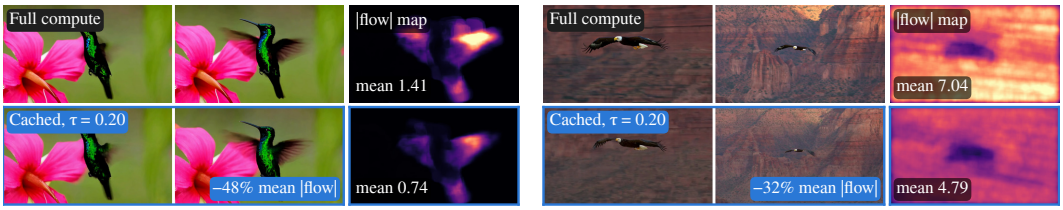


Figure 9: Same-seed winner pairs at $\tau = 0.20$, full compute above and cached below (blue outline): two frames and the temporal-mean Farneback optical-flow magnitude of each video, color scale shared within a pair. The pairs are the two largest drops in mean flow among the 25 of 50 pairs with above-median flow; Figure 12 samples the other ranks.

selected the cached video also scores it, so selection noise inflates its score: the selected cached video even outscores its own full-compute twin (Appendix G). And a frame-level verifier cannot see motion, which caching does dampen: mean optical flow of the cached videos falls by 3.1% at τ=0.10 and by 8.0% at τ=0.20 (Appendix F.1; Figure 9 shows the two largest drops).

CachedSearch also multiplies with pruning, which cuts how many candidates run to completion while caching cuts what each costs. We ran all 8 candidates cached to step 20 of 50, scored a preview decoded from the sampler's clean-sample estimate, and continued the top 4. Exploration falls to 175 s, 3.12× below full best-of-8 and 1.59× below cached exploration alone. End to end, including preview scoring and the commit, a prompt costs 256 s instead of 346 s, while per-prompt capture moves only from 90.1% to 88.6% (94.0% to 93.3% as a ratio of means) and the true best candidate survives the prune on 86% of prompts.

6 Related work

Ma et al. (2025) cast inference-time scaling for diffusion as verifier-guided search over sampling noise. Video-T1 (Liu et al., 2025b) searches over frames with multimodal rewards, EvoSearch (He et al., 2025) evolves denoising trajectories, and latent beam search (Oshima et al., 2025) branches at every step. LatSearch (Zhao et al., 2026) and early-failure detection (Kwon et al., 2026) prune candidates before they finish, and GEARS (He et al., 2026) recycles rejected video candidates. These methods decide *which* candidates to generate; each still generates them at full compute, the cost CachedSearch cuts, so it composes with them (Section 5 shows this for a preview-based prune). Training-free feature caches reuse computation across adjacent denoising steps (Zhao et al., 2024; Lv et al., 2024; Liu et al., 2024; Zhou et al., 2025; Cui et al., 2025; Liu et al., 2025a; Gao et al., 2025; Aliev et al., 2026) and are evaluated by the fidelity of one video to its full-compute reference. We use them unchanged but judge them by the preservation of a verifier's ranking across candidates, under which pixel fidelity and search value can diverge (Table 1). Speculative sampling (De Bortoli et al., 2025; Hu & Zhang, 2026; Bajpai et al., 2026) and model stitching (Pan et al., 2024; Sun et al., 2026) accelerate a *single* trajectory by verifying drafts within it; CachedSearch drafts whole candidates and verifies across them with the search's own verifier. Concurrent work on image generation, Flash-BoN (Rawal et al., 2026), also ranks cheap drafts for best-of-N; we add the property a draft must have and its test on over 15,000 paired video rollouts, the contrast between trajectory-preserving caching and step truncation, exact full-compute delivery, and a theory that predicts the retained gain.

7 Conclusion

Search needs only the verifier's ranking of its candidates, and training-free caching preserves it: median rank correlation 0.905 on VBench and 0.881 on VBench-2.0, where step truncation at the same saving loses it, and a Gaussian-copula argument predicts the retained gain at every width. CachedSearch explores under caching and commits the winner at full compute. It returns exact full-compute videos, keeps 94% of best-of-8's gain at 63% of its cost, beats full search at near-equal cost, and plugs into any best-of-N pipeline with one threshold calibrated on 50 prompts. Approximations inside search should be judged by the rankings they preserve, not the pixels they reproduce.

Reproducibility statement

All experiments use publicly released checkpoints: Wan2.1-T2V-1.3B (Wan Team, 2025a), Wan2.1-T2V-14B (Wan Team, 2025a), Wan2.2-TI2V-5B (Wan Team, 2025b;a), CogVideoX-5B (Yang et al., 2024), HunyuanVideo-13B (Kong et al., 2024), and LTX-Video-2B (HaCohen et al., 2024); the qualitative gallery also uses LTX-2 (HaCohen et al., 2026). We use ImageReward (Xu et al., 2023) and VideoScore (He et al., 2024) as verifiers with a training-free caching wrapper; no model is trained or fine-tuned. Appendix D gives exact configurations and measured costs. We will release the code.

Ethics statement

This work reduces the inference cost of an existing capability (text-to-video generation with test-time search) rather than introducing new generative capabilities; risks are those of the underlying open models. Cheaper high-quality sampling lowers the cost of both beneficial and malicious video synthesis. Provenance mitigations (watermarking, content credentials) are orthogonal to and unaffected by our method: caching changes how exploration candidates are computed, not what is delivered.

AI USE STATEMENT

Generative AI assistant (ChatGPT) was used during preparation of this submission for format conversion, manuscript editing, and code to visualize stored outputs. This assistance did not produce new model outputs or experimental measurements. The research and experiments preceded this submission- preparation use. The authors are responsible for the final content and interpretation of this work.

References

Mishan Aliev, Eva Neudachina, Ilya Bykov, Aleksandr Oganov, Kirill Struminsky, Aibek Alanov, and Denis Rakitin. Recache: Learning budget-aware caching schedules for diffusion models via reinforce. *arXiv preprint arXiv:2606.06060*, 2026.

Divya Jyoti Bajpai, Shubham Agarwal, Apoorv Saxena, Kuldeep Kulkarni, Subrata Mitra, and Manjesh Kumar Hanawal. Flowcast: Trajectory forecasting for scalable zero-cost speculative flow matching. *arXiv preprint arXiv:2602.01329*, 2026. ICLR 2026.

Hanshuai Cui, Zhiqing Tang, Zhifei Xu, Zhi Yao, Wenyi Zeng, and Weijia Jia. Bwcache: Accelerating video diffusion transformers through block-wise caching. *arXiv preprint arXiv:2509.13789*, 2025.

Valentin De Bortoli, Alexandre Galashov, Arthur Gretton, and Arnaud Doucet. Accelerated diffusion models via speculative sampling. *arXiv preprint arXiv:2501.05370*, 2025.

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. *arXiv preprint arXiv:2511.00090*, 2025. NeurIPS 2025 Spotlight.

Youping Gu, Xiaolong Li, Yuhao Hu, Minqi Chen, and Bohan Zhuang. Blade: Block-sparse attention meets step distillation for efficient video generation. *arXiv preprint arXiv:2508.10774*, 2025.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, Victor Kulikov, Yaki Bitterman, Zeev Melumian, and Ofir Bibi. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, Eitan Richardson, Guy Shiran, Itay Chachy, Jonathan Chetboun, Michael Finkelson, Michael Kupchick, Nir Zabari, Nitzan Guetta, Noa Kotler, Ofir Bibi, Ori Gordon, Poriya Panet, Roi Benita, Shahar Armon, Victor Kulikov, Yaron Inger, Yonatan Shiftan, Zeev Melumian, and Zeev Farbman. LTX-2: Efficient joint audio-visual foundation model. *arXiv preprint arXiv:2601.03233*, 2026.

Hangzhou He, Lunhao Duan, Shanshan Zhao, Kaiwen Li, Qing-Guo Chen, Weihua Luo, and Yanye Lu. Test-time scaling for video diffusion models via diagnosis-guided candidate recycling. *arXiv preprint arXiv:2608.29322*, 2026.

Haoran He, Jiajun Liang, Xintao Wang, Pengfei Wan, Di Zhang, Kun Gai, and Ling Pan. Scaling image and video generation via test-time evolutionary search. *arXiv preprint arXiv:2505.17618*, 2025.

Xuan He, Dongfu Jiang, Ge Zhang, Max Ku, Achint Soni, Sherman Siu, Haonan Chen, Abhranil Chandra, Ziyan Jiang, Aaran Arulraj, Kai Wang, Quy Duc Do, Yuansheng Ni, Bohan Lyu, Yaswanth Narsupalli, Rongqi Fan, Zhiheng Lyu, Yuchen Lin, and Wenhu Chen. Videoscore: Building automatic metrics to simulate fine-grained human feedback for video generation. *arXiv preprint arXiv:2406.15252*, 2024.

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

Yuezhou Hu and Jintao Zhang. Speculative decoding for autoregressive video generation. *arXiv preprint arXiv:2604.17397*, 2026.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Vbench: Comprehensive benchmark suite for video generative models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. arXiv:2311.17982.

Yicheng Ji, Zhizhou Zhong, Jun Zhang, Qin Yang, XiTai Jin, Ying Qin, Wenhan Luo, Shuiyang Mao, Wei Liu, and Huan Li. Forcing-kv: Hybrid kv cache compression for efficient autoregressive video diffusion models. *arXiv preprint arXiv:2605.09681*, 2026.

Sejoon Jun, Zheng Ding, Huangyuan Su, Weirui Ye, and Yilun Du. Temporal backtracking search for test-time generative video reasoning. *arXiv preprint arXiv:2606.13861*, 2026.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

William H. Kruskal. Ordinal measures of association. *Journal of the American Statistical Association*, 53(284):814–861, 1958.

Byung-Ki Kwon, Sohwi Lim, Hyeon-Woo Nam, Ye-Bin Moon, and Tae-Hyun Oh. Early failure detection and intervention in video diffusion models. *arXiv preprint arXiv:2603.14320*, 2026.

Yitong Li, Junsong Chen, Haopeng Li, Haozhe Liu, Jincheng Yu, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Sol video inference engine: Agent-native full-stack acceleration framework for efficient video generation. *arXiv preprint arXiv:2606.23743*, 2026.

Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *European Conference on Computer Vision (ECCV)*, 2024.

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

Dong Liu, Yanxuan Yu, Jiayi Zhang, Yifan Li, Ben Lengerich, and Ying Nian Wu. Fastcache: Fast caching for diffusion transformer through learnable linear approximation. *arXiv preprint arXiv:2505.20353*, 2025a.

Fangfu Liu, Hanyang Wang, Yimo Cai, Kaiyan Zhang, Xiaohang Zhan, and Yueqi Duan. Video-t1: Test-time scaling for video generation. *arXiv preprint arXiv:2503.18942*, 2025b.

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. *arXiv preprint arXiv:2411.19108*, 2024. CVPR 2025.

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

Zhengyao Lv, Chenyang Si, Junhao Song, Zhenyu Yang, Yu Qiao, Ziwei Liu, and Kwan-Yee K. Wong. Fastercache: Training-free video diffusion model acceleration with high quality. *arXiv preprint arXiv:2410.19355*, 2024.

Zheqi Lv, Zhibo Zhu, Jinke Wang, Qi Tian, Shengyu Zhang, Zhengyu Chen, Chengxi Zang, Zhou Zhao, and Fei Wu. Navicache: Test-time self-calibration caching for video generation. *arXiv preprint arXiv:2606.26795*, 2026. ICML 2026.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. Inference-time scaling for diffusion models beyond scaling denoising steps. *arXiv preprint arXiv:2501.09732*, 2025.

Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024.

Yuta Oshima, Masahiro Suzuki, Yutaka Matsuo, and Hiroki Furuta. Inference-time text-to-video alignment with diffusion latent beam search. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Zizheng Pan, Bohan Zhuang, De-An Huang, Weili Nie, Zhiding Yu, Chaowei Xiao, Jianfei Cai, and Anima Anandkumar. T-stitch: Accelerating sampling in pre-trained diffusion models with trajectory stitching. *arXiv preprint arXiv:2402.14167*, 2024.

Ruchit Rawal, Reza Shirkavand, Sayak Paul, Yuxin Wen, Heng Huang, Yizheng Chen, Tom Goldstein, and Gowthami Somepalli. Flash-BoN: Instant drafts for inference-time scaling in diffusion models. *arXiv preprint arXiv:2607.04461*, 2026.

Shreshth Saini, Shashank Gupta, and Alan C. Bovik. Rectified-cfg++ for flow based models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a.

Shreshth Saini, Ru-Ling Liao, Yan Ye, and Alan C. Bovik. LGDM: Latent guidance in diffusion models for perceptual evaluations. In *International Conference on Machine Learning (ICML)*, 2025b.

Desen Sun, Jason Hon, Jintao Zhang, and Sihang Liu. Hybridstitch: Pixel and timestep level model stitching for diffusion acceleration. *arXiv preprint arXiv:2603.07815*, 2026.

Yijing Tu, Shaojin Wu, Mengqi Huang, Wenchuan Wang, Yuxin Wang, Chunxiao Liu, and Zhendong Mao. Stream-t1: Test-time scaling for streaming video generation. *arXiv preprint arXiv:2605.04461*, 2026.

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

Wan Team. Wan2.2: Open and advanced large-scale video generative models. https://github.com/Wan-Video/Wan2.2, 2025b. Open-weights release including Wan2.2-TI2V-5B; technical report shared with Wan2.1 (arXiv:2503.20314).

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992. doi: 10.1007/BF00992696.

Junyi Wu, Zhiteng Li, Zheng Hui, Yulun Zhang, Linghe Kong, and Xiaokang Yang. Quantcache: Adaptive importance-guided quantization with hierarchical latent and layer caching for video generation. *arXiv preprint arXiv:2503.06545*, 2025.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2304.05977.

Jiazheng Xu, Yu Huang, Jiale Cheng, Yuanming Yang, Jiajun Xu, Yuan Wang, Wenbo Duan, Shen Yang, Qunlin Jin, Shurun Li, Jiayan Teng, Zhuoyi Yang, Wendi Zheng, Xiao Liu, Dan Zhang, Ming Ding, Xiaohan Zhang, Xiaotao Gu, Shiyu Huang, Minlie Huang, Jie Tang, and Yuxiao Dong. Visionreward: Fine-grained multi-dimensional human preference learning for image and video generation. *arXiv preprint arXiv:2412.21059*, 2024.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

Jintao Zhang, Kaiwen Zheng, Kai Jiang, Haoxu Wang, Ion Stoica, Joseph E. Gonzalez, Jianfei Chen, and Jun Zhu. Turbodiffusion: Accelerating video diffusion models by 100-200 times. *arXiv preprint arXiv:2512.16093*, 2025.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018. arXiv:1801.03924.

Xuanlei Zhao, Xiaolong Jin, Kai Wang, and Yang You. Real-time video generation with pyramid attention broadcast. *arXiv preprint arXiv:2408.12588*, 2024. ICLR 2025.

Zengqun Zhao, Ziquan Liu, Yu Cao, Shaogang Gong, Zhensong Zhang, Jifei Song, Jiankang Deng, and Ioannis Patras. Latsearch: Latent reward-guided search for faster inference-time scaling in video diffusion. *arXiv preprint arXiv:2603.14526*, 2026. ECCV 2026.

Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Lulu Gu, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, Yu Qiao, and Ziwei Liu. Vbench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*, 2025.

Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.

Xin Zhou, Dingkang Liang, Kaijin Chen, Tianrui Feng, Xiwu Chen, Hongkai Lin, Yikang Ding, Feiyang Tan, Hengshuang Zhao, and Xiang Bai. Less is enough: Training-free video diffusion acceleration via runtime-adaptive caching. *arXiv preprint arXiv:2507.02860*, 2025.

Appendix

A Ranking-noise model for cached exploration . . . 14
B Theory against measurement . . . 18
C Method details . . . 18
D Experimental details and reproducibility . . . 19
E Qualitative examples . . . 21
F Additional results and ablations . . . 24
G Verifier choice and verifier bias . . . 27
H Models, scale, and threshold calibration . . . 29
I Extended related work . . . 32
J Limitations . . . 33

A Ranking-noise model for cached exploration

Section 3.3 states the rank-noise model, Theorem 1, and Corollary 1. Here we give their full forms and proofs (Propositions 1 and 5), derive top-1 agreement, extend the analysis to heterogeneous prompts and bounded scores, and test each prediction against measurement (Section 4).

A.1 Setup and assumptions

We use the notation of Section 3. For prompt c and width N, candidate i has full score $S_i = V(G(c, s_i), c)$ and cached score $\hat{S}_i = V(G_\tau(c, s_i), c)$, both deterministic given seed s_i, so randomness is only over i.i.d. seeds. CachedSearch selects $\hat{\imath} = \arg\max_i \hat{S}_i$; committing the selected seed delivers $S_{\hat{\imath}}$ by seed determinism (Section 3.2), while the full-compute oracle attains $\max_i S_i$. The assumptions of Section 3.3 are:

Assumption 1 (Exchangeable candidates)**.** *The pairs $(S_i, \hat{S}_i)$, $i = 1, \ldots, N$, are i.i.d. across candidates, reflecting independently drawn seeds that enter symmetrically.*

Assumption 2 (Gaussian-copula ranking noise)**.** *Within a prompt, $(S_i, \hat{S}_i)$ is bivariate normal with $S_i \sim \mathcal{N}(\mu, \sigma^2)$ and correlation $r \in [0, 1]$. Because selection uses only the rank of $\hat{S}$ and $\arg\max$ is invariant to strictly increasing transforms, its marginal is irrelevant. The assumption therefore has two parts: (i) a Gaussian dependence structure (copula) with parameter r between full and cached scores, and (ii) a Gaussian marginal for the full score. Part (ii) converts rank statements into reward units.*

Assumption 3 (Verifier as value)**.** *The full score S is treated as candidate value. Verifier misspecification (reward hacking) is an orthogonal failure mode of all verifier-guided search, examined empirically through verifier ensembles in Section 4; the analysis isolates only the additional error introduced by caching.*

Under Assumption 2, write $S_i = \mu + \sigma Z_i$ and standardize $\hat{S}_i$ as $\hat{Z}_i$. Each $(Z_i, \hat{Z}_i)$ is standard bivariate normal with correlation r, and $\hat{\imath} = \arg\max_i \hat{Z}_i$. Define

$$e_N = \mathbb{E}\left[\max_{i \le N} Z_i\right] = \int_{-\infty}^{\infty} z\, N\phi(z)\Phi(z)^{N-1}\, dz \tag{2}$$

as the expected maximum of N standard normals ($e_2 = 1/\sqrt{\pi} \approx 0.564$; $e_3 \approx 0.846$; $e_4 \approx 1.029$; $e_6 \approx 1.267$; $e_8 \approx 1.424$; $e_{16} \approx 1.766$).

A.2 Gain capture and regret under rank noise

Proposition 1 (Selection under rank noise)**.** *Under Assumptions 1–2,*

$$\mathbb{E}\big[S_{\hat{\imath}}\big] = \mu + \sigma r e_N, \qquad \mathbb{E}\big[\max_i S_i\big] = \mu + \sigma e_N. \tag{3}$$

Hence the expected regret relative to full-compute best-of-N is

$$\mathcal{R}(N, r) = \mathbb{E}\big[\max_i S_i - S_{\hat{\imath}}\big] = \sigma(1 - r)e_N, \tag{4}$$

and the expected gain capture*, the fraction of best-of-N improvement over a random candidate retained under noisy selection, is*

$$\text{capture}(N, r) = \frac{\mathbb{E}[S_{\hat{\imath}}] - \mu}{\mathbb{E}[\max_i S_i] - \mu} = r, \qquad \textit{independent of } N. \tag{5}$$

Proof. Write $Z_i = r\hat{Z}_i + \sqrt{1 - r^2}\,\eta_i$ with $\eta_i \sim \mathcal{N}(0, 1)$ independent of $(\hat{Z}_j)_{j \le N}$, which reproduces Assumption 2. Since $\hat{\imath}$ depends only on $(\hat{Z}_j)$,

$$\mathbb{E}[\eta_{\hat{\imath}}] = \mathbb{E}\Big[\mathbb{E}[\eta_{\hat{\imath}} \mid (\hat{Z}_j)]\Big] = 0.$$

Thus $\mathbb{E}[Z_{\hat{\imath}}] = r\,\mathbb{E}[\hat{Z}_{\hat{\imath}}] = r\,\mathbb{E}[\max_i \hat{Z}_i] = re_N$. The oracle term follows from Eq. 2; subtract and rescale by σ. For Eq. 5, a random, or first, candidate has expected value μ. □

At $r = 0$, selection is uniform and $\mathbb{E}[S_{\hat{\imath}}] = \mu$. Thus for $r \ge 0$, expected value never falls below the no-search baseline, while the commit still returns a full-compute sample. We calibrate latent r from the observed per-prompt Spearman correlation ρ.

Corollary 2 (Spearman calibration)**.** *Under Assumption 2, the classical Gaussian rank-correlation relationship (Kruskal, 1958) maps the* 8*-seed gate pilot's median $\rho = 0.905$ to $r \approx 0.913$ and mean $\rho = 0.820$ to $r \approx 0.833$. The extended* 16*-seed grid has median $\rho = 0.891$, giving $r \approx 0.900$.*

On the extended 16-seed grid, Proposition 1 with median-calibrated r predicts roughly 0.90 capture at every width. Measured capture follows this value through N=8 and rises at N=16 and 32 (Figure 5b). Section A.5 separates finite-sample calibration bias from the Gaussian-marginal shape error in Assumption 2(ii) and corrects the prediction (Table 5); Proposition 3 explains the aggregate level.

A.3 Top-1 agreement and the zero-regret median

Proposition 2 (Top-1 agreement)**.** *Under Assumptions 1–2, with $\phi_2(\cdot, \cdot; r)$ and $\Phi_2(\cdot, \cdot; r)$ the standard bivariate normal density and CDF,*

$$p_1(N, r) = \Pr\Big[\arg\max_i Z_i = \arg\max_i \hat{Z}_i\Big] = N \int_{\mathbb{R}^2} \phi_2(z, \hat{z}; r)\, \Phi_2(z, \hat{z}; r)^{N-1}\, dz\, d\hat{z}. \tag{6}$$

This quantity increases with r and is numerically decreasing in N over the widths used here ($N \le 32$). Since ties have measure zero, $\Pr[\text{regret} = 0] = p_1(N, r)$, so median regret is 0 if and only if $p_1(N, r) \ge \frac{1}{2}$.

Proof. By exchangeability, $p_1 = N \Pr[Z_1 = \max_i Z_i, \hat{Z}_1 = \max_i \hat{Z}_i]$. Conditioning on $(Z_1, \hat{Z}_1) = (z, \hat{z})$, the remaining $N - 1$ pairs are i.i.d. bivariate normal, so the conditional probability is $\Phi_2(z, \hat{z}; r)^{N-1}$; integration against ϕ_2 gives Eq. 6. Monotonicity in r follows from Slepian's inequality applied to the orthant event $\{Z_j \le Z_1, \hat{Z}_j \le \hat{Z}_1\ \forall j\}$, whose cross-covariances increase with r. Monotonicity in N is not a pathwise inclusion, since a new candidate can top both lists, so we verify the decrease numerically for $N \le 32$ at both calibrations used here. □

At $N = 8$, $p_1(8, r) = 0.58$ for mean-calibrated $r = 0.833$ and 0.69 for median-calibrated $r = 0.913$. The measured top-1 agreement at $N = 8$ and $\tau = 0.10$ is 64%, inside this bracket. Both predictions exceed $\frac{1}{2}$, matching the measured zero median regret and the 64% of prompts with zero regret. Still, $\rho \approx 0.9$ implies roughly 30% top-1 disagreement, so safety depends on the cost of those disagreements, not on agreement alone. Proposition 1 bounds that cost by $\sigma(1-r)e_N$; the next result locates it.

A.4 Heterogeneous-prompt regret

Prompts differ in score spread σ_p and ranking fidelity r_p. Aggregating Proposition 1 therefore weights prompts by their spread.

Proposition 3 (Spread-weighted capture)**.** *Let prompts p have parameters (μ_p, σ_p, r_p) satisfying Assumptions 1–2 prompt-wise. Then $\mathcal{R}_p = \sigma_p(1 - r_p)e_N$, and aggregate capture is the* spread-weighted *mean correlation*

$$\overline{\text{capture}} = \frac{\mathbb{E}_p[\sigma_p r_p]}{\mathbb{E}_p[\sigma_p]} = \bar{r} + \frac{\text{Cov}_p(\sigma_p, r_p)}{\mathbb{E}_p[\sigma_p]}, \tag{7}$$

where $\bar{r} = \mathbb{E}_p[r_p]$. Hence, if spread and ranking fidelity are positively associated, $\text{Cov}_p(\sigma_p, r_p) > 0$, aggregate capture exceeds mean per-prompt capture $\bar{r}$. Moreover, $\mathcal{R}_p \le \sigma_p e_N$, so low-spread prompts incur little absolute regret.

Proof. Summing Proposition 1 over prompts gives oracle gain $\mathbb{E}_p[\sigma_p]e_N$ and CachedSearch gain $\mathbb{E}_p[\sigma_p r_p]e_N$; their ratio gives Eq. 7. The covariance identity follows from $\mathbb{E}[\sigma r] = \mathbb{E}[\sigma]\mathbb{E}[r] + \text{Cov}(\sigma, r)$. Finally, $\mathcal{R}_p \le \sigma_p e_N$ follows from Eq. 4 and $r_p \ge 0$. □

Ranking corruption is therefore self-limiting on low-spread prompts, where mistakes are cheap. On the 50-prompt gate grid, 64% of regret mass lies on below-median-spread prompts. Spread and ρ have Spearman association $+0.25$, and the $9/50$ prompts with $\rho < 0.7$ have low spread. Aggregate capture (94%) exceeds both mean-calibrated capture (0.83) and its spread-weighted correction (0.85).

Remark 1 (The model is conservative)**.** *The model correctly predicts zero median regret and self-limiting corruption, but predicts mean regret* 0.107 *instead of* 0.039 *and capture* 0.85 *instead of* 0.94*. It overestimates loss because it distributes disagreement across the score range, whereas observed swaps cluster among near-tied, mid-ranked candidates. Measured selection is therefore more stable than the model predicts.*

A.5 Capture versus search width

Equation 5 remains flat in N under prompt heterogeneity, because e_N cancels from Eq. 7 for any joint distribution of (σ_p, r_p). Measurements are not flat. Across all $\binom{16}{N}$ subsets of the extended 16-seed grid, capture is 90.6%, 90.4%, 91.8%, and 95.0% at $N = 2, 4, 8, 16$, a $+4.6$-point increase from N=4 to 16; a 32-seed extension remains at 94.4% (Section 4.3, Figure 5b). Heterogeneity cannot induce this N-dependence, so the within-prompt idealization must fail. Two effects explain the level and direction, with a remaining unmodeled component. The first is the score marginal: width independence follows from the Gaussian marginal, not from rank noise itself.

Proposition 4 (Capture under a general score marginal)**.** *Retain the Gaussian copula of Assumption 2(i) with $r > 0$, but let the full score have any continuous marginal F with mean μ_F and $\mathbb{E}|S| < \infty$, so $S_i = g_F(Z_i)$ with $g_F = F^{-1} \circ \Phi$ nondecreasing. Let $M_N = \max_{i \le N} \hat{Z}_i$ and $\eta \sim \mathcal{N}(0, 1)$ independently. Then*

$$\text{capture}(N, r, F) = \frac{\mathbb{E}\big[g_F\big(rM_N + \sqrt{1 - r^2}\,\eta\big)\big] - \mu_F}{\mathbb{E}[g_F(M_N)] - \mu_F}. \tag{8}$$

If F is Gaussian, capture equals r as in Proposition 1. If F is bounded above by $b = \operatorname{ess\,sup} S$, then both selected and oracle values approach b, and

$$\text{capture}(N, r, F) \longrightarrow 1 \qquad (N \to \infty). \tag{9}$$

Proof. As in Proposition 1,

$$Z_{\hat{\imath}} \overset{d}{=} rM_N + \sqrt{1 - r^2}\,\eta,$$

while $\max_i S_i = g_F(\max_i Z_i)$ with $\max_i Z_i \overset{d}{=} M_N$; a random candidate has mean μ_F. This gives Eq. 8, and the Gaussian case follows immediately. For bounded F, couple widths using maxima of the first N terms of one i.i.d. sequence. Then $M_N \uparrow \infty$ a.s., so both arguments in Eq. 8 increase a.s. Since $b - g_F(\cdot) \ge 0$ decreases pointwise to 0 and is integrable at $N = 2$, monotone convergence gives both expectations $\uparrow b$. Their ratio therefore tends to $(b - \mu_F)/(b - \mu_F) = 1$. □

Verifier scores are bounded in practice, with observed rewards in $[-2.3, 2.3]$, so capture approaches one as width grows, although this limit does not imply monotonicity at finite N. In this sense,

Eq. 5's flat extrapolation is conservative. The bounded-score model predicts the observed direction but underestimates its magnitude: plugging the observed 16-score vectors into Eq. 8 predicts less than one point of growth from $N{=}2$ to 16. The second effect is finite-sample attenuation in the calibration, which shifts the level.

Remark 2 (Finite-sample calibration attenuation)**.** *Finite-sample Spearman correlation understates the latent relationship. For n bivariate-normal pairs, Hoeffding and Kendall's identity gives $\mathbb{E}[r_S] = ((n-2)\rho + 3\tau_K)/(n+1)$, where τ_K is Kendall's tau. At $n = 16$, population $\rho = 0.905$ has expected sample value 0.875. Inverting this identity prompt-wise on the 16-seed grid raises the median corrected r_p from 0.900 to 0.927, the spread-weighted flat prediction from 0.879 to 0.903, and reduces the 8-seed regret overprediction from $2.7\times$ to $2.0\times$. The identity corrects a mean, while the sampling law of r_S is left-skewed: at $n = 16$ and $r = 0.913$, its median is 0.891. Thus the inversion is exact for the spread-weighted mean prediction but overcorrects the median; a median-consistent inversion gives 0.913. The correction improves the level but leaves the width trend flat, so the remaining error remains conservative.*

Table 5 applies both corrections without free parameters. For each prompt, it keeps the observed score vector, adds Gaussian rank noise at corrected r_p, and evaluates the ratio-of-mean-gains estimator over every seed subset; the sd column gives sampling variation from one modeled noise realization per prompt. Predictions match the level through $N = 8$ within sampling noise. At $N = 16$, the model is conservative by 5.7 points, so it captures the direction but not the full rise. Measured rank errors preserve the top of the list better than exchangeable noise: at $N = 16$, the cached winner lies in the true top two for 92% of prompts, versus 86% under the corrected Gaussian copula. We therefore keep the headline result empirical and use the model only as a conservative envelope.

Table 5: Predicted and measured gain capture (50 prompts, 16 seeds, $\tau = 0.10$). Δ: measured minus predicted, in points; sd and quantile: the model estimator's sampling standard deviation and the measured value's quantile under it.

N	meas.	pred. (sd)	Δ	quantile
2	90.6%	89.7% (1.0)	+0.9	0.81
4	90.4%	89.9% (1.3)	+0.5	0.65
8	91.8%	89.5% (2.0)	+2.3	0.87
16	95.0%	89.3% (3.9)	+5.7	0.94

A.6 Iso-cost comparison and break-even

Proposition 5 (Break-even and iso-cost widening)**.** *Let $\gamma = C_c/C_f$. (i) A full-compute commit costs less than full-compute best-of-N iff*

$$N > \frac{1}{1-\gamma}.$$

(ii) At fixed budget $B \geq C_f$, full search affords $N_f = B/C_f$ candidates, while CachedSearch affords

$$N_c = \left\lfloor \frac{B - C_f}{C_c} \right\rfloor = \left\lfloor \frac{N_f - 1}{\gamma} \right\rfloor$$

cached candidates plus one commit. Under Assumptions 1–2, CachedSearch has higher expected value at equal cost iff

$$re_{N_c} > e_{N_f}. \tag{10}$$

Proof. Part (i) is the break-even condition from Section 3.2. For (ii), Proposition 1 gives expected gains over no search of σre_{N_c} and σe_{N_f}, yielding Eq. 10. □

For measured $\gamma = 0.508$ and median-calibrated $r = 0.913$, $N_f = 2$ allows at most $N_c = 2$, and $re_2 = 0.52 < e_2 = 0.56$: cached search loses because the commit consumes the saving. At $N_f = 3$, $N_c = 3$ and $re_3 = 0.77 < e_3 = 0.85$; a fourth cached candidate would win ($re_4 = 0.94$) but exceeds budget by 1.1%. At $N_f = 4$, $N_c = 5$ fits and $re_5 = 1.06 > e_4 = 1.03$; a sixth candidate, 1.2% over budget, gives $re_6 = 1.16$.

The advantage is not monotone in budget. Since $N_c \approx 2N_f$ and $e_{2N}/e_N \to 1$, the ratio $re_{N_c}/e_{N_f} \to r < 1$. Under the Gaussian model, the iso-cost gain is therefore $+3\%$ at $N_f = 4$, peaks near $+7\%$ at $N_f = 6$–8, falls to $+5\%$ at $N_f = 16$ and $+3\%$ at 32, and vanishes beyond roughly $N_f \approx 100$, where both searches saturate and only cost differs. These two measured quantities, (γ, ρ), reproduce

the strategy simulation in Section 4: near parity at $N = 2$ and an iso-cost advantage from $N_f = 4$ onward over the budgets studied here.

A.7 Scope of the model

The parameter r belongs to the (model, cache threshold τ, verifier) triple and must be measured per prompt. Sweeping τ moves median ρ from 0.905 ($\tau \leq 0.10$) to 0.857 ($\tau = 0.20$). Assumption 2 is conservative in the sense documented by Remark 1. Assumption 3 excludes reward hacking, which caching neither causes nor fixes. KEEP falls outside these guarantees because it delivers the cached sample (Section 3.2). Finally, choosing τ online from the spread in Proposition 3 does not improve on the fixed-τ frontier (Section F.2): its noisy two-sample probe costs more than the nearly flat capture–speedup tradeoff returns.

B Theory against measurement

Appendix A predicts capture and cost from measured $\rho = 0.905$, which gives $r = 0.913$ through Corollary 2, and $\gamma = C_c/C_f = 0.508$, without fitting to search outcomes. Table 6 compares predictions with the 50×8 gate grid at $\tau = 0.10$ (Section 4, Appendix F). Calibration ranges span mean-calibrated $r = 0.833$ to median-calibrated $r = 0.913$; regret and spread-weighted capture use per-prompt statistics.

Table 6: Rank-noise predictions versus measurements (50 prompts, 8 seeds, $\tau = 0.10$). Capture is the ratio of mean gains above the mean full-compute single-sample score (Table 3).

Quantity	Predicted	Measured	Source
gain capture at N=8	83–91% ($= r$)	94.0%	Prop. 1; Tab. 3
capture trend, N=2/4/8	flat ($= r$)	90.1/89.8/94.0%	Prop. 1; Tab. 3
top-1 agreement $p_1(8, r)$	58–69%	64%	Prop. 2; Tab. 2
median regret	0 ($p_1 > \frac{1}{2}$)	0 (64% zero-regret)	Prop. 2; Sec. 4.2
mean regret (reward units)	0.107	0.039	Eq. 4; Sec. 4.2
aggregate capture (plug-in)	0.829 / 0.853 (spread-wtd.)	0.940	Prop. 3
sign of corr(spread, ρ)	not implied	+0.25	Prop. 3; App. F.4
regret mass, low-spread half	not implied	64%	Prop. 3
break-even width N^*	2.03	near parity at N=2	Prop. 5; Tab. 3
gain, N_c=6 vs. N_f=4	$re_6/e_4 - 1$ ($\approx +12\%$)	+6.6% (near-equal cost) 277 vs. 273 s	Cor. 1; Sec. 4.3

Top-1 agreement lies within the calibration range, and both calibrations predict zero median regret. Capture stays high across widths, although its rise departs from the flat prediction (Appendix A.5). The cost model places N=2 near parity. At near-equal cost, six cached candidates versus four full candidates yield +6.6% gain on the gate grid ($[0.7, 12.0]$%) and +11.1% on VBench (Huang et al., 2024), against the predicted 12%. At exactly equal cost, the respective gains are +5.8% and +9.8% (Section 4.3).

The model overestimates loss in these measurements. Equation 4, evaluated with per-prompt (σ_p, r_p), predicts mean regret 0.107 versus 0.039, a 2.7× overprediction, and capture 0.85 versus 0.94. The Gaussian model does not reproduce the observed concentration of swaps among near-tied, mid-ranked candidates (Remark 1; Figure 8), whose selection cost is small. Measured regret is lower than predicted at all three evaluated τ values. The model is therefore a conservative reference for these experiments, not a general deployment guarantee; headline results remain empirical.

C Method details

This appendix details the sampler used in Section 3.1 and the cache wrapper's update rules. We consider latent video diffusion models trained with flow matching, including Wan2.1-T2V-1.3B (Lipman et al., 2023; Liu et al., 2023; Wan Team, 2025a). A video is represented by a spatiotemporally compressed latent $\boldsymbol{x} \in \mathbb{R}^{C \times F \times H \times W}$. A diffusion transformer (DiT) $v_\theta(\boldsymbol{x}_t, t, c)$ predicts the velocity field transporting Gaussian noise $\boldsymbol{x}_1 \sim \mathcal{N}(0, \boldsymbol{I})$ to a data sample $\boldsymbol{x}_0$, conditioned on

Table 7: Model-card settings for six backbones: native resolution and length, guidance, VAE precision, and measured full and cached rollout latency at $\tau = 0.10$. All use $T = 50$ steps, with transformers and text encoders in bf16.

model	resolution × frames	guidance	VAE	$C_f \to C_c$ (τ=0.10)
WAN2.1-1.3B	480 × 832 × 81	5.0	fp32	68.3 → 34.7 s
LTX-VIDEO-2B	480 × 704 × 161	3.0	bf16	10.1 → 3.8 s
WAN2.2-TI2V-5B	704 × 1280 × 121	5.0	fp32	130.1 → 63.5 s
COGVIDEOX-5B	480 × 720 × 49	6.0	bf16	91.3 → 44.3 s
HUNYUANVIDEO-13B	480 × 720 × 61	6.0	bf16	87.1 → 39.8 s
WAN2.1-14B	480 × 832 × 81	5.0	fp32	341.3 → 166.8 s

text prompt c. Sampling integrates the probability-flow ODE over T steps on the decreasing grid $1 = t_1 > \cdots > t_T > t_{T+1} = 0$ using an Euler-type update,

$$\boldsymbol{x}_{t_{k+1}} \;=\; \boldsymbol{x}_{t_k} + (t_{k+1} - t_k)\,\tilde{v}_k, \tag{11}$$

followed by VAE decoding of $\boldsymbol{x}_{t_{T+1}}$ into pixels. With classifier-free guidance (CFG) (Ho & Salimans, 2022) at scale w, each step evaluates the transformer once per guidance branch and combines their predictions:

$$\tilde{v}_k \;=\; v_\theta(\boldsymbol{x}_{t_k}, t_k, \varnothing) + w\big(v_\theta(\boldsymbol{x}_{t_k}, t_k, c) - v_\theta(\boldsymbol{x}_{t_k}, t_k, \varnothing)\big), \tag{12}$$

A rollout therefore costs $2T$ transformer evaluations, which dominate wall-clock time. Since seed s fixes $\boldsymbol{x}_1$ and the solver is deterministic, the prompt and seed fully determine $G(c, s)$ (Section 2).

The wrapper of Section 3.1 intercepts every transformer call in Eq. 12. Let $\boldsymbol{x}$ denote the latent input of the current call and $v_\theta(\boldsymbol{x})$ its output. Rather than the output itself, the wrapper caches the transformation vector of the most recent computed call, whose input we denote $\boldsymbol{x}_{\mathrm{ref}}$, and a skipped call returns an approximation built from it:

$$\Delta \;=\; v_\theta(\boldsymbol{x}_{\mathrm{ref}}) - \boldsymbol{x}_{\mathrm{ref}}, \tag{13}$$

$$\hat{v}_\theta(\boldsymbol{x}) \;=\; \boldsymbol{x} + \Delta, \tag{14}$$

which assumes that the residual $v_\theta(\boldsymbol{x}) - \boldsymbol{x}$ varies slowly across adjacent steps even where v_θ itself does not. Under CFG the conditional and unconditional inputs of a step differ and drift at different rates, so the wrapper keeps fully independent state $(\Delta, \boldsymbol{x}_{\mathrm{ref}}, a)$ per guidance branch, and each branch schedules its own skips. The wrapper is training-free and model-agnostic, since it replaces the transformer module of an existing pipeline with no model changes, and it preserves determinism: for fixed (c, s, τ), the cached rollout $G_\tau(c, s)$ is deterministic, exactly like $G(c, s)$, which off mode recovers. A static variant that computes every k-th step between warmup and cooldown serves as a uniform-schedule baseline.

D EXPERIMENTAL DETAILS AND REPRODUCIBILITY

All results use public checkpoints, fixed seeds, and the configurations below. On a fixed stack, generation is deterministic in (prompt, seed, τ), so each cached or full video can be regenerated exactly. A no-cache control runs the full-compute arm through the same harness with the wrapper disabled and matches reference scores bit-exactly on every overlapping record (Appendix D.5). We will release the caching wrapper, search harness, analysis and figure scripts, prompt lists, seeds, configurations, and evaluation scores.

D.1 MODELS AND GENERATION SETTINGS

We use Wan2.1-T2V-1.3B (Wan Team, 2025a), Wan2.1-T2V-14B (Wan Team, 2025a), Wan2.2-TI2V-5B (Wan Team, 2025b;a), CogVideoX-5B (Yang et al., 2024), HunyuanVideo-13B (Kong et al., 2024), and LTX-Video-2B (HaCohen et al., 2024), each at its native resolution, length, and guidance (Table 7). Cached and full arms share all precision settings. The three Wan models decode with an fp32 VAE; CogVideoX, HunyuanVideo, and LTX-Video use bf16 throughout, following their official recipes. Guidance follows each pipeline: Wan makes separate conditional and unconditional transformer calls per step; CogVideoX and LTX-Video batch both branches into one call; HunyuanVideo uses distilled guidance with one call per step.

Table 8: Published training-free caches as exploration engines. Four methods use the same 50-prompt, 8-seed gate protocol and full references. Rows report candidate fidelity, speedup, and best-of-8 commit cost; *reuses* names the cheaply recomputed component. Capture is the mean of per-prompt gain ratios (as in Table 2); *delivered* is the ratio of mean gains of the recommitted winners over the mean single sample, the estimator of Figure 4 and Table 3.

exploration engine	reuses	speedup ↑	med ρ ↑	p10 ↑	top-1 ↑	regret ↓	capture ↑	delivered ↑	cost (s) ↓
no caching (control)	-	0.99×	1.000	1.000	100%	0.000	100%	100%	621
CachedSearch + PAB (Zhao et al., 2024)	attn. outputs	1.29×	0.976	**0.950**	90%	**0.004**	99.5%	**99.5%**	493
CachedSearch + CFG-Cache (Lv et al., 2024)	uncond. branch	1.38×	**1.000**	0.929	**92%**	**0.004**	**99.6%**	**99.5%**	465
CachedSearch + TeaCache (Liu et al., 2024)	full model	1.84×	0.929	0.783	68%	0.029	93.2%	95.6%	365
CachedSearch + EasyCache (Zhou et al., 2025) (default)	full model	**1.97**×	0.905	0.614	64%	0.039	90.1%	94.0%	**346**

D.2 Cache wrapper

The adaptive wrapper uses EasyCache's transformation-vector formulation (Zhou et al., 2025) (Section 3.1): it caches $\Delta = v(x) - x$ and returns $x + \Delta$ on skipped steps. The skip rule accumulates $a \mathrel{+}= \|x - x_{\text{ref}}\|_F / \|x_{\text{ref}}\|_F$, where x_{ref} is the last computed input. When $a > \tau$, the wrapper recomputes, refreshes (x_{ref}, Δ), and resets a. The first $K_w = 5$ and last $K_c = 5$ steps always compute, as do steps with an empty cache. The default is $\tau = 0.10$; sweeps also use $\{0.05, 0.20\}$. At $\tau = 0.10$, the wrapper skips $26/50$ steps on both Wan and CogVideoX-5B. Table 15 gives skip fractions for all six models, and Table 11 shows their dependence on schedule length and resolution. CogVideoX's batch-concatenated guidance uses a single-branch wrapper variant. The cache resets between generations.

D.3 Protocol and verifier

Each grid contains 50 prompts and seeds 0–7, with paired full and cached generations: 800 scored rollouts per model at $\tau = 0.10$, plus 400 cached rollouts per additional threshold, reusing the same full references. Noise comes from a per-candidate `torch.Generator` initialized with that candidate's seed. VBench (Huang et al., 2024) and VBench-2.0 (Zheng et al., 2025) use the same paired 8-seed protocol. The verifier averages ImageReward (Xu et al., 2023) v1.0 scores over 8 uniformly spaced frames, converted to uint8 before scoring. The temporal study (Appendix F.1) materializes 50 winner pairs per τ (300 videos total) and evaluates DINO subject consistency, CLIP background consistency, motion smoothness, temporal flicker, mean optical flow, and LPIPS (Zhang et al., 2018). Intervals in the text are 95% percentile intervals from a prompt-level bootstrap that jointly resamples paired strategies.

D.4 Hardware, software, and timing

All rollouts ran on an NVIDIA GH200 node with one Grace Hopper superchip. Each used one GPU, batch size 1, and one generation process per GPU. In our tests, batching neither improved throughput nor preserved bit-exact outputs (Appendix F.2). Generation used Python 3.11, PyTorch 2.11 with CUDA 12.8, Diffusers 0.39, and Transformers 4.49. Attention used Diffusers' default backend. Speedups compare paired runs on the same backend; both ratios and absolute latencies remain specific to the tested hardware and software.

Latency measures wall-clock time for one pipeline call, text encoding, 50 denoising steps, VAE decoding, and transfer of frames to host memory, which waits for GPU completion. It excludes model loading, verifier scoring, and video writing. Full and cached rollouts for each (prompt, seed) run back to back in the same process. Reported latencies average all rollouts of each arm ($n = 400$ per arm on the gate grid). The paper's generation records total about 871 GPU-hours of timed rollouts (51,907 rollouts), excluding loading, scoring, and retries. Verifiers use separate environments: VideoScore v1.1 in bf16, VQAScore through t2v-metrics with CLIP-FlanT5-XXL, and VBench (Huang et al., 2024) metrics through VBench 0.1.5. In the full metric suite, aesthetic quality, subject and background consistency, temporal flickering, and LPIPS (Zhang et al., 2018) were computed on CPU.

D.5 Caching baselines

Figure 4 compares four published training-free caches as exploration engines. Table 8 reports their full metrics under the gate protocol; the following paragraph gives reproduction details for each port.

TeaCache (Liu et al., 2024) runs the official Wan2.1 (Wan Team, 2025a) rule with the published 1.3B polynomial coefficients and threshold 0.08. Below threshold, it skips all transformer blocks and reuses the cached token residual ($24/50$ steps per CFG branch). PAB (Zhao et al., 2024) broadcasts attention *outputs* across steps (official VideoSys rule). Wan has no official configuration, so its unified 3D self-attention uses the CogVideoX (Yang et al., 2024) "spatial" range 2; cross-attention uses OpenSora's (Zheng et al., 2024) range 6. FasterCache's CFG-Cache (Lv et al., 2024) is a verbatim port with the official frequency-compensation constants. It rebuilds $26/50$ unconditional forwards from conditional outputs with boosted low/high FFT deltas. This isolates CFG reuse from attention reuse (PAB) and full-model reuse (TeaCache and our EasyCache-style rule (Zhou et al., 2025)). A *no-caching* control regenerates the full-compute variant in the same harness. It reproduces all overlapping scores bit-exactly, with every common record identical ($\rho = 1$, speedup $0.99\times$, interval $[0.96, 1.02]$). This validates seed determinism and the protocol.

E Qualitative examples

E.1 Selection across seeds and backbones

Figure 7 in Section 4.5 shows the videos returned by search. Figure 10 adds two same-pick prompts in the same three-column layout. Figure 11



Figure 10: Two additional same-pick VBench (Huang et al., 2024) prompts, with the columns of Figure 7: single sample, full best-of-8, and COMMIT at $\tau = 0.10$. Chips give ImageReward (Xu et al., 2023) and seed.

E.2 Appearance and motion of cached rollouts

The same-seed pairs in Figure 2 illustrate the LPIPS (Zhang et al., 2018) differences reported in Appendix F.1 at $\tau = 0.10$: mean 0.142, range 0.021–0.343 over $n = 50$ suite pairs. Across the displayed range, cached and full videos share subject, layout, and palette but differ in fine texture and motion phase.

Caching also dampens motion, an effect neither learned judge detects (Section F.1). Figure 12 shows optical flow maps for same-seed winner pairs at $\tau = 0.20$. Cached videos retain layout and identity, but the change in mean flow varies by pair; across all 50 pairs, mean optical flow falls by 8.0%. Figure 13 compares KEEP and COMMIT at this threshold for the pairs with the largest mean frame difference: KEEP preserves layout, identity, and style but reduces displacement.

E.3 Delivery strategies and backbones

Figure 14 includes KEEP and a ranking failure. The single samples show a melted clock face, omit the handbag, or lack a couch, while full best-of-8 finds candidates that satisfy the prompts. For the clock and couch, cached exploration selects the same seed, returning a bit-exact COMMIT result at 64% of full-search cost or a visually near-identical KEEP result at 51%. For the hot dog, it selects a different seed, losing 0.49 reward out of a +3.2 full-search gain.

Figure 15 compares one prompt and seed across all six backbones at the same τ. In these examples, backbones within their calibrated skip regime produce cached videos that differ from their full-compute counterparts mainly in instance-level detail. LTX-Video is the exception: the fixed threshold is too aggressive, and its cached video visibly diverges. These examples illustrate appearance changes under caching; ranking fidelity is evaluated separately in Section 4.4.

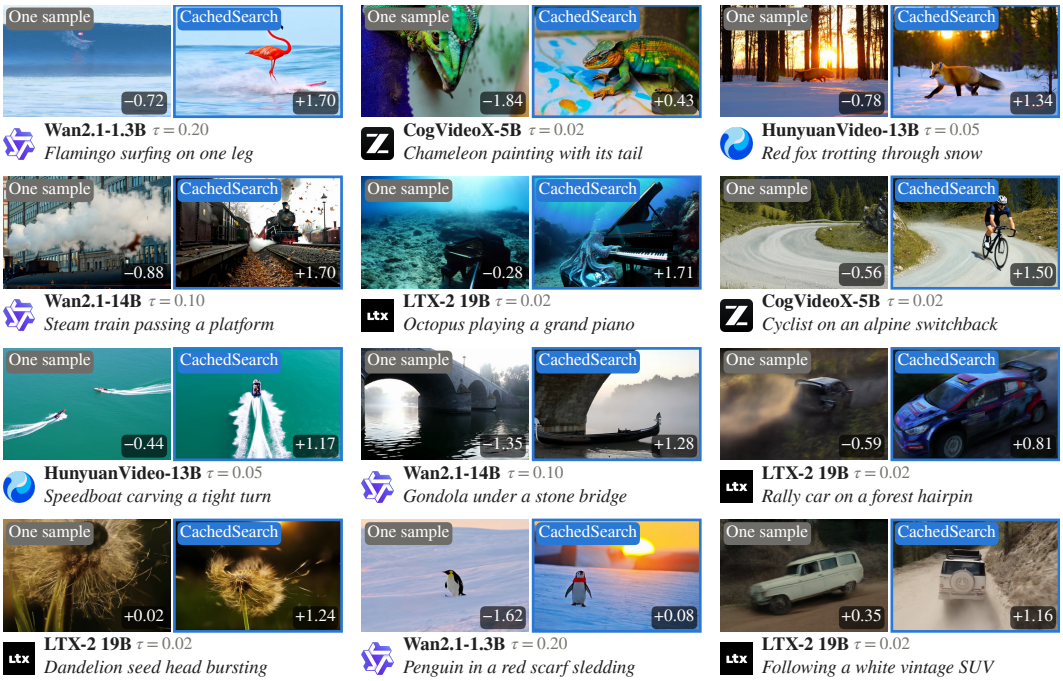


Figure 11: Left of each pair: the full-compute video sample from BoN; right: the full-compute video of the seed CachedSearch selects from its cached scores, at the same frame index and crop. Each backbone runs at the threshold printed under it, calibrated on these prompts with the 85% capture rule. The selected seed is the full-compute best of 8 in six pairs, second in five (flamingo, chameleon, rally car, dandelion, SUV).

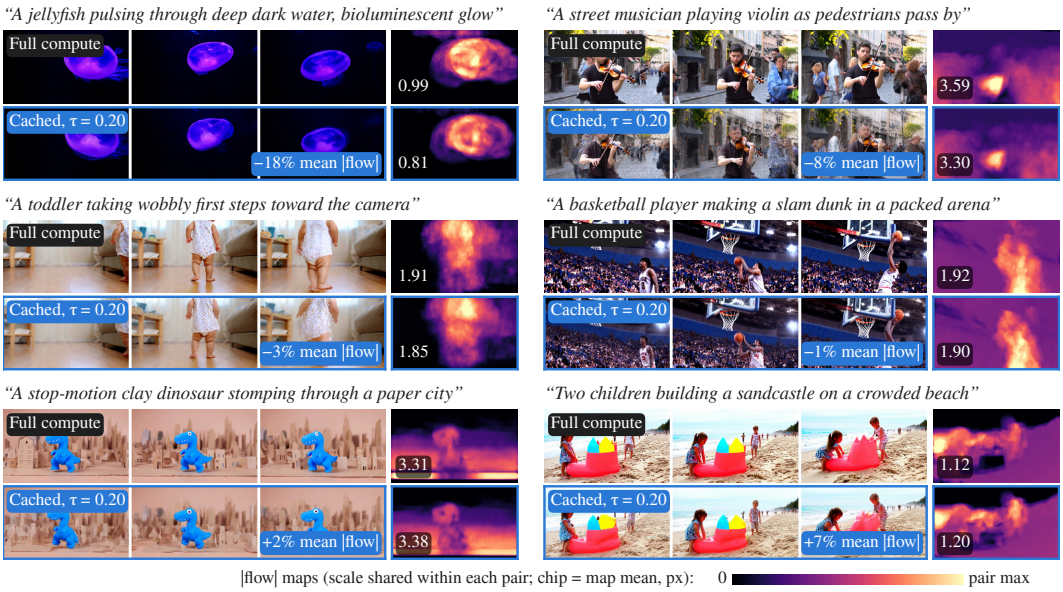


Figure 12: Six same-seed winner pairs at $\tau = 0.20$, full compute above and cached below (blue outline): three frames and the temporal-mean Farneback optical-flow magnitude, with the color scale shared within each pair. Selection: the 25 of 50 pairs with above-median full-compute flow are sorted by that change, and the pairs at ranks 3, 7, 12, 16, 21 and 25 are shown, left to right and top to bottom; ranks 1 and 2 are in Figure 9.

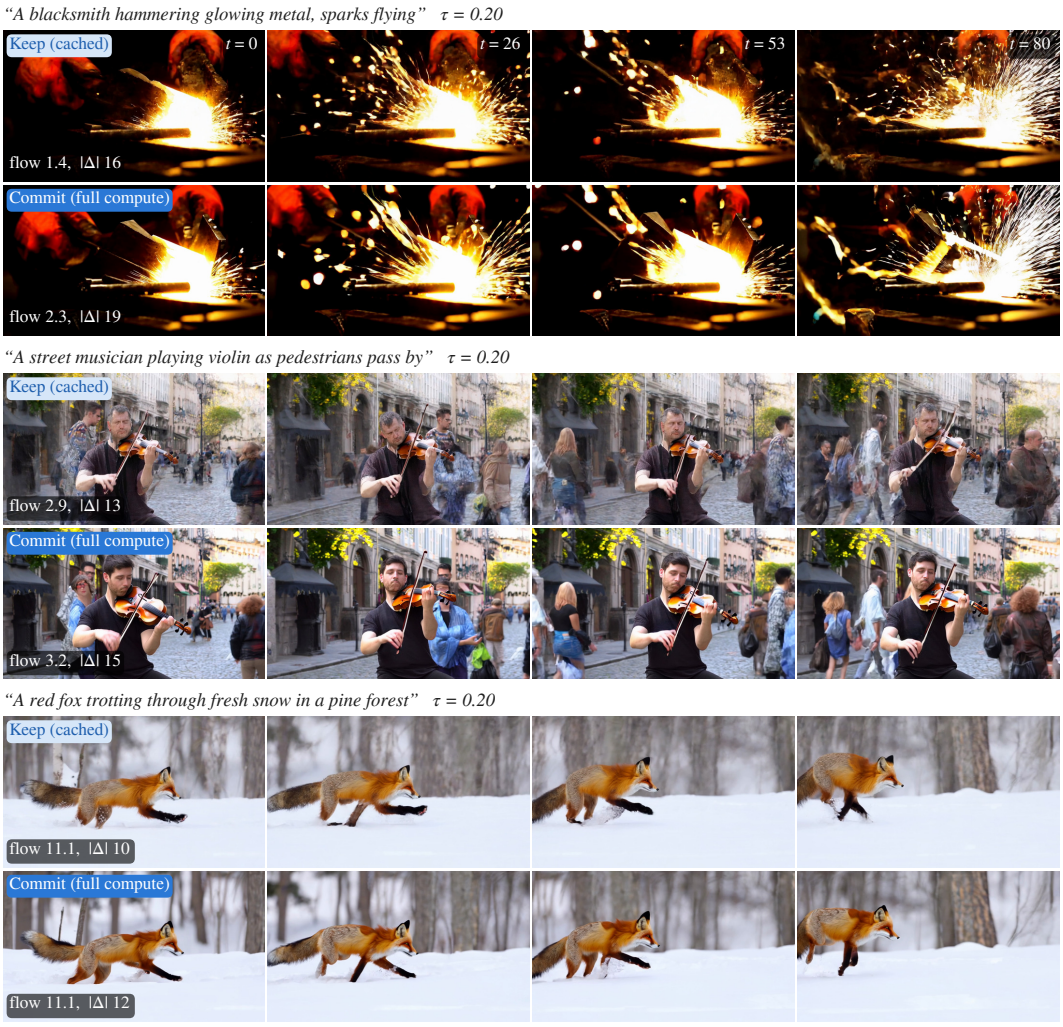


Figure 13: KEEP (top row) and COMMIT (bottom row) winners at $\tau = 0.20$, shown in four evenly spaced frames. These pairs have the largest mean frame difference among the 50 winner pairs. Chips give RAFT mean flow magnitude and $|\Delta|$, the mean absolute difference between consecutive frames in 8-bit pixel values.



Figure 14: All four delivery strategies on three VBench (Huang et al., 2024) prompts: single sample, full best-of-8, commit, and keep. Top-row chips give cost relative to full best-of-8; corner chips give measured delivered ImageReward (Xu et al., 2023).

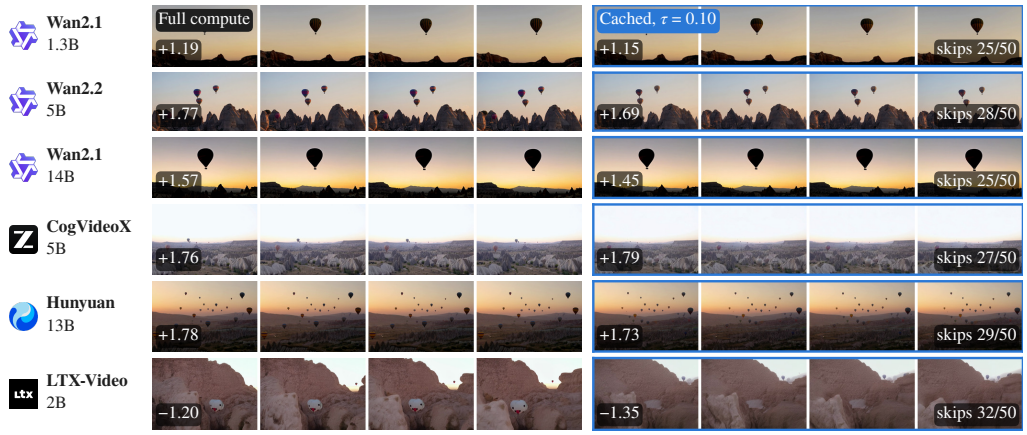


Figure 15: One shared prompt and seed on six backbones at $\tau = 0.10$, full compute (left) and cached (right) under each model-card recipe. Chips report ImageReward (Xu et al., 2023) and reused denoising steps. CogVideoX-5B renders this seed as a pale, overexposed sky in both modes.

F Additional results and ablations

This appendix gives the search and cache ablations supporting Sections 4 and 5; Appendices G and H cover verifiers and backbones. Unless noted, runs use the gate grid: Wan2.1-T2V-1.3B, 50 prompts, seeds 0–7, $\tau = 0.10$, $N = 8$, and ImageReward (Xu et al., 2023) as verifier.

F.1 Search strategies, width, and delivery mode

Figure 16 plots reward gain against cost for the strategies in Table 3; commit costs less than full best-of-N above the break-even width. Pruning (Section 5) scores 4 frames decoded from the clean-sample estimate at step 20, a preview about 21× cheaper than a full decode. Exploration costs 175 s per prompt versus 278 s without pruning; the 256 s total includes preview decoding, scoring, and the 68 s commit. Median regret remains zero. Pruning retains 93.3% of the gain at 3.75 C_f, versus 5.07 C_f without pruning. This capture is a ratio of mean gains; the 88.6% in Section 5 averages per-prompt ratios.

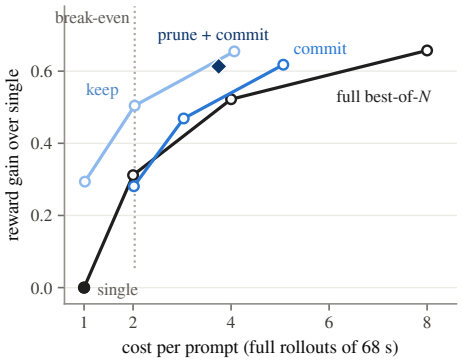


Figure 16: Reward gain over the mean full-compute single sample versus cost in full rollouts ($C_f = 68$ s), at $N = 2, 4, 8$ (markers, left to right) and $\tau = 0.10$. Dotted: break-even width $N^* = 1/(1 - C_c/C_f) \approx 2$. Diamond: top-4 pruning after step 20, followed by commit. Capture is the ratio of mean gains.

We re-evaluate all four strategies over every $\binom{16}{N}$ subset of the extended 16-seed grid, adding N=32 from a 32-seed extension. Commit capture rises overall (Figure 5b; Section 4), while top-1 agreement falls from 86% at N=2 to roughly 65–70% at $N \geq 8$. Relative commit cost falls toward the exploration ratio: 75.8/63.3/57.1/53.9% of full best-of-N at $N = 4, 8, 16, 32$, respectively. From $N = 16$ to 32, capture falls by 0.6 points, within noise. A saturating fit, $\text{capture}(N) = c_\infty - \beta/N$, gives $c_\infty = 94.2\%$ ($R^2 = 0.64$), a five-point consistency check rather than an established law. The flat spread-weighted copula prediction (88%; Remark 1) misses the rise; Appendix A.5 examines the Gaussian-marginal idealization. Top-1 agreement alone therefore misses the improved gain capture at larger widths, where cost savings are greatest.

Section 5 quantifies the motion cost of keep: mean optical flow falls by 3.1% at $\tau = 0.10$ and 8.0% at $\tau = 0.20$ relative to the winners' full-compute counterparts. We score all 300 winner videos (50 pairs per τ) with VBench (Huang et al., 2024)-style metrics: DINO subject consistency, CLIP background consistency, warping-error motion smoothness, temporal flickering, and mean optical flow. Apart from flow, these metrics change little: motion smoothness changes by $|\Delta| \leq 0.0003$,

Table 9: Settings for quality, speed, motion, width, and new backbones, measured on Wan2.1-T2V-1.3B (Wan Team, 2025a) at $N = 8$ unless noted. Cost is relative to full best-of-N. Capture is the ratio of mean gains over the mean single sample (Table 3), except the τ=0.05 row (†), which is the mean of per-prompt ratios from Table 10. Keep capture is nominal.

deployment goal	mode	τ	expected capture ↑	cost ↓
guaranteed quality (default)	commit	0.10	94.0%	63%
maximum speed (drafts)	keep	0.10	99.6% nominal	51%
motion-critical prompts	commit	0.05	93.6%†	76%
widest search, fixed budget	commit	0.10, N=16	95.0%	57%
new architecture or schedule	commit	calibrate	see Tab. 16, Tab. 11	pilot calibration

Table 10: Cache-threshold sweep (50 prompts × 8 seeds, shared full-compute references). Regret and capture use $N = 8$; capture averages per-prompt gain ratios. Shading marks the default.

τ	speedup ↑	median ρ ↑	p10 ρ ↑	top-1 ↑	regret ↓	capture ↑
0.05	1.58×	0.905	0.738	70%	0.025	93.6%
0.10	1.97×	0.905	0.614	64%	0.039	90.1%
0.20	2.41×	0.857	0.474	52%	0.057	88.3%

Table 11: Schedule length and resolution on Wan2.1-1.3B (Wan Team, 2025a) at $\tau = 0.10$, with 50 prompts and 8 seeds per row. *Skip* is the fraction of reused denoising steps; shading marks the default.

configuration	med. ρ ↑	p10 ↑	top-1 ↑	regret ↓	capture ↑	skip	speedup ↑
25 steps	0.964	0.85	86%	0.019	95.8%	32%	1.41×
50 steps (default)	0.905	0.61	64%	0.039	90.1%	51%	1.97×
100 steps	0.798	0.46	62%	0.126	82.5%	66%	2.80×
720×1280, 50 steps	0.881	0.69	66%	0.059	90.8%	53%	2.05×

flickering improves slightly (+0.0007 to +0.0015), background consistency is unchanged, and subject consistency falls by at most 0.005. The videos nevertheless differ: mean LPIPS (Zhang et al., 2018) between KEEP and COMMIT rises from 0.122 to 0.142 to 0.170 at $\tau = 0.05/0.10/0.20$ ($n = 50$ each). Both ImageReward (Xu et al., 2023) and VideoScore (He et al., 2024) prefer cached outputs despite the motion reduction (Appendix G). We recommend KEEP only at $\tau \leq 0.10$ when motion is secondary; COMMIT remains the default, especially for aggressive caching or motion-critical prompts. Table 9 summarizes deployment settings; Appendix E shows examples and flow maps.

F.2 Cache threshold, schedule length, and batching

Higher thresholds trade ranking fidelity for speed (Table 10): candidate speedup rises from 1.58× to 2.41× as p10 ρ falls from 0.74 to 0.47. The default $\tau = 0.10$ nearly doubles exploration speed, retaining 90.1% mean per-prompt capture or 94.0% as the ratio of mean gains used in Table 3.

The association between candidate spread and ranking fidelity (Appendix F.4) motivates a per-prompt threshold. We simulate this policy on the three-threshold grid with shared prompts and seeds, without new generation. The rule probes $K = 2$ candidates at $\tau = 0.20$. If their score gap exceeds t, it runs the remaining 6 at $\tau = 0.20$; otherwise, it runs all 8 at $\tau = 0.05$, counting the probes as sunk cost. Sweeping t traces a frontier that never exceeds the fixed-τ curve (Figure 17). At matched 1.97× speedup, capture is 89.7% versus 90.1% for $\tau = 0.10$: a paired difference of −0.4 points, statistically indistinguishable from zero. The conservative endpoint reaches 93.6% at only 1.36×, versus 1.58× for fixed $\tau = 0.05$. The ratio-of-means estimator and reversed low-spread rule give the same conclusion. Capture varies by only 5.3 points across a 1.5× speedup range, leaving little headroom; the two-sample spread estimate is noisy, and fallback adds probe overhead. A signal already available during the first rollout, such as the wrapper's cache-error indicator, may avoid this overhead. Fixed $\tau = 0.10$ remains our default.

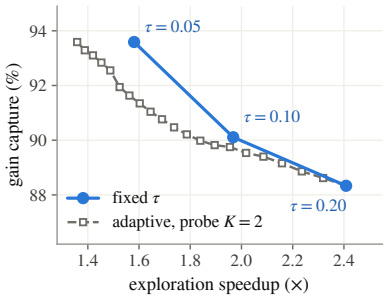


Figure 17: Gain capture versus exploration speedup for a per-prompt adaptive rule and fixed τ ($N = 8$, 50 prompts).

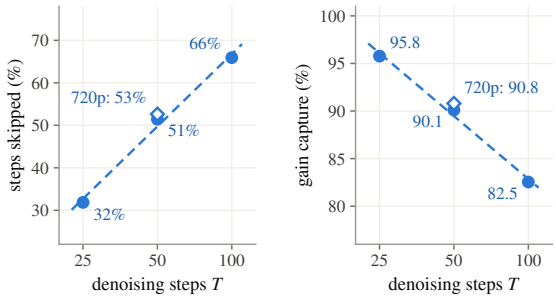


Figure 18: Skip fraction (left) and gain capture (right) versus denoising steps ($\tau = 0.10$, 50 prompts). Dashed: log-linear trends; open diamond: the 720×1280 run.

Schedule length changes effective cache aggressiveness alongside threshold and architecture (Appendix H.1). Table 11 varies Wan2.1-1.3B's default 50 steps and 480×832 resolution at fixed $\tau = 0.10$. Finer schedules take smaller steps, allowing more reuse under the same drift threshold: from 25 to 100 steps, the skip fraction rises from 32% to 66% and speedup from 1.41× to 2.80×, while capture falls from 95.8% to 82.5%. Figure 18 fits skip and capture, both in percent, as $-46+24.6 \ln T$ and $127 - 9.5 \ln T$, respectively. These are trends across three schedule lengths, not scaling laws, and are consistent with the U-shaped redundancy profile in Section I.1: finer schedules allocate more steps to the flatter middle of the trajectory. Skip fractions vary little at fixed T, while capture varies across prompts. All three points lie within 2.0 points of the Wan-1.3B frontier in Figure 6(a), which was fitted without these schedule runs. Longer schedules therefore offer larger savings here; 25 steps leave less to reuse but preserve more gain, resembling the limited reuse in few-step distilled samplers. At 720×1280 (2.3× the pixels), the operating point stays near the default. These results support recalibrating τ after schedule changes; the threshold transferred across the two tested resolutions.

Batching did not replace caching in our GH200 tests. We benchmarked candidate batch sizes $\{1, 2, 4, 6, 8\}$ at the study configuration on one GPU. From batch 1 to 2, full-compute throughput falls from 0.92 to 0.90 videos/min and cached throughput improves by only 8%; batch 8 is infeasible. At 32K tokens per sample, these results suggest little batching headroom for Wan-1.3B (Wan Team, 2025a). Batching also changes the seed-to-video mapping: a seed's batched output differs from its single-sample output (maximum first-frame pixel deviation 0.6). Exact agreement with a batched reference would require reproducing its generation configuration at commit. We therefore use per-rollout latency for this workload; the measured batching gains do not close the roughly 2× cost gap.

F.3 Truncation and reduced resolution as cheap proxies

Table 1 compares caching with fewer denoising steps. The truncation arm uses existing full-compute $T = 25$ rollouts as drafts and evaluates their selection against $T = 50$ references, giving roughly the same 2× exploration speedup as caching. Truncation ranks outputs from a shorter schedule rather than the 50-step outputs delivered by commit; caching retains the target schedule and initial noise. Changing resolution also changes the latent shape, so reusing the seed does not preserve the full-resolution initial noise. We regenerate the gate grid at 352×608, roughly half the pixels of the 480×832 default, and rank these drafts against the same full references. This is the cheapest proxy, but its capture is -0.8% with prompt-bootstrap interval $[-27.2, 23.5]\%$: there is no statistically resolved gain over random selection. This differs from Appendix F.2, where cached and full arms both use 720×1280. The tested threshold transfers when both arms change resolution, but direct seed reuse across resolutions does not establish a reliable ranking proxy. Regenerated full references are bit-identical to the published ones, so all rows share one reference grid. Re-running the default arm also reproduces median $\rho = 0.905$, 64% top-1 agreement, 0.039 regret, and 90.1% capture.

F.4 Where ranking errors fall

Figure 19 shows zero regret on 64% of gate prompts and 72% of VBench (Huang et al., 2024) prompts at $N = 8$, with much lower regret than random selection at every tested threshold. On VBench (Huang et al., 2024), prompts with $\rho < 0.7$ have lower spread (Mann–Whitney $p = 3\times10^{-10}$; Figure 8, Section 5). This is consistent with ranking errors concentrating where candidate separation and attainable gain are small (Eq. 7). Across spread quartiles, capture rises from 81% (lowest quartile, median $\rho = 0.857$) to 96% (highest, median $\rho = 0.952$), while attainable gain more than triples (0.36 → 1.27 reward units). Regret does not rise with spread in these data (corr(spread, regret) $= -0.11$). The low-correlation tail still matters: the 11% of prompts with

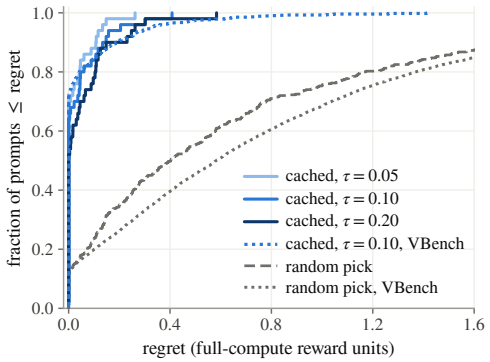


Figure 19: Per-prompt regret CDFs at $N = 8$ versus pooled random-pick baselines on the gate grid (solid) and VBench (Huang et al., 2024) suite (dotted).

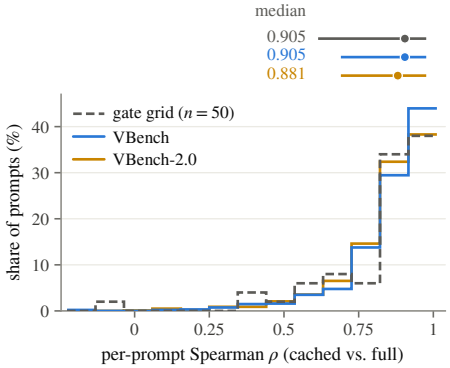


Figure 20: Per-prompt Spearman ρ between cached ($\tau = 0.10$) and full-compute ImageReward (Xu et al., 2023) scores of Wan2.1-T2V-1.3B on the three suites in Table 2, with medians marked.

$\rho < 0.7$ average 0.161 regret versus 0.043 elsewhere, accounting for 32% of total regret. Within this group, 51% still select the exact winner, and capture is 68%. Despite these failures, mean regret (0.056 on VBench (Huang et al., 2024), 0.039 on the gate grid) remains an order of magnitude below random selection.

Keyword groups probe whether ranking errors vary by content. Each gate prompt belongs to one group: humans, animals, nature (landscapes, weather, natural phenomena), objects (vehicles, machines, manufactured items), or stylized (non-photorealistic). Nature and animals have higher top-1 agreement and lower mean regret, although animals have the lowest median ρ, consistent with near-tie swaps (Table 12). Humans and objects have higher mean regret. Stylized prompts have the highest mean regret but only $n = 4$: two contribute 0.302 and 0.148, and two contribute zero. The worst individual prompt is an object case, "robot arm assembling electronics" ($\rho = -0.07$, regret 0.408). One hypothesis is that non-photorealistic and fine-mechanism content, such as stop-motion, unfolding origami, and articulated manipulation, is more sensitive to mid-trajectory reuse. These small groups suggest failure modes, not population-level conclusions; the tested spread probe does not improve the tradeoff (Appendix F.2).

Table 12: Keyword categories of the 50 gate prompts at $\tau = 0.10$ and $N = 8$. *Spread* is the mean within-prompt score SD; regret is in reward units.

category	n	med. ρ	regret	top-1	spread
humans	18	0.881	0.045	56%	0.53
animals	9	0.833	0.008	78%	0.51
nature	10	0.929	0.005	80%	0.53
objects	9	0.857	0.066	56%	0.43
stylized	4	0.917	0.112	50%	0.54

Both earlier suites follow the VBench (Huang et al., 2024) prompt style. VBench-2.0 (Zheng et al., 2025) tests intrinsic faithfulness through compositional interactions, physics, commonsense, and camera control, posing a harder test for current generators. Headline quantities remain statistically consistent with VBench (Huang et al., 2024) (Table 2), and all three ρ distributions share a high-correlation mode (Figure 20). Median $\rho = 0.881$ is one 8-candidate lattice step below 0.905; the means differ by -0.011 (0.848 versus 0.859). The harder prompts have tighter candidate spreads: mean 0.49 versus 0.56, a statistically reliable difference. The spread–fidelity association persists (corr(spread, ρ) $= +0.29$, $p = 8\times10^{-21}$), yet capture changes by only -0.9 points, statistically consistent with zero. These results support the same low-spread error pattern across both suites.

G Verifier choice and verifier bias

The main results use ImageReward (Xu et al., 2023), a frame-level image-text reward. Here we repeat the ranking test with two other verifiers, re-score delivered videos with a standard metric suite, and examine the preference for cached outputs.

G.1 A video-native verifier

We rescore the ranking and regret study with VideoScore (He et al., 2024) v1.1. It rates visual quality, temporal consistency, dynamic degree, text alignment, and factual consistency over 24 frames, so unlike ImageReward it observes motion. The data are all 300 winner-pair videos (Section F.1) and 23,542 VBench-suite rollouts (Section 4.2), which yield $n = 467$ prompts with validated, complete $8\times$full $+ 8\times$cached coverage. A score-consistency check removes ambiguous prompt matches, and the retained subset reproduces the VBench suite's median ρ, top-1 agreement, and capture.

Ranking preservation is weaker when VideoScore barely separates the candidates. Cached rankings still exceed chance (Table 13), with 55% zero regret and dimension medians of 0.73–0.79, but they trail ImageReward on the same videos. Two measurements explain the drop. First, candidate scores are nearly tied: VideoScore's median within-prompt spread is only 0.083 on $[1, 4]$, against 0.55 for ImageReward, so its rankings occupy the near-tied regime of Section F.4, where rank noise is large but mistaken picks are cheap. Second, the two verifiers agree less with each other than cached agrees with full: median ρ(IR, VS) is $+0.24$ within prompts and $+0.13$ over 7,701 pooled videos, against 0.905 and 0.762 for cached-to-full. The perturbation from caching is smaller than the standing disagreement between the judges, so verifier choice changes the ranking more than cached execution does. This separates approximation error from ordinary evaluator disagreement.

Table 13: Cached-versus-full ranking agreement under three verifiers, each beside ImageReward on the same videos. Last column: median within-prompt ρ with ImageReward (full-compute rollouts for VQAScore).

		ρ				ρ vs.
verifier	n	median	p10	top-1	capture	IR
VBench subset						
ImageReward	467	0.905	–	73%	91%	–
VideoScore	467	0.762	0.31	54%	65.6%	+0.24
Gate grid, regenerated						
ImageReward	49	0.905	0.581	65%	90.2%	–
VQAScore	49	0.810	0.495	57%	80.7%	0.548

Caching adds at most a marginal cross-verifier loss. Delivery uses the selected seed, not the full candidate ranking, so the decision-relevant test judges each IR pick by full-compute VideoScore. The cached-exploration IR winner captures 18.6% of the attainable gain, against 21.2% for the full-compute IR winner (mean VS-regret 0.245 vs. 0.230; random baseline 0.263; $n = 467$). The paired difference is -2.5 capture points, statistically consistent with zero, and regret increases by $+0.015$ on VideoScore's $[1, 4]$ scale. Both effects are small beside the verifier choice itself: even the full-compute ImageReward winner captures only 21.2% of the VideoScore gain. VideoScore judges ImageReward's cached and full-compute picks almost identically, and most cross-verifier loss comes from choosing ImageReward, not from caching its search.

G.2 A second image-text verifier

We repeat the test with VQAScore (Lin et al., 2024), an image-text alignment score, on the regenerated gate grid ($n = 49$ prompts with complete coverage after score-consistency filtering). Ranking preservation under caching is again weaker than for ImageReward on the same videos but far from absent (Table 13). The comparison that matters is not between those two rows. On full-compute rollouts, where no caching is involved, the two verifiers rank the same eight candidates at median $\rho = 0.548$, so each agrees with its own cached ranking (0.905 and 0.810) substantially better than with the other. The perturbation that caching introduces is smaller than the standing disagreement between two reasonable definitions of quality, which is the conclusion Section G.1 reaches from VideoScore.

G.3 Full metric suite

Elsewhere the search verifier also evaluates each strategy. Here we score the delivered videos with a standard suite on a held-out VBench subset drawn from the validated population of Section G.1: six VBench dimensions[1], VideoScore, ImageReward, mean RAFT optical-flow magnitude, and LPIPS against the same-seed full reference. The strategies deliver a single, best-of-8, keep, or commit video;

[1]Computed in `custom_input` mode, which applies every dimension to every video (the VBench protocol scores temporal flickering only on static-scene prompts), so these columns compare strategies on a fixed prompt set and are not leaderboard-comparable absolutes.

Table 14: Four delivery strategies re-scored under a standard metric suite on a held-out VBench subset ($\tau = 0.10$, $N = 8$). IR: ImageReward; VS: VideoScore average; flow: mean RAFT optical flow (px); cost: % of best-of-8. Bold: best per column; †: commit statistically consistent with best-of-8.

	Quality		VBench dimensions						Fidelity		Cost (%)	
strategy	IR↑	VS↑	subj↑	bg↑	mot↑	flk↑	aes↑	img↑	flow	LPIPS↓	wall↓	FLOPs↓
single (no search)	−0.06	2.58	0.970	0.975	**0.992**	**0.990**	0.554	65.3	1.85	0	**12.5**	**12.5**
best-of-8 full	**+0.81**	2.60	**0.978**	**0.980**	0.991	0.987	**0.615**	**67.7**	1.56	0	100	100
CachedSearch-keep	+0.79	**2.67**	0.976	0.978	0.991	0.987	0.604	66.7	1.96	0.114	51.4	48.5
CachedSearch-commit	+0.75	2.60†	0.976†	0.979	0.991†	0.987†	0.610†	67.3†	1.97	0†	63.9	61.0

we score the original rollouts without regeneration, and costs follow Section 4.1. Table 14 gives the means.

Commit is cheaper and no worse on metrics it never optimized. Its paired deltas from best-of-8 are statistically zero across the quality suite, and ImageReward concedes 0.06 of the 0.87 gain, a 93.2% capture consistent with Table 3. The rollout costs do not depend on the scorer: commit spends 64% of the wall-clock and 61% of the transformer FLOPs of best-of-8, while keep roughly halves the cost but pays a small frame-quality price that the reference-free scorers reward. Keep beats best-of-8 on VideoScore by +0.065, including +0.103 on dynamics, and beats its own full re-render by +0.036 on the frame verifier, matching the keep-minus-commit gap of Section G.4. Yet keep has LPIPS 0.114, aesthetic −0.012, and imaging −1.0. Its temporal signature is mild at this τ: flicker improves by +0.0005, while the −0.7% winner-level flow change is within the noise of the gate-grid winner flow loss (Section F.1).

G.4 Verifier preference for cached outputs

For the same winning seed, ImageReward (Xu et al., 2023) scores the cached video above its full-compute counterpart on average. The keep minus commit gap is +0.008, +0.037, and +0.028 at $\tau = 0.05$, 0.10, and 0.20, respectively ($n = 50$ pairs each; keep ≥ commit on ∼60% of pairs). VideoScore (He et al., 2024) shows a stronger preference across the same pairs, favoring keep on every dimension at every threshold. Its mean margin grows from +0.036 to +0.058 to +0.129 (Wilcoxon $p \leq 0.004$ each). At $\tau = 0.20$, the dynamic-degree margin is +0.176, with keep preferred on 92% of pairs ($p < 0.001$), despite reduced motion (Section F.1). The reference-free scorers in Section G.3 also rank keep above best-of-8.

Motion dampening is a plausible explanation. Cached videos move less and flicker less in these measurements (Section F.1). A frame verifier sees only 8 stills; slower motion may reduce blur and transient deformation and produce more canonical poses, potentially raising aesthetic and alignment scores. The keep–commit LPIPS (Zhang et al., 2018) differences show that the scores compare distinct outputs, but do not identify the cause of the preference. Both judges favor cached videos in these tests; this does not establish a general bias across learned verifiers.

For search, a candidate-independent additive score shift leaves the within-prompt $\arg\max$ unchanged. Candidate-specific changes can still reorder seeds (Section 4.2), but commit regret and capture use full-compute scores and already include these selection errors. By contrast, keep is evaluated on cached outputs (Table 3): its ∼99% nominal capture is verifier-based, not a guarantee of motion quality, and may reflect verifier over-optimization (Ma et al., 2025). Motion-aware or video-native verifiers (Xu et al., 2024; He et al., 2024) may help, but VideoScore's preference shows that a video-native judge alone is not a sufficient safeguard here. commit avoids delivering caching artifacts, but does not remove selection effects. Deployments using keep should report paired keep–commit score differences, mean optical flow, and LPIPS (Zhang et al., 2018); this check scores 2×50 videos per threshold.

H Models, scale, and threshold calibration

Section 4.4 summarizes transfer across backbones. Here we report six models at shared $\tau = 0.10$, their capture–speedup frontiers, per-model threshold calibration, and the effects of scale on cacheability.

H.1 Six backbones at a fixed threshold

We evaluate Wan2.1-T2V-1.3B (Wan Team, 2025a), Wan2.1-T2V-14B (Wan Team, 2025a), Wan2.2-TI2V-5B (Wan Team, 2025b;a), CogVideoX-5B (Yang et al., 2024), HunyuanVideo-13B (Kong et al., 2024), and LTX-Video-2B (HaCohen et al., 2024) under the full protocol, each at its native configuration (Table 7) and scored by ImageReward (Xu et al., 2023) (Table 15). CogVideoX batches both guidance branches into one transformer call, so it uses the single-branch cache wrapper rather than the per-branch state of Appendix C. The results show a stronger family pattern than a size trend. Across Wan2.1's 10.8× parameter gap, median ρ stays unchanged, with modest changes in capture and top-1 agreement; Wan2.2-5B also retains high rank correlation and capture. All three non-Wan models have lower median ρ and capture than the Wan models; LTX-Video skips the most steps and scores lowest on both. At $\tau = 0.10$, candidate savings are 49–54% of C_f, except for LTX-Video at 62%. Absolute savings therefore grow with rollout cost (Figure 21): cached exploration saves the most compute where full search is most expensive.

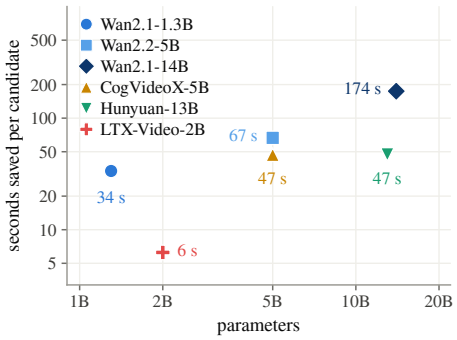


Figure 21: Seconds saved per candidate, $C_f - C_c$, against parameter count for the six models of Table 15 ($\tau = 0.10$, 50 prompts, 8 seeds).

Table 15: Six models from four families at fixed $\tau = 0.10$, each with 50 prompts and 8 seeds under its model-card recipe. Capture averages per-prompt gain ratios; *skip* is the fraction of reused denoising steps. Bold and underline mark the best and second-best fidelity values. See Table 4 for the main-text comparison.

model	params	med. ρ ↑	p10 ↑	top-1 ↑	capture ↑	skip	speedup ↑
Wan2.1-1.3B	1.3B	**0.905**	**0.61**	**64%**	**90.1%**	51%	1.97×
LTX-Video-2B	2B	0.536	0.14	50%	67.6%	64%	2.63×
Wan2.2-TI2V-5B	5B	<u>0.881</u>	<u>0.60</u>	46%	86.0%	55%	2.05×
CogVideoX-5B	5B	0.762	0.35	48%	75.2%	53%	2.06×
HunyuanVideo-13B	13B	0.762	0.44	54%	79.9%	57%	2.19×
Wan2.1-14B	14B	**0.905**	0.52	<u>58%</u>	<u>87.5%</u>	52%	2.05×

H.2 Per-model frontiers and threshold calibration

Within one backbone, the amount of reuse alone predicts fidelity. Figure 6(a) fits the Wan2.1-T2V-1.3B (Wan Team, 2025a) frontier on six points that vary either the threshold or the caching family (PAB (Zhao et al., 2024), CFG-Cache (Lv et al., 2024), TeaCache (Liu et al., 2024)): capture(%) $= 104.2 - 19.1 \ln(\text{speedup})$, $R^2 = 0.92$. Four configurations the fit never saw, the 25- and 100-step schedules, 720p, and an independent rerun of the default (not drawn in Figure 6(a), where it would sit on the default), land within 2.0 points (held-out $R^2 = 0.90$), and refitting all ten points barely moves the curve ($104.2 - 19.9 \ln x$, $R^2 = 0.94$). Candidate speedup therefore predicts selection fidelity across thresholds, caching families, schedules, and resolutions. The mechanism that produces the reuse does not create a new trade-off; each method moves the operating point along the same curve.

This relation fails across backbones. At $\tau = 0.10$ the Wan frontier over-predicts the other models' captures by 3–18 points ($R^2 = -0.86$), and a pooled fit over all 18 non-degenerate points reaches only $R^2 = 0.59$. Skip fraction behaves the same way without latency: it explains $R^2 = 0.89$ within Wan-1.3B but 0.40 across models. The threshold sweeps in Figure 6(b) are all monotone and smooth but shifted, so the same nominal threshold buys different effective aggressiveness on different backbones. CogVideoX's (Yang et al., 2024) $\tau = 0.10$ point lies well below Wan-1.3B's most aggressive setting, while its $\tau = 0.05$ point climbs back to the edge of the Wan band; its sweep follows $114.8 - 52.2 \ln x$ ($R^2 = 0.98$), 2.7× the slope of the Wan frontier. The other sweeps have slopes of −12.5 on Wan-1.3B, −16.1 on Wan2.2-5B (Wan Team, 2025b), −16.7 on HunyuanVideo (Kong et al., 2024), and −28.3 on LTX-Video (HaCohen et al., 2024), whose fit $94.5 - 28.3 \ln x$ ($R^2 = 0.99$) reaches 85% capture only near 1.4×. Wan2.1-14B, the most expensive model in the set, is the most caching-tolerant. Its sweep spans the full dial: at $\tau = 0.005$ the drift indicator never

Table 16: Per-model threshold calibration. The left block gives the capture of every measured τ; the right block gives the operating point at τ^*, the most aggressive measured threshold with at least 85% capture (bold in the left block). † marks LTX-Video, where no measured threshold qualifies and the row shows its best setting. Table 4 in the main text summarizes the operating points.

	capture (%) ↑ at measured τ =					operating point at τ^*				
model	.005	.02	.05	.10	.20	τ^*	med ρ ↑	capture ↑	skip	speedup ↑
Wan2.1-1.3B	-	-	93.6	90.1	**88.3**	0.20	0.857	88.3%	60%	2.41×
Wan2.2-TI2V-5B	-	-	89.3	**86.0**	83.1	0.10	0.881	86.0%	55%	2.05×
Wan2.1-14B	100	92.1	88.0	**87.5**	84.7	0.10	0.905	87.5%	52%	2.05×
CogVideoX-5B	-	-	**85.9**	75.2	64.7	0.05	0.810	85.9%	44%	1.78×
HunyuanVideo-13B	-	-	**85.1**	79.9	79.7	0.05	0.810	85.1%	46%	1.77×
LTX-Video-2B	-	79.6	71.6	67.6	-	none†	0.786	79.6%	43%	1.71×

crosses the threshold, so no step is skipped and the cached arm reproduces full compute bit-exactly (1.01×, 100% capture), an end-to-end determinism control like the no-caching row of Table 8. At $\tau = 0.02$, 0.05, and 0.20 it buys 1.21×, 1.59×, and 2.52×, and its four non-degenerate arms fit $93.3 - 9.2 \ln x$ ($R^2 = 0.93$), the flattest curve measured. We therefore report one frontier per family rather than a universal law: reuse sets fidelity within a family, while architecture sets the level and slope of the trade-off.

Because the operating point varies by model, we calibrate τ separately. Table 16 applies the rule from Section 4.4: choose the most aggressive measured threshold retaining at least 85% capture. Calibration uses one prompt set and both rollout modes (Appendix A). For a new family, start with a 25-prompt pilot: choose a target capture (the dotted line in Figure 6(b)), sweep τ to locate the crossing, and read off the corresponding speedup. All Wan backbones qualify at $\tau = 0.10$; the 5B and 14B models fall short at $\tau = 0.20$ by less than 2 points (83.1% and 84.7% capture). CogVideoX-5B and HunyuanVideo-13B need $\tau^* = 0.05$, which recovers most of their gap to Wan. LTX-Video-2B remains the boundary: tightening to $\tau = 0.02$ adds 12 capture points, but leaves it 5–6 points below CogVideoX and HunyuanVideo at similar skip fractions. Calibration helps, but LTX-Video has a lower measured frontier.

H.3 Which scaling axes change cacheability

CachedSearch depends on two properties of each (model, τ, verifier) triple: candidate speedup C_f/C_c and ranking fidelity r. We examine three hypotheses. *(H1) Cacheable redundancy grows with scale.* Larger models, higher resolution, or longer videos may introduce more mid-trajectory redundancy, allowing a fixed-τ cache to skip more. Forcing-KV (Ji et al., 2026) provides related evidence on the token-count axis: its KV-compression speedup rises from 1.35–1.5× at 480p to 2.82× at 1080p. *(H2) Ranking preservation does not follow redundancy.* Selection depends on cache-induced score noise relative to candidate spread, not parameter count alone. *(H3) Self-limiting corruption transfers across scale.* At 1.3B, spread and ranking fidelity are positively associated, and Proposition 3 relates aggregate capture to both. This association may persist at larger scales even when ρ falls.

H1 depends on the scaling axis. Schedule length has the clearest effect in our tests: at fixed τ, skip fraction follows a log-linear trend over the tested denoising lengths (Section F.2, Figure 18). Resolution changes it little (Table 11); Forcing-KV's resolution gains instead arise from autoregressive KV compression. Skip fraction also changes little across the two Wan2.1 sizes (Table 15). The results are consistent with H2, but architecture, training data, resolution, clip length, and CFG implementation confound cross-family comparisons. Wan2.1 retains median ρ across the 10.8× parameter gap, although this comparison does not isolate size causally. The threshold sweep largely recovers CogVideoX's fixed-threshold deficit (Table 16), supporting per-model calibration rather than a parameter-count rule.

Evidence for H3 is limited. On CogVideoX at uncalibrated $\tau = 0.10$, the spread–fidelity association remains positive but weaker (corr = +0.17). Mean ρ is 0.684, and 20/50 prompts fall below 0.7. Yet mean regret is 0.124 reward units versus 0.517 for random selection, retaining ∼76% of the gain as a ratio of means at roughly half the exploration cost. commit preserves the full-compute output for the selected seed. Median regret is 0.008, with exactly zero regret on 48% of prompts. A controlled

grid should vary parameter count, resolution, and clip length separately, calibrate τ per model, and measure skip fraction, iso-quality speedup, ranking fidelity, and the spread–fidelity association. The CogVideoX 2B$\leftrightarrow$5B comparison remains open.

I Extended related work

I.1 Training-free caching for video diffusion

Training-free caching accelerates video diffusion transformers by exploiting limited feature variation between adjacent denoising steps. Feature and attention differences are typically larger near the trajectory ends and smaller in the middle, a pattern observed in Open-Sora (Zheng et al., 2024), Latte (Ma et al., 2024), Wan 2.1 (Wan Team, 2025a), and HunyuanVideo (Kong et al., 2024). PAB (Zhao et al., 2024) broadcasts attention outputs across steps, with reuse ranges ordered cross $>$ temporal $>$ spatial. TeaCache (Liu et al., 2024) replaces uniform intervals with output-change estimates from timestep-embedding-modulated inputs. FasterCache (Lv et al., 2024) reduces feature loss from adjacent-step reuse; its CFG-Cache also exploits redundancy between guidance branches at the same timestep. EasyCache (Zhou et al., 2025) reuses per-step transformations while an accumulated change indicator stays below τ. Its reuse rule needs no offline calibration, and we adopt this wrapper. BWCache (Cui et al., 2025) gates reuse at DiT-block granularity using relative-L_1 similarity. FastCache learns a linear feature approximation, while NaviCache self-calibrates its schedule (Liu et al., 2025a; Lv et al., 2026). LeMiCa (Gao et al., 2025) formulates scheduling as a shortest-path problem with lexicographic minimax optimization to control accumulated error. ReCache (Aliev et al., 2026) instead learns budget-aware schedules with REINFORCE (Williams, 1992) against uncached targets. These approaches range from local reuse rules to global error control and learned schedules. Near-lossless training-free caching commonly targets 2–3$\times$ speedups, evaluated through *single-sample fidelity* to full compute using PSNR (Wang et al., 2004), SSIM (Wang et al., 2004), LPIPS (Zhang et al., 2018), or absolute VBench (Huang et al., 2024) scores. Such measurements do not establish whether candidate rankings survive. We instead measure ranking fidelity and the selection regret introduced by caching.

Guidance and perceptual scoring are complementary design axes. Rectified-CFG++ (Saini et al., 2025a) replaces CFG extrapolation with a predictor-corrector update designed to limit off-manifold drift in rectified flows. LGDM (Saini et al., 2025b) uses guidance-conditioned latent-diffusion features as a perceptual quality signal. These methods target sampling dynamics or scoring, whereas cached exploration takes the sampler and verifier as given.

I.2 Test-time scaling and diffusion search

A complementary literature spends more inference compute to improve quality. Ma et al. (2025) formulate verifier-guided search over sampling noise after showing that gains from additional denoising steps saturate. With modest search compute, a 0.6B model can outperform a 12B model without search. They also show that optimizing one verifier can improve its score while degrading held-out metrics, a failure mode we examine in Section G.1. Video-T1 (Liu et al., 2025b) uses multimodal rewards in Tree-of-Frames search. EvoSearch (He et al., 2025) applies evolutionary denoising to make Wan 1.3B competitive with Wan 14B. LatSearch (Zhao et al., 2026) scores partially denoised latents and prunes before VAE decoding, matching EvoSearch-level gains at lower cost. Diffusion latent beam search (Oshima et al., 2025) branches at each denoising step and scores beams through short deterministic DDIM lookaheads that estimate clean samples. It also calibrates a weighted combination of quality metrics against vision-language-model feedback. Stream-T1 (Tu et al., 2026) explores streaming generation with few-step chunks, while early-failure detection (Kwon et al., 2026) aborts candidates using cheap RGB previews. Temporal backtracking search (Jun et al., 2026) restarts full-quality generation from verified prefixes rather than resampling entire rollouts.

These methods change how search allocates compute through branching, pruning, lookahead, or restarts. We target the complementary cost of candidate generation through training-free caching. Caching can combine with pruning, as demonstrated by our latent-RM-style composition (Section 5), but the resulting savings and selection errors must be measured jointly. Our question is whether cheaper rollouts preserve the verifier ranking on which search depends.

I.3 Speculative draft-verify sampling

Our cheap-exploration/full-commit scheme resembles speculative execution. De Bortoli et al. (2025) extend speculative sampling to diffusion using acceptance rules for draft–target denoising pairs. A video system with a 1.3B drafter and 14B autoregressive target reports $1.59\times$ speedup at 98% quality retention through reward-based routing (Hu & Zhang, 2026). FlowCast (Bajpai et al., 2026) uses a velocity-MSE acceptance test for training-free self-speculative flow matching. For images, T-Stitch (Pan et al., 2024) assigns early and late sampling to small and large models; HybridStitch (Sun et al., 2026) extends this split across pixels and timesteps. These methods accelerate individual trajectories through draft acceptance or model switching. We instead use the same model under caching, retaining the seed, latent space, and sampling schedule without assuming identical trajectories. The verifier ranks completed cached candidates, and only the selected seed is regenerated at full compute. This guarantees that seed's full-compute output, not the same selection as full best-of-N.

I.4 Composition of efficiency techniques

Combining inference accelerators is not automatically safe. Video-BLADE (Gu et al., 2025) shows that directly combining pretrained sparse attention with a step-distilled model can degrade quality when distillation ignores the sparsity pattern; joint training addresses the mismatch. QuantCache (Wu et al., 2025) combines caching, quantization, and pruning through shared heuristics. TurboDiffusion (Zhang et al., 2025) reports 100–200$\times$ acceleration by jointly training sparse-linear attention, low-bit quantization, and consistency distillation. Sol (Li et al., 2026) tunes caching, sparsity, pruning, and quantization per model. These works address interactions among accelerators and their effects on generation quality. We study a different interaction: whether approximation preserves verifier-guided selection. CachedSearch combines cached candidate exploration with full-compute regeneration of the winning seed. Its ranking and regret audit measures when this preserves the full-search selection and how much full-compute verifier value is lost when it does not.

J Limitations

We use the frame-level ImageReward (Xu et al., 2023) verifier, which does not directly assess temporal quality. In our tests, VideoScore (He et al., 2024) preserves rankings less well and also favors motion-dampened cached outputs (Appendix G). We therefore default to commit; motion-sensitive uses still require direct temporal audits alongside keep–commit LPIPS (Zhang et al., 2018) comparisons (Section F.1). Our study covers six public checkpoints from four families and best-of-N seed selection up to $N = 32$. New architectures require threshold calibration. Searches that verify intermediate states also need separate validation, as in the preview-based pruning study of Section 5.