GOODBYE DRIFT: ANCHORED TREE SAMPLING FOR LONG-HORIZON VIDEO-TO-VIDEO GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon video generation suffers from two intertwined issues. First, there is drift, where video quality degrades over time. Second, there are continuity issues which manifest as object permanence issues, or improperly rendering transient content (e.g., an object that appears in non-consecutive frames changing color/style). Recent autoregressive distillation methods seek to improve both, but retain sequential rollouts that can accumulate errors over long horizons. We introduce **Anchored Tree Sampling (ATS)**: a training-free inference-time scheduler that addresses this accumulation by replacing left-to-right rollout with sparse-to-dense, anchor-bounded imputation organized as a tree. A root call produces sparse anchors over the full horizon, recursive refinement generates intermediate anchors, and final leaf spans are synthesized between neighboring anchors. For a horizon requiring K autoregressive chunks, ATS uses a balanced hierarchy with $O(\log K)$ sequential generation levels, replacing open-ended rollout with anchor-bounded infilling. We focus on V2V generation in the *static-camera* regime, where sparse anchors over the horizon are well approximated by the dense conditioning signal, and the base model can produce them without retraining. We evaluate ATS against two contemporary autoregressive baselines on Wan 2.1 + VACE, across five conditioning modalities (inpainting, outpainting, edge, pose, depth). ATS improves average frame quality and reduces mean within-window and cross-window quality variation in this evaluation. We additionally demonstrate stable ≥ 40-minute generation on LTX-2.3 across the same five modalities.

1 INTRODUCTION

Long-horizon video generation faces two related challenges. *Drift* manifests as errors which compound over time, degrading video quality. Second, *continuity* issues manifest as a lack of object permanence, or by improperly rendering spurious content (e.g., inconsistent structure/style of objects that enter/leave the frame at different points in time). These challenges stem from the fact that video generators have finite context windows, so when the requested output exceeds the window the default approach is to sample autoregressively. Autoregressive training and distillation methods (Yin et al., 2025; Huang et al., 2025; Cui et al., 2026b; Yang et al., 2025; Chen et al., 2024) seek to reduce drift and continuity failures during sequential generation. Despite these advances, autoregressive approaches still suffer from drift, or hard maximum duration constraints (Cui et al., 2026b) resulting in continuity issues. Autoregressive sampling also requires sequential rollout, forcing linear growth in wall-clock generation time, even for offline settings where all conditions are known.

Existing long-form video work concentrates overwhelmingly on text-to-video (T2V). Long-form *video-to-video* (V2V), where a per-frame conditioning signal $\boldsymbol{c}_{1:T}$ (mask, edge, pose, depth, etc.) is given over the full horizon and the model must regenerate content faithful to it, is by contrast largely underexplored at length. Bidirectional V2V conditioning adapters such as VACE (Jiang et al., 2025) assume short clips, and the causal-distilled V2V variants we are aware of (Fosdick, 2026) inherit the same drift-and-continuity problem as their T2V counterparts once pushed past their training horizon.

We therefore ask a key question: if the full conditioning $\boldsymbol{c}_{1:T}$ is known in advance, why insist on causal rollout at all? Our answer is **Anchored Tree Sampling (ATS)**, a training-free sampler that attacks drift and continuity issues directly by replacing left-to-right rollout with *sparse-to-dense bounded imputation*. It assumes only that the base model can generate an interior span while

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

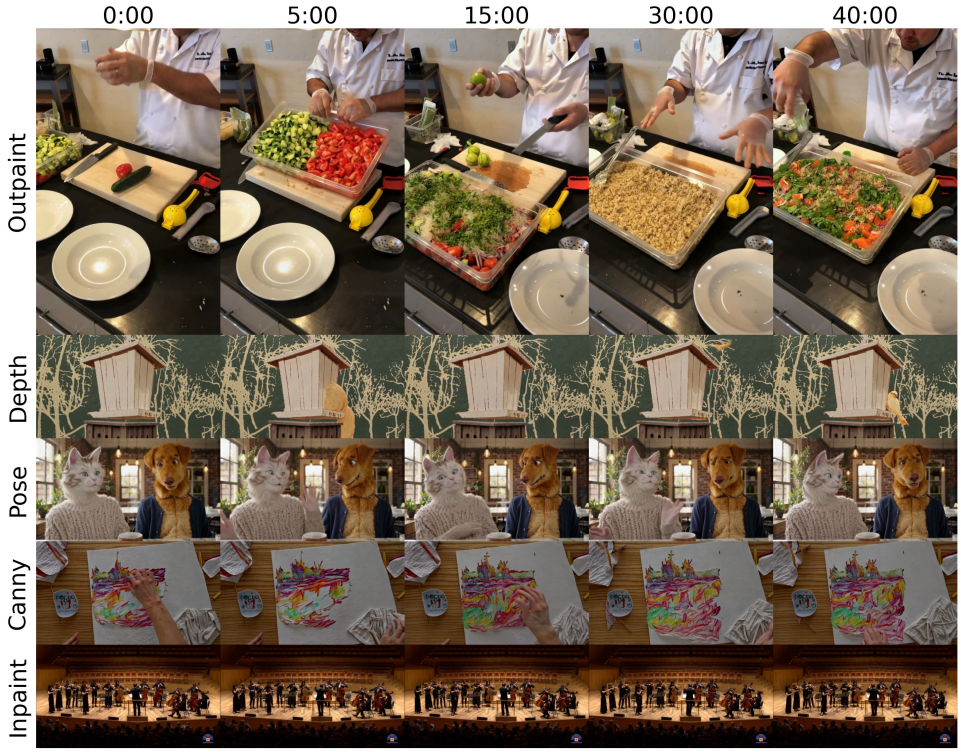


Figure 1: Long-horizon V2V with ATS on LTX-2.3 in the static-camera regime. ATS maintains identity, appearance, and scene state across the illustrated anchor-bracketed spans.

conditioning on information at both its start and end (we call these boundary conditions *anchors*) and organizes long-horizon generation hierarchically. First, we establish sparse anchors over the full horizon, recursively generate intermediate anchors if needed, then generate dense leaves between neighboring anchors. This distinguishes our method from AR training and distillation approaches that stabilize sequential rollouts but retain a fundamentally horizon-scaling dependency graph (Yin et al., 2025; Guo et al., 2025; Zhu et al., 2026; Huang et al., 2025; Chen et al., 2026; Zhao et al., 2026; Yang et al., 2025; Chen et al., 2024; Gu et al., 2025; Ji et al., 2025; Lu et al., 2025; Zeng et al., 2026), from token-level bidirectional decoding (Ghazvininejad et al., 2019; Stern et al., 2019; Gu et al., 2019; Chang et al., 2022; Savinov et al., 2021), and from training-time hierarchical or multi-scale generation that bakes the hierarchy into the model. ATS uses the base model's local-continuity prior and shared anchors; entities absent from the anchors can still vary across independently generated leaves.

We focus on the *static-camera* regime in this paper. The root call requires the base model to emit sparse output at chosen anchor times, but base video generators are trained on continuous footage. This makes producing sparse outputs a challenge for general video generation. In the static-camera examples studied here, much of the scene remains fixed, allowing the base model to generate useful sparse anchors without retraining. For dynamic-camera inputs paired with weak motion-control signals, multi-shot or multi-scene content, and pure long-form text-to-video, the same continuity bias overrides the sparse-output request. One could try to prompt the base model to produce cuts, but without post-training this is not reliable. We discuss a sparse-generation specialist and a semantic directed acyclic graph (DAG) as possible extensions to those regimes. The closest existing primitive is the keyframe generator of DCARL (Ouyang et al., 2026), which is the right kind of sparse model but is composed autoregressively with a dense generator. ATS's bidirectional, parallel-sibling tree is the natural replacement for that autoregressive composition.

Figure 1 shows ATS at work on LTX-2.3 (HaCohen et al., 2026) for $\geq$ 40-minute generations, covering each modality at both 720p and 1080p. These examples illustrate sustained visual quality over long durations. See Figure 6 in Appendix A.1 for a visual illustration of ATS.

2 BACKGROUND AND RELATED WORK

We organize this section around the question that motivates our method: when a base generator has only a bounded temporal window, what dependency structures can be used to compose it into long-form output? We first formalize the standard autoregressive (AR) answer and its failure modes, then describe the two-sided primitive that replaces it, and finally survey existing schedules that compose such a primitive over long horizons and connect them to broader trends in inference-time compute.

2.1 AUTOREGRESSIVE LONG-FORM GENERATION AND ITS FAILURE MODES

We assume that a V2V generative model produces a target signal $\boldsymbol{x}_{1:T}$ aligned with conditioning $\boldsymbol{c}_{1:T}$, where $\boldsymbol{c}_{1:T}$ represents some per-frame conditioning information. When T exceeds the model's native context $M_{\max}$, the standard solution is chunked AR sampling. The horizon is partitioned into chunks $B_1, \ldots, B_K$ with $|B_k| \leq M_{\max}$, and each chunk is generated according to

$$\widehat{\boldsymbol{x}}_{B_k} \;\sim\; p_\theta(\boldsymbol{x}_{B_k} \,|\, \boldsymbol{h}_{k-1},\, \boldsymbol{c}_{B_k})\,, \tag{1}$$

where $\boldsymbol{h}_{k-1}$ is some historical representation of previous chunks. Equation (1) is exact only if $\boldsymbol{h}_{k-1}$ is a sufficient statistic for the entire history $\boldsymbol{x}_{<B_k}$, which is rarely the case for nontrivial signals. The resulting approximation is responsible for the long-range failure modes that motivate this paper.

The first failure is *drift*, which manifests as quality degradation over time. The conditioning of chunk $k+1$ is itself a sample from Equation (1), so any local distribution shift at chunk k enters the suffix that conditions chunk $k+1$, which in turn enters the suffix conditioning chunk $k+2$, and so on. Thus, errors compound over time, leading directly to quality degradation. There is no mechanism in Equation (1) to suppress drift, because chunk k's output is conditioned only on its past with no future constraint to correct it. In V2V, the per-frame conditioning track $\boldsymbol{c}_{1:T}$ acts as a strong local anchor on the quality axis: each chunk is pinned to a structural reference (mask, edge, depth, pose) and remains locally faithful. The content axis is not anchored, instead identity and appearance live in $\boldsymbol{h}_{k-1}$ and still accumulate error as $\boldsymbol{h}_k$ grows.

The AR implementations evaluated here use bounded KV context and periodic hard resets. Within a reset window, generated history can accumulate quality errors; at a reset, discarding history can produce a content jump. These are properties of the evaluated rollout configuration, not requirements of AR generation. We also examine a no-reset position-shift control in Section 4. Training-free inference methods such as Deep Forcing (Yi et al., 2025), Infinity-RoPE (Yesiltepe et al., 2026), and LoL (Cui et al., 2026a) modify cache or positional-encoding behavior to improve long rollouts while retaining sequential generation.

AR methods also suffer from a sequential rollout requirement. This is a direct consequence of Equation (1) forcing $\widehat{\boldsymbol{x}}_{B_1}, \ldots, \widehat{\boldsymbol{x}}_{B_{k-1}}$ to be available before $\widehat{\boldsymbol{x}}_{B_k}$ can begin. Thus, the wall-clock time of a full rollout grows linearly in K regardless of how much hardware is available.

Recent methods improve sequential generation through distillation (Yin et al., 2025; Zhu et al., 2026), long-context or noise-conditioned training (Guo et al., 2025; Chen et al., 2024), and training on generated histories (Huang et al., 2025; Chen et al., 2026; Zhao et al., 2026; Yang et al., 2025). FramePack (Zhang et al., 2025) additionally explores endpoint-conditioned bidirectional sampling with a compressed history to reduce drift. Once the task is offline and the full conditioning $\boldsymbol{c}_{1:T}$ is known before sampling, there is no reason for the right-hand side of Equation (1) to omit future information.

2.2 TWO-SIDED GENERATORS AND BOUNDED INFILLING

A natural alternative to Equation (1) is a primitive that conditions on both past and future context simultaneously. Given anchors $\boldsymbol{A}_a$ and $\boldsymbol{A}_b$ at the start and end of an interval $[a, b]$, the two-sided primitive samples

$$\widehat{\boldsymbol{x}}_{a:b} \;\sim\; p_\theta(\boldsymbol{x}_{a:b} \,|\, \boldsymbol{A}_a,\, \boldsymbol{A}_b,\, \boldsymbol{c}_{a:b})\,. \tag{2}$$

Several pretrained video generators implement Equation (2) explicitly. LTX-2.3 (HaCohen et al., 2026) natively consumes left and right boundary frames together with a per-frame conditioning track (mask, edge, depth, pose, etc.). The unified V2V conditioning adapter VACE (Jiang et al.,

2025) provides the same primitive on top of other open-weight backbones. Learned video frame-interpolation methods (Reda et al., 2022; Huang et al., 2022; Danier et al., 2024) can be read as the limiting case of Equation (2) in which $\boldsymbol{A}_a$ and $\boldsymbol{A}_b$ collapse to single frames. The same primitive recurs in token-level parallel decoders (Ghazvininejad et al., 2019; Stern et al., 2019; Gu et al., 2019; Savinov et al., 2021; Chang et al., 2022).

We treat any generator that supports Equation (2) as a black-box primitive and ask how to compose many calls to it into a coherent long-form sample. This separation between the modeling problem and the scheduling problem makes our method training-free and modality-agnostic.

2.3 Hierarchical scheduling and inference-time compute

Composing Equation (2) over a long horizon requires a schedule for placing anchors and ordering the resulting calls. Several existing systems each commit to a particular hierarchy baked into model design or training, NUWA-XL's diffusion-over-diffusion keyframes, Phenaki's shared-token clip sequencing, Lumiere's space-time U-Net (Yin et al., 2023; Villegas et al., 2023; Bar-Tal et al., 2024), or extend fixed-window generation through overlap or attention fusion (Qiu et al., 2024; Wang et al., 2023; Henschel et al., 2025; Lu et al., 2024). We treat the hierarchy as an *inference-time* decision. The depth L, branching, and anchor representation are chosen at runtime as a function of T and $M_{\max}$, and no part of the base model is modified. Flexible Diffusion Modeling (Harvey et al., 2022) already supports inference-time hierarchical schedules without retraining. ATS applies recursive sparse-to-dense scheduling to existing V2V backbones through sparse conditioning construction. For V2V, two regimes follow naturally. In *source-conditioned editing* a dense source signal supplies local fidelity while the tree supplies global consistency (the natural setting for inpainting, outpainting, edge, depth, pose). In *reference-conditioned generation* long-range consistency must propagate through sparse anchor structure rather than through a full source trace (Yin et al., 2023; Villegas et al., 2023; Yang et al., 2025).

Recently, (Ouyang et al., 2026) proposed DCARL, a system which trains a dedicated sparse-keyframe generator alongside a dense generator and composes them in a divide-and-conquer pipeline. The keyframe generator emits a full keyframe track over the long horizon, after which the dense generator fills in the inter-keyframe spans. DCARL conditions dense generation on surrounding keyframes and preceding dense content, whereas ATS generates sibling spans without conditioning on previously generated sibling spans. We discuss DCARL further in Section B.2.1.

3 Anchored Tree Sampling

We now describe how the two-sided primitive of Equation (2) can be composed into a long-form sampler. The construction reorganizes long-horizon generation as a tree of bounded infilling problems. The resulting dependency graph is top-down rather than left-to-right.

3.1 A hierarchy of anchors

We retain the notation of Section 2.1, where a target $\boldsymbol{x}_{1:T}$ is generated with time-aligned conditioning $\boldsymbol{c}_{1:T}$, and the base model is treated as a black-box implementation of Equation (2) with leaf-span lengths constrained to $[M_{\min}, M_{\max}]$, the configured minimum and maximum numbers of frames per leaf span. An *anchor* $\boldsymbol{A}_t$ is any boundary representation the model can consume in place of $\boldsymbol{x}_t$ (a short clip or a keyframe). What matters is that anchors sit at known temporal positions, are consumable as start or end conditions in Equation (2), and carry enough local identity and dynamics information to keep the resulting infilling problem bounded.

The sampler organizes generation around a nested family of anchor sets

$$\mathcal{A}^{(0)} \subseteq \mathcal{A}^{(1)} \subseteq \cdots \subseteq \mathcal{A}^{(L-1)}, \qquad \mathcal{A}^{(\ell)} = \{\boldsymbol{A}_{t_0^{(\ell)}}, \boldsymbol{A}_{t_1^{(\ell)}}, \ldots, \boldsymbol{A}_{t_{N_\ell}^{(\ell)}}\}, \tag{3}$$

with $t_0^{(\ell)} = 1$ and $t_{N_\ell}^{(\ell)} = T$ at every level. The coarsest set $\mathcal{A}^{(0)}$ covers the full horizon, and each finer level inserts new anchors strictly between adjacent anchors of the previous level, so coarser anchors persist at all finer levels. The leaves of the tree are the dense intervals enclosed by adjacent anchors at the finest level $\mathcal{A}^{(L-1)}$, each of length in $[M_{\min}, M_{\max}]$. Depth L is determined entirely

by the relationship between T, the effective context span of the chosen sparse representation, and the dense leaf budget. The root call produces $\mathcal{A}^{(0)}$ to seed the tree, and $L = 1$ suffices when the resulting consecutive level-0 anchors are spaced within $[M_{\min}, M_{\max}]$ so leaf calls can fill them in directly. Deeper L is added recursively whenever any level-ℓ interval still exceeds $M_{\max}$, until every node call fits within the base model's context budget. The planner must also keep every packed sparse call within the model budget, including its anchor inputs.

3.2 Top-down factorization and parallel inference

Generation proceeds in three structurally distinct stages. The root stage produces the coarsest anchor set $\mathcal{A}^{(0)}$ in a single *conditioning-only* call to p_θ, emitting the level-0 anchors at the sparse times $t_0^{(0)}, \ldots, t_{N_0}^{(0)}$ from a sparse view of $\boldsymbol{c}_{1:T}$ alone (e.g. sparsely sampled source frames for source-conditioned video, or position-matched references for reference-conditioned video). We construct the root input by packing selected source-frame windows into a short clip and applying the task-specific conditioning operator. Conditioning on $\boldsymbol{c}_{a:b}$ denotes access to the source interval, not a requirement to pass every source frame in one call. Root and refinement calls require bounded sparse views; dense leaves use contiguous conditioning. For Wan, root and refinement share a four-frame-window packer; RoPE indices are local to the packed latent grid, with no global-time or FPS conditioning (Appendix G). In our source-conditioned experiments, we use a single keyframe per window; reference-conditioned settings instead retain the full decoded window as a multi-frame anchor clip so descendants inherit local motion context. Equation (2) accommodates either; the five modalities in Section 4 are all source-conditioned and use the keyframe variant. The refinement stage then generates each new anchor at level ℓ via an instance of Equation (2) whose start and end conditions are the enclosing pair of level-$(\ell{-}1)$ anchors. Writing $I_i^{(\ell)} = (t_i^{(\ell)}, t_{i+1}^{(\ell)})$, $\Delta_i^{(\ell)}$ for the level-ℓ anchors strictly inside the i-th level-$(\ell{-}1)$ interval, and $\widehat{\boldsymbol{x}}_j^{(L)}$ for the leaf-level dense content inside the j-th level-$(L{-}1)$ interval, the idealized sampler with independent node noise has joint

$$
\begin{aligned}
q_{\mathrm{ATS}}(\widehat{\boldsymbol{x}}_{1:T} \mid \boldsymbol{c}_{1:T}) \;:=\; & \underbrace{p_\theta\Big(\mathcal{A}^{(0)} \,\Big|\, \boldsymbol{c}_{1:T}\Big)}_{\text{root}} \cdot \prod_{\ell=1}^{L-1} \prod_{i=0}^{N_{\ell-1}-1} p_\theta\Big(\Delta_i^{(\ell)} \,\Big|\, \boldsymbol{A}_{t_i^{(\ell-1)}}, \boldsymbol{A}_{t_{i+1}^{(\ell-1)}}, \boldsymbol{c}_{I_i^{(\ell-1)}}\Big) \\
& \cdot \prod_{j=0}^{N_{L-1}-1} p_\theta\Big(\widehat{\boldsymbol{x}}_j^{(L)} \,\Big|\, \boldsymbol{A}_{t_j^{(L-1)}}, \boldsymbol{A}_{t_{j+1}^{(L-1)}}, \boldsymbol{c}_{I_j^{(L-1)}}\Big), \qquad (4)
\end{aligned}
$$

corresponding to root generation, anchor refinement, and dense leaf interiors. The interval index j and endpoint notation $a{:}b$ identify the same leaf. Our Wan implementation uses one checkpoint at every level and assembles decoded leaves by keeping the first leaf whole and dropping the first frame of each later leaf (Appendix G). Non-root calls use the two-sided primitive of Equation (2); the root uses conditioning alone. The factorization assumes independent node noise; the reported implementation reuses seed 40 per node, coupling the noise draws while preserving parallel execution (Appendix G). The execution order is root $\rightarrow$ refinement $\rightarrow$ leaves.

Algorithm 1 defines the ATS procedure. Three properties follow. No leaf extrapolates open-endedly, since every dense interval has both a left and right anchor; sibling generations are runtime-independent and execute concurrently; and the sampler never modifies the base model, so the construction is training-free. ATS thereby separates two roles entangled in AR rollout: coarse levels of Equation (4) carry global structure (aiding continuity) while leaves solve local bounded-infilling problems.

3.3 Compute, latency, and drift

The systems consequence of Equation (4) concerns latency rather than total compute. The tree organizes generation into $L + 1$ steps, featuring a root call followed by L subsequent levels (the last of which produces dense leaves). Each level of the tree consists of conditionally independent sibling calls, so under sufficient parallel hardware the critical-path latency reduces to

$$
\mathcal{T}_{\text{tree}} \;\approx\; (L+1)\,\tau_{\text{call}}, \qquad \mathcal{T}_{\text{AR}} \;\approx\; K\,\tau_{\text{call}}, \qquad (5)
$$

where τ_{call} assumes comparable call costs and $K = \lceil T/M_{\max} \rceil$ is the idealized non-overlapping AR chunk count; Appendix F treats unequal level costs and fixed worker counts. For a balanced tree with fixed branching and bounded per-call inputs, L scales logarithmically in T while K scales linearly.

Algorithm 1 Anchored Tree Sampling (ATS).

Require: horizon T, conditioning $\boldsymbol{c}_{1:T}$, base model p_θ implementing Equation (2), limits $(M_{\min}, M_{\max})$
Ensure: sample $\widehat{\boldsymbol{x}}_{1:T}$

Plan tree depth $L \geq 1$ and anchor times $\{t_i^{(\ell)}\}_{\ell,i}$ so every leaf-level interval has length in $[M_{\min}, M_{\max}]$
$\mathcal{A}^{(0)} \sim p_\theta(\cdot \mid \boldsymbol{c}_{1:T})$ at sparse times $t_0^{(0)}, \ldots, t_{N_0}^{(0)}$ ▷ bounded sparse input, no anchors
for $\ell = 1, \ldots, L$ **do** ▷ coarse-to-fine
  for all adjacent anchor pairs $(\boldsymbol{A}_a, \boldsymbol{A}_b) \in \mathcal{A}^{(\ell-1)}$ **in parallel do**
    $\widehat{\boldsymbol{x}}_{a:b}^{(\ell)} \sim p_\theta(\cdot \mid \boldsymbol{A}_a, \boldsymbol{A}_b, \boldsymbol{c}_{a:b})$ ▷ sparse if $\ell < L$; dense otherwise
  end for
  if $\ell < L$ **then**
    $\mathcal{A}^{(\ell)} \leftarrow \mathcal{A}^{(\ell-1)} \cup \{\text{anchors extracted at planned times from } \widehat{\boldsymbol{x}}_{a:b}^{(\ell)}\}$
  end if
end for
return $\widehat{\boldsymbol{x}}_{1:T}$ assembled from $\{\widehat{\boldsymbol{x}}_{a:b}^{(L)}\}$ in temporal order ▷ each output time assigned once

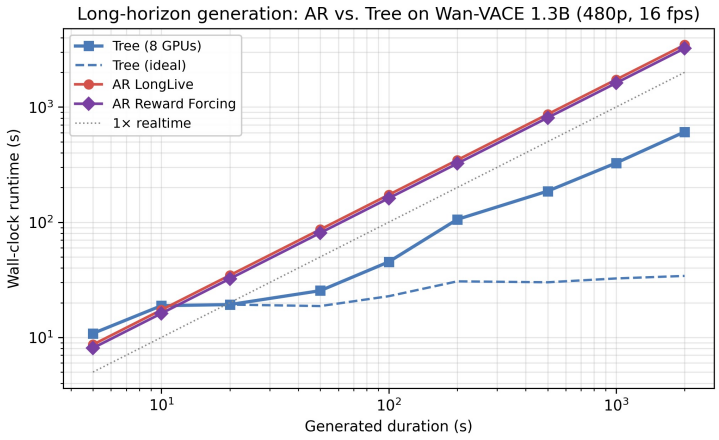


Figure 2: Runtime vs. generated duration on Wan 2.1 + VACE. ATS' parallelism allows it to achieve faster-than-realtime performance.

By construction, all non-root nodes produced by ATS are performing bounded infilling. This bounds the temporal span of each dense generation call, but does not guarantee identity or motion continuity. Poor anchors can propagate errors to their descendants. Note that a failed anchor is different than a single bad leaf, which can be regenerated without rerunning its siblings.

4 Experimental Results

Setup. We compare ATS, periodic-reset AR, and a no-reset position-shift control on the LongLive and Reward Forcing VACE adaptations (Yang et al., 2025; Lu et al., 2025; Fosdick, 2026; Wan et al., 2025; Jiang et al., 2025). Each ATS configuration uses one fused AR-distilled-base/VACE checkpoint for root, refinement, and leaves, with full bidirectional attention. Reward Forcing uses identical learned weights for AR and ATS. LongLive AR loads the released rank-256 LoRA, whereas ATS uses its SF/DMD base without that LoRA; this comparison is not weight-matched. All ATS levels use four denoising steps without CFG (Appendix G).

We evaluate five 30-minute source clips, one per modality (inpainting, outpainting, edge, pose, depth), at 16 fps with task-specific canvases, using seed 40 throughout. ATS uses $M_{\min} = 49$, $M_{\max} = 81$, and 360 leaves, each decoded to 81 frames; planned spans exclude shared endpoints (Appendix G). Sources and control extraction are detailed in Appendix E. We report full-frame



Figure 3: Pose-conditioned generation on the two Wan 2.1 + VACE checkpoint families. *Baseline* denotes the no-reset AR position-shift control; *AR* uses periodic resets.

Aesthetic Quality (AQ) and Imaging Quality (IQ), VBench-1.0 indicators (Huang et al., 2024). For inpainting/outpainting, we score the complete composite because generated content is viewed in the retained source context; both regions contribute. Global scores average 60 uniformly sampled keyframes. For either predictor M, within-window variation $|\Delta M|_c$ averages $|M(s_i) - M(e_i)|$ over ten nominal 192 s windows, sampling 5 s inside each edge. The last endpoint is clamped to the final available frame, giving a 65.25–67 s partial tenth comparison. Cross-window variation $|\Delta M|_r$ averages $|M(e_i) - M(s_{i+1})|$ over nine boundaries. All samplers use matched nominal timestamps; ATS and the no-reset control have no resets at these boundaries. These metrics measure frame-quality variation, not identity or motion continuity (Appendix F).

Global quality. Table 1 gives equal-weight five-source means for all six configurations. ATS improves AQ/IQ over periodic-reset AR by $3.25/5.44$ points for LongLive and $4.69/4.65$ for Reward Forcing, computed before rounding. The mean advantage is not uniform: LongLive ATS-minus-AR IQ ranges from -5.10 to $+16.48$ across sources (Appendix F).

Within- and cross-window variation. ATS has the best mean in all six metrics for each checkpoint family, including comparison with the no-reset control. Dropping the partial tenth window preserves every within-window ordering (Appendix F). Figure 5 shows a sustained mean IQ advantage across most sampled times, not monotonic degradation of AR quality. Lower window differences indicate steadier predicted quality and do not establish semantic continuity; per-window results and AQ trajectories are in Appendix F.

Qualitative reels. We assess identity and appearance continuity through the qualitative reels at matched nominal timestamps. Figure 3 shows pose-guided outputs across the 30-minute horizon, including no-reset controls that retain KV history through position shifts (Appendix G). The illustrated AR outputs change in character appearance and shading, while ATS retains these attributes more consistently across the shown samples. Figure 4 examines within-window changes (left pair) and a



Figure 4: Qualitative drift and cache-reset visualization for pose-guided Wan 2.1 + VACE generation. The left pair shows intra-chunk drift accumulated within a single cache window, while the right pair illustrates the cross-reset jump.

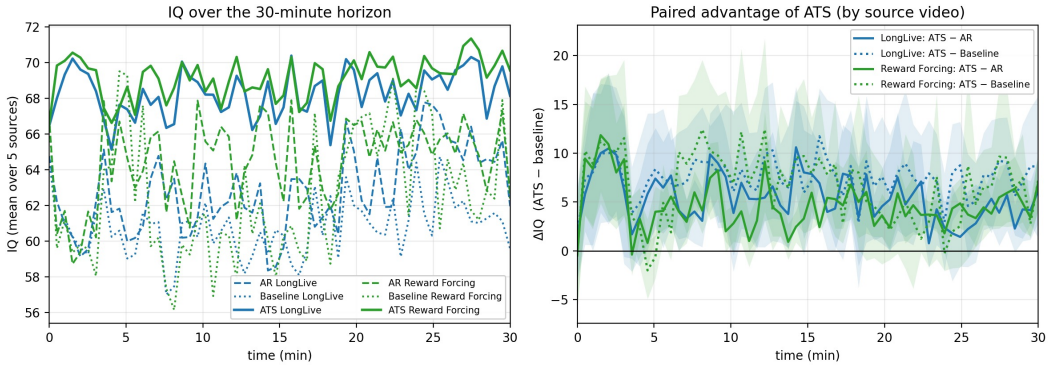


Figure 5: IQ trajectories for six configurations (left) and paired ATS-minus-control differences (right), averaged over five sources at 60 relative clip positions. AR uses periodic resets; Baseline uses no resets. Shading is $\pm\mathrm{SD}/\sqrt{5}$ across sources, shown descriptively; it is not uncertainty from independent trials.

reset boundary (right pair). LongLive exhibits both appearance drift and a reset jump in this example; Reward Forcing has smaller within-window changes but still changes appearance across the reset. Neither pattern is apparent in the illustrated ATS samples. These observations provide qualitative evidence for continuity in the shown scenes, separately from AQ/IQ; they do not imply that every leaf join or transient entity is consistent. The remaining four modalities appear in Appendix A.

Runtime. Figure 2 reports runtime vs. generated duration with Wan 2.1 + VACE on the two AR methods under test, ATS with 8 Nvidia H100 GPUs, and the theoretical-best (one-GPU-per-sibling) parallel configuration that ATS targets. AR has linear sequential depth in T, whereas ATS has logarithmic depth; realizing this latency scaling requires sufficient parallel hardware. These are

Table 1: Equal-weight means over five 30-minute sources. Each checkpoint family is evaluated with periodic-reset AR, a no-reset position-shift control (Baseline), and ATS. Global AQ/IQ average 60 keyframes; within-window and cross-window differences use ten windows (the last partial) and nine matched boundaries. Lower differences indicate steadier predicted frame quality, not identity continuity. Protocol and per-source results are in Appendix F.

		Global ↑		Within-window ↓		Cross-window ↓	
Checkpoint	Sampler	AQ	IQ	$\lvert\Delta\text{AQ}\rvert_c$	$\lvert\Delta\text{IQ}\rvert_c$	$\lvert\Delta\text{AQ}\rvert_r$	$\lvert\Delta\text{IQ}\rvert_r$
LongLive (Yang et al., 2025)	AR	55.91	62.85	3.86	5.92	4.01	6.28
	Baseline (no reset)	53.82	60.81	4.53	3.98	4.19	4.30
	ATS	**59.16**	**68.28**	**3.05**	**3.08**	**1.98**	**2.34**
Reward Forcing (Lu et al., 2025)	AR	53.61	64.50	3.60	3.46	3.57	4.71
	Baseline (no reset)	51.82	62.60	4.66	5.36	4.06	5.19
	ATS	**58.30**	**69.15**	**3.49**	**2.77**	**2.40**	**1.81**

elapsed-time comparisons, not evidence of lower total GPU time (Appendix F). With just 8 GPUs, ATS is 3.3x faster than realtime and 5.3x faster than the best AR method when generating a 2000s long video.

Comparison scope. We compare generation strategies within a shared offline V2V protocol: the same sources, prompts, and task-specific controls across samplers. Other training-free methods, including Deep Forcing (Yi et al., 2025), Infinity-RoPE (Yesiltepe et al., 2026), and FreeLong (Lu et al., 2024), are relevant, but comparison under this protocol requires integrating and validating their inference modifications with VACE conditioning. We do not undertake those adaptations here; our empirical claims are restricted to the evaluated configurations. Reward Forcing uses matched learned weights, and ATS improves both global predictors and all four variation metrics over each AR control. That comparison evaluates bidirectional endpoint conditioning together with tree scheduling, without isolating each component. LongLive uses different adaptations because its AR paths include a LoRA absent from ATS; their aggregate difference cannot be attributed solely to scheduling. The effect of adding that LoRA to the tree remains unmeasured.

Evidence across sources and time. The aggregate advantage survives exclusion of the final partial window, supporting the conclusion that it is not driven by that endpoint alone. The mean paired IQ advantage is positive at all 60 sampled positions for LongLive against either control, and at 58 and 56 positions for Reward Forcing against reset and no-reset AR, respectively. These counts describe the sampled trajectories, not independent repetitions. Variation across sources remains substantial: Reward Forcing ATS has lower global IQ than reset AR for outpainting despite its positive five-source mean. Full-frame AQ/IQ quantify the rendered video's predicted quality and its variation. Evidence for identity and appearance continuity comes from the qualitative reels, where recurring subjects and scene details can be compared directly.

5 Conclusion

We presented *Anchored Tree Sampling* (ATS), a training-free sampler for long-horizon offline V2V generation. ATS reorganizes long-form synthesis into one conditioning-only root call followed by a coarse-to-fine cascade of bidirectional refinement and dense leaf calls (Algorithm 1, Equation (4)), replacing the K sequential rollout steps of chunked AR with $L+1$ critical-path steps ($L = O(\log T)$), with sibling calls within a level executable in parallel across devices.

On five 30-minute videos with Wan 2.1 + VACE, ATS improves aggregate VBench quality (AQ, IQ) and reduces mean within-window and cross-window quality variation relative to the evaluated AR configurations. It is also faster: $5.3\times$ over the strongest AR baseline at the 2000 s horizon. Pushing further, ATS on LTX-2.3 produces V2V outputs exceeding 40 minutes across all five modalities, with sustained visual quality in the demonstrated static-camera examples.

Limitations. The study uses one source per modality and seed 40; neither sources nor sampled frames constitute independent trials. Continuity evidence is qualitative and comparisons cover the evaluated VACE configurations. Static-camera scope, anchor errors, motion seams, and transient entities limit generalization (Appendix B).

AI USE STATEMENT

In this work, we used large language models (LLMs) to assist with implementing experimental and evaluation code, providing feedback on research methodology and experiments, supporting qualitative analysis of generated videos, and analyzing and interpreting experimental results. Additionally, we used LLMs for manuscript drafting and editing, literature searches and reference checks, and critical feedback on the presentation of methods and results. We did not use LLMs to generate synthetic datasets, clean or reformat datasets, develop theoretical models or conceptual frameworks, propose or refine hypotheses, formulate mathematical claims, provide critical ingredients for proofs, assist with writing proofs, or perform translation.

Checks of AI-assisted work included cross-checking quantitative analyses against experimental outputs and run logs, recomputing reported metrics, and verifying citations against primary sources. The authors take responsibility for the final content of this work, including AI-assisted text, code, analyses, citations, and scientific claims.

REPRODUCIBILITY STATEMENT

Upon acceptance, we will release all code related to our Wan 2.1 + VACE experiments so that they can be reproduced.

REFERENCES

Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, Yuanzhen Li, Michael Rubinstein, Tomer Michaeli, Oliver Wang, Deqing Sun, Tali Dekel, and Inbar Mosseri. Lumiere: A space-time diffusion model for video generation, 2024. URL https://arxiv.org/abs/2401.12945.

John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986. doi: 10.1109/TPAMI.1986.4767851.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11315–11325, 2022.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

Shuo Chen, Cong Wei, Sun Sun, Ping Nie, Kai Zhou, Ge Zhang, Ming-Hsuan Yang, and Wenhu Chen. Context forcing: Consistent autoregressive video generation with long context. *arXiv preprint arXiv:2602.06028*, 2026.

Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. LoL: Longer than longer, scaling video generation to hour. *arXiv preprint arXiv:2601.16914*, 2026a.

Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. Self-forcing++: Towards minute-scale high-quality video generation. In *The Fourteenth International Conference on Learning Representations*, 2026b.

Duolikun Danier, Fan Zhang, and David Bull. Ldmvfi: Video frame interpolation with latent diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 1472–1480, 2024.

Ryan Fosdick. Adapting vace for real-time autoregressive video diffusion. *arXiv preprint arXiv:2602.14381*, 2026.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 6112–6121, 2019.

Jiatao Gu, Changhan Wang, and Junbo Zhao. Levenshtein transformer. *Advances in neural information processing systems*, 32, 2019.

Yuchao Gu, weijia Mao, and Mike Zheng Shou. Long-context autoregressive video modeling with next-frame prediction. *arXiv preprint arXiv:2503.19325*, 2025.

Yuwei Guo, Ceyuan Yang, Ziyan Yang, Zhibei Ma, Zhijie Lin, Zhenheng Yang, Dahua Lin, and Lu Jiang. Long context tuning for video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17281–17291, 2025.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, et al. Ltx-2: Efficient joint audio-visual foundation model. *arXiv preprint arXiv:2601.03233*, 2026.

William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Roberto Henschel, Levon Khachatryan, Hayk Poghosyan, Daniil Hayrapetyan, Vahram Tadevosyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Streamingt2v: Consistent, dynamic, and extendable long video generation from text. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 2568–2577, 2025.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *Advances in Neural Information Processing Systems*, 38:167283–167308, 2025.

Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *European conference on computer vision*, pp. 624–642. Springer, 2022.

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

Sihui Ji, Xi Chen, Shuai Yang, Xin Tao, Pengfei Wan, and Hengshuang Zhao. Memflow: Flowing adaptive memory for consistent and efficient long video narratives. *arXiv preprint arXiv:2512.14699*, 2025.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17191–17202, 2025.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Tao Liu, Leela Krishna, Gouti Pavan Kumar, Sreeja K, and Vishav Garg. V2V-Bench: A comprehensive benchmark for video-to-video generation evaluation. *arXiv preprint arXiv:2606.05665*, 2026.

Yu Lu, Yuanzhi Liang, Linchao Zhu, and Yi Yang. FreeLong: Training-free long video generation with SpectralBlend temporal attention. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Yunhong Lu, Yanhong Zeng, Haobo Li, Hao Ouyang, Qiuyu Wang, Ka Leong Cheng, Jiapeng Zhu, Hengyuan Cao, Zhipeng Zhang, Xing Zhu, et al. Reward forcing: Efficient streaming video generation with rewarded distribution matching distillation. *arXiv preprint arXiv:2512.04678*, 2025.

Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. In *International Conference on Learning Representations*, volume 2025, pp. 1045–1064, 2025.

Junyi Ouyang, Wenbin Teng, Gonglin Chen, Yajie Zhao, and Haiwei Chen. Dcarl: A divide-and-conquer framework for autoregressive long-trajectory video generation, 2026. URL https://arxiv.org/abs/2603.24835.

Haonan Qiu, Menghan Xia, Yong Zhang, Yingqing He, Xintao Wang, Ying Shan, and Ziwei Liu. Freenoise: Tuning-free longer video diffusion via noise rescheduling. In *International Conference on Learning Representations*, volume 2024, pp. 5260–5274, 2024.

Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. Film: Frame interpolation for large motion. In *European Conference on Computer Vision*, pp. 250–266. Springer, 2022.

Nikolay Savinov, Junyoung Chung, Mikolaj Binkowski, Erich Elsen, and Aaron van den Oord. Step-unrolled denoising autoencoders for text generation. *arXiv preprint arXiv:2112.06749*, 2021.

Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, et al. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026.

Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. Insertion transformer: Flexible sequence generation via insertion operations. In *International Conference on Machine Learning*, pp. 5976–5985. PMLR, 2019.

Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual descriptions. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vOEXS39nOF.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models, 2025. URL https://arxiv.org/abs/2503.20314.

Fu-Yun Wang, Wenshuo Chen, Guanglu Song, Han-Jia Ye, Yu Liu, and Hongsheng Li. Gen-l-video: Multi-text to long video generation via temporal co-denoising, 2023. URL https://arxiv.org/abs/2305.18264.

Weijia Wu, Mingyu Liu, Zeyu Zhu, Xi Xia, Haoen Feng, Wen Wang, Kevin Qinghong Lin, Chunhua Shen, and Mike Zheng Shou. Moviebench: A hierarchical movie level dataset for long video generation, 2025. URL https://arxiv.org/abs/2411.15262.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, et al. Longlive: Real-time interactive long video generation. *arXiv preprint arXiv:2509.22622*, 2025.

Zhendong Yang, Ailing Zeng, Chun Yuan, and Yu Li. Effective whole-body pose estimation with two-stages distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 4210–4220, 2023.

Hidir Yesiltepe, Tuna Han Salih Meral, Adil Kaan Akan, Kaan Oktay, and Pinar Yanardag. Infinity-RoPE: Action-controllable infinite video generation emerges from autoregressive self-rollout. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Jung Yi, Wooseok Jang, Paul Hyunbin Cho, Jisu Nam, Heeji Yoon, and Seungryong Kim. Deep Forcing: Training-free long video generation with deep sink and participative compression. *arXiv preprint arXiv:2512.05081*, 2025.

Shengming Yin, Chenfei Wu, Huan Yang, Jianfeng Wang, Xiaodong Wang, Minheng Ni, Zhengyuan Yang, Linjie Li, Shuguang Liu, Fan Yang, et al. Nuwa-xl: Diffusion over diffusion for extremely long video generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1309–1320, 2023.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22963–22974, 2025.

Ailing Zeng, Casper Yang, Chauncey Ge, Eddie Zhang, Garvey Xu, Gavin Lin, Gilbert Gu, Jeremy Pi, Leo Li, Mingyi Shi, et al. Lpm 1.0: Video-based character performance model. *arXiv preprint arXiv:2604.07823*, 2026.

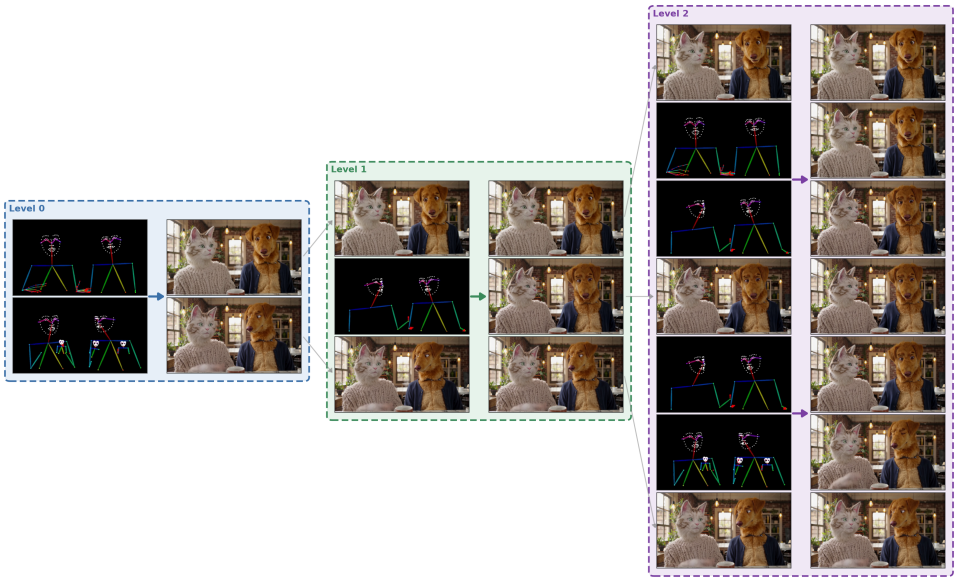
Lvmin Zhang, Shengqu Cai, Muyang Li, Gordon Wetzstein, and Maneesh Agrawala. Frame context packing and drift prevention in next-frame-prediction video diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=J8JCF64aEn.

Zengqun Zhao, Yanzuo Lu, Ziquan Liu, Jifei Song, Jiankang Deng, and Ioannis Patras. Relax forcing: Relaxed kv-memory for consistent long video generation. *arXiv preprint arXiv:2603.21366*, 2026.

Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation. *arXiv preprint arXiv:2602.02214*, 2026.

A ADDITIONAL FIGURES

A.1 TREE VISUALIZATION



(a) Tree structure. Three level chunks side-by-side, each showing the cells visible at that level (cond input on the left, gen output on the right of each chunk).



(b) Sparse-to-dense filling. Each chunk shows the full timeline twice: input row above (state before this level) and output row below (state after).

Figure 6: An overview of our proposed sampling method, ATS. A root call produces sparse anchors that span the entire horizon (a); subsequent guidance and leaf calls progressively fill in intermediate anchors and dense neighbour frames (b). For illustration, each leaf generates two new frames between its boundary anchors; these counts do not represent the model's generation limits.

The top panel shows the three-level call hierarchy: a root call generates sparse anchors that span the entire horizon, a guidance call inserts intermediate anchors between them, and leaf calls fill in the

remaining dense neighbour frames. The bottom panel shows the corresponding timeline view, where each level's generator only modifies the cells it is responsible for.

A.2 FULL-DURATION REELS

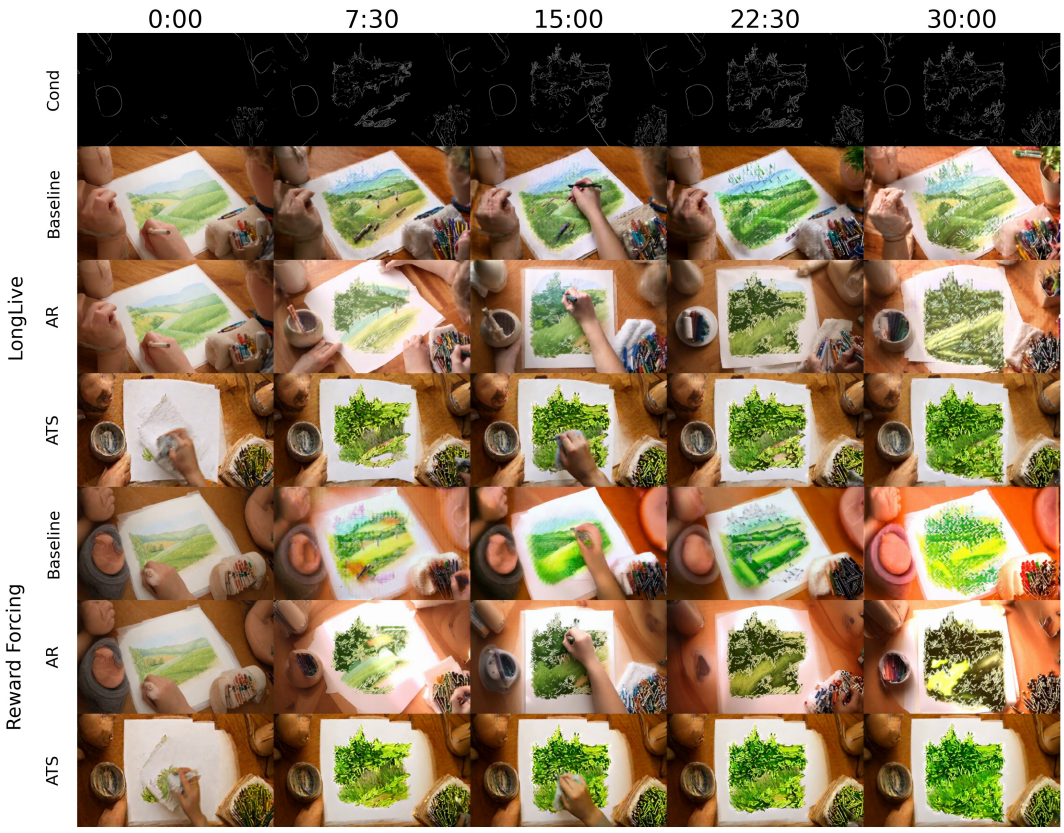


Figure 7: Long-form edge-conditioned generation on Wan 2.1 + VACE: AR vs. ATS on the two checkpoint families.

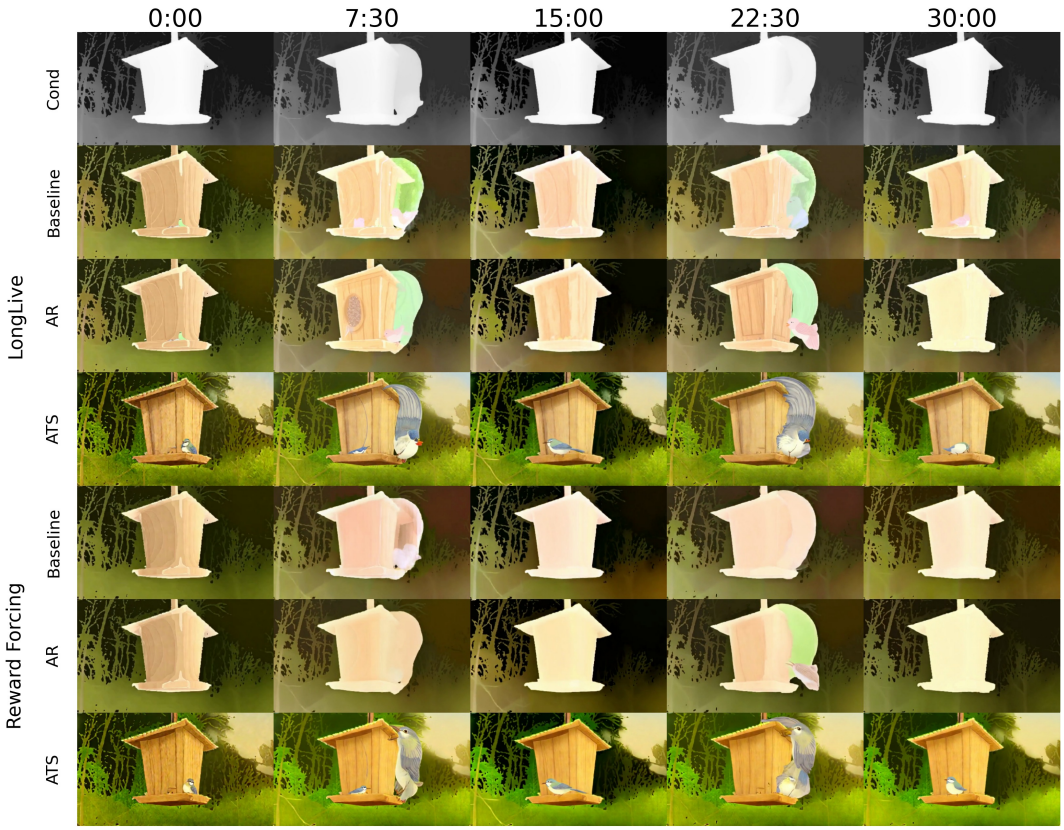


Figure 8: Long-form depth-conditioned generation on Wan 2.1 + VACE: AR vs. ATS on the two checkpoint families.

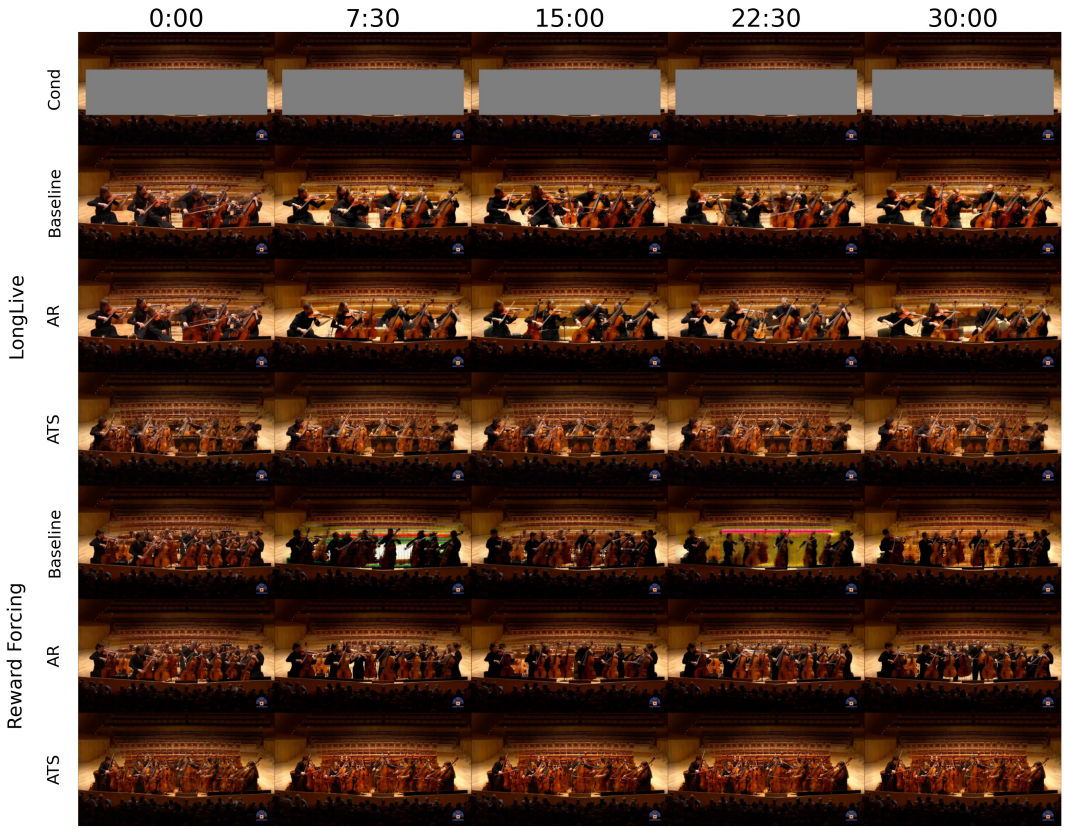


Figure 9: Long-form inpainting generation on Wan 2.1 + VACE: AR vs. ATS on the two checkpoint families.

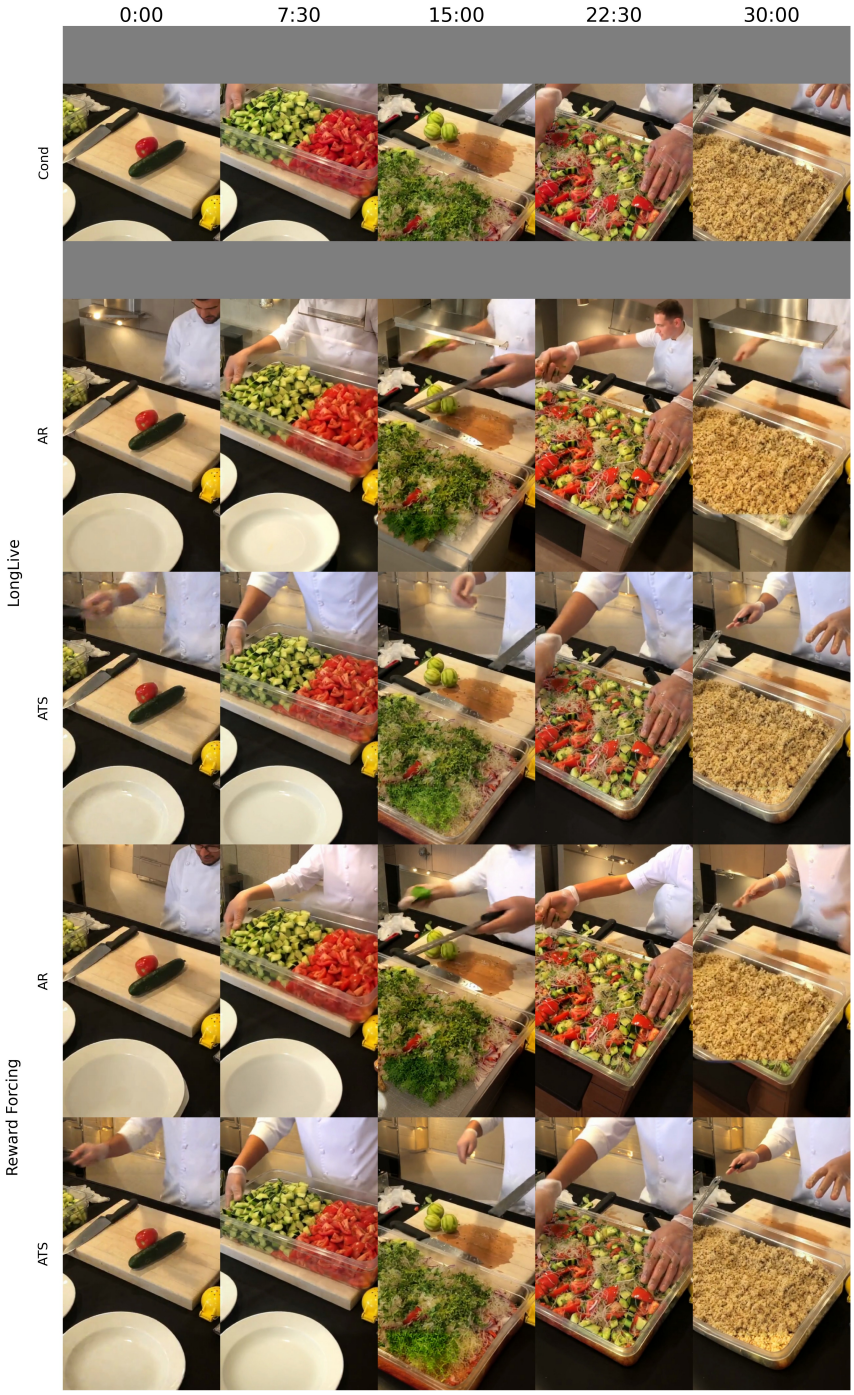


Figure 10: Long-form outpainting on Wan 2.1 + VACE: AR vs. ATS on the two checkpoint families.

A.3 DRIFT REELS

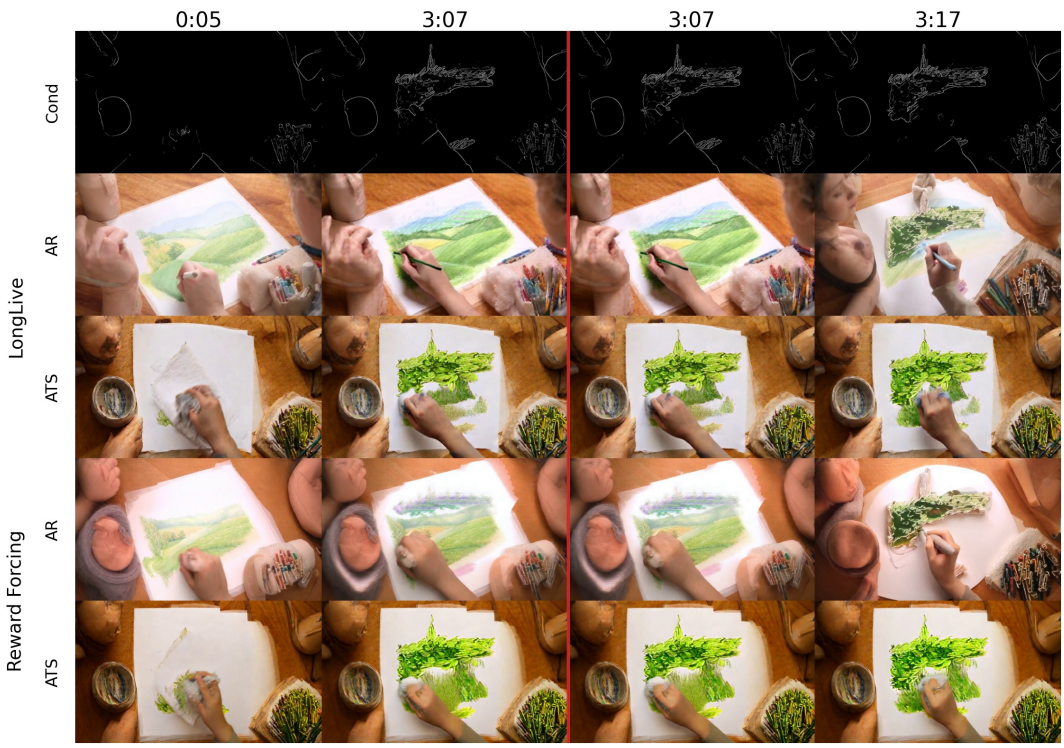


Figure 11: Qualitative drift and cache-reset visualization for edge-guided Wan 2.1 + VACE generation. The left pair shows intra-chunk drift accumulated within a single cache window, while the right pair illustrates the cross-reset jump.

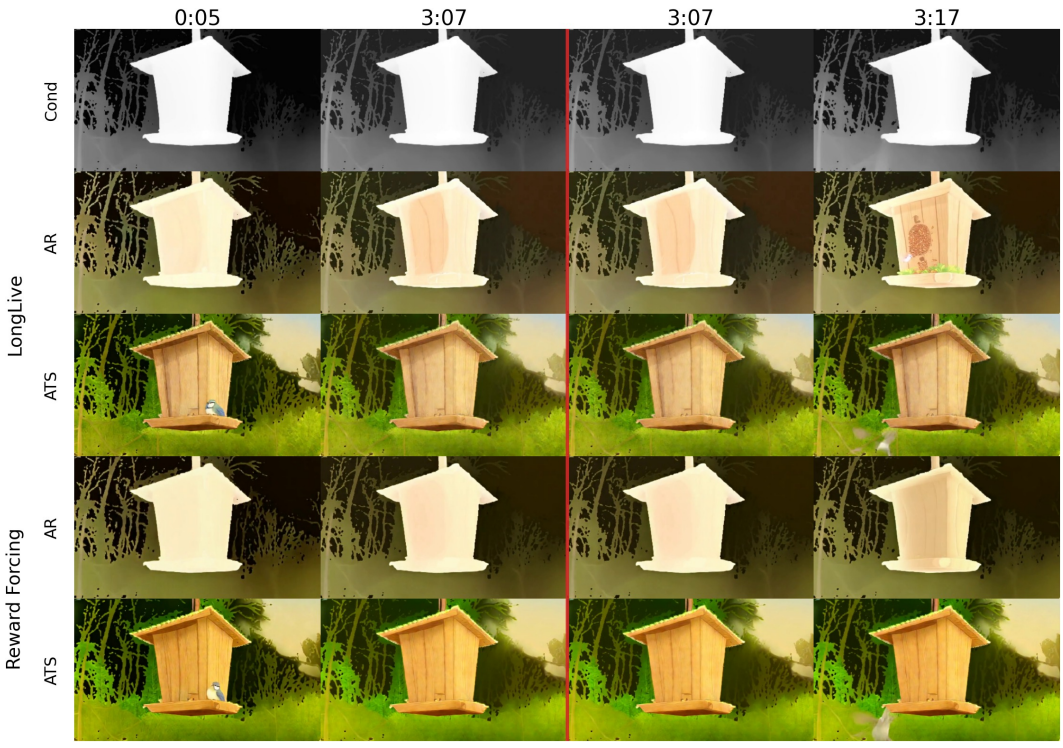


Figure 12: Qualitative drift and cache-reset visualization for depth-guided Wan 2.1 + VACE generation. The left pair shows intra-chunk drift accumulated within a single cache window, while the right pair illustrates the cross-reset jump.

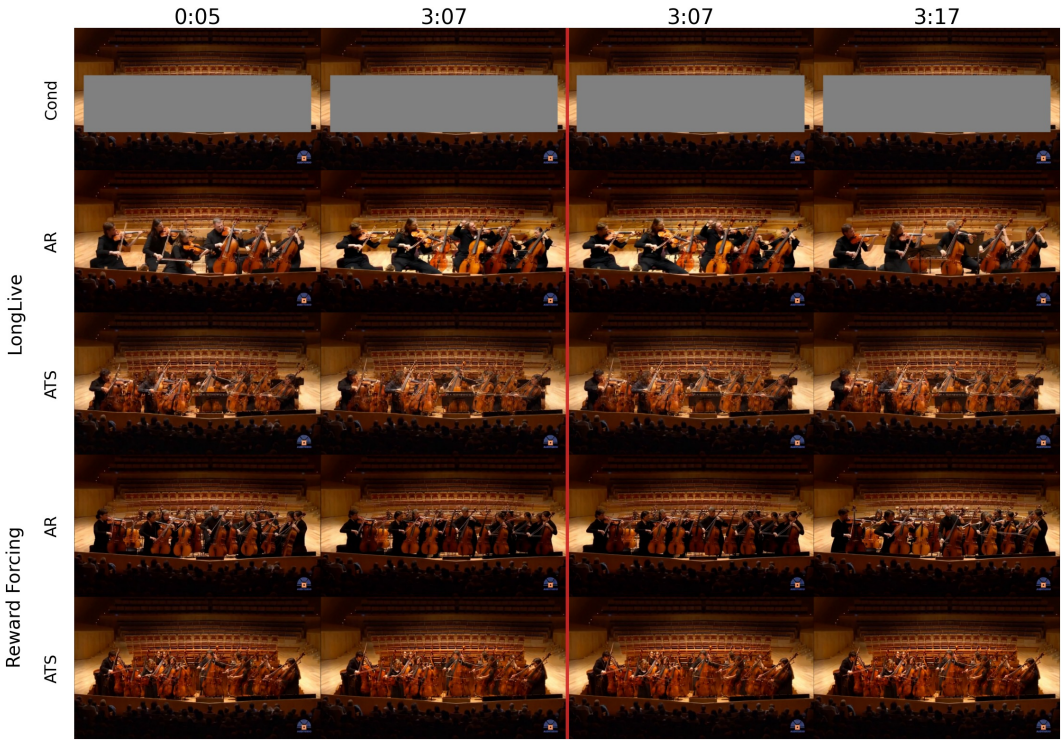


Figure 13: Qualitative drift and cache-reset visualization for Wan 2.1 + VACE inpainting. The left pair shows intra-chunk drift accumulated within a single cache window, while the right pair illustrates the cross-reset jump.

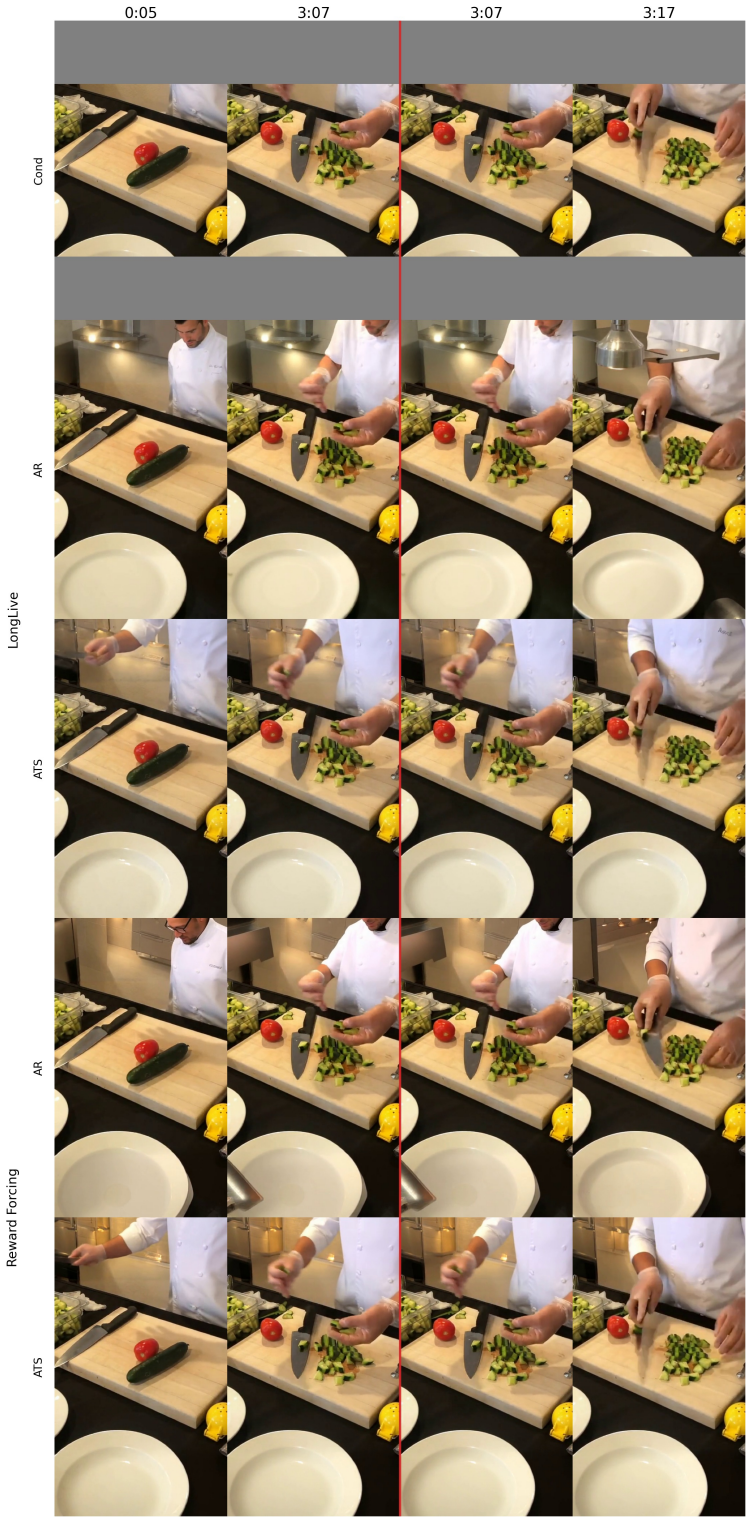


Figure 14: Qualitative drift and cache-reset visualization for Wan 2.1 + VACE outpainting. The left pair shows intra-chunk drift accumulated within a single cache window, while the right pair illustrates the cross-reset jump.

B LIMITATIONS OF ATS AND PROPOSED EXTENSIONS

B.1 LIMITATIONS

The five examples illustrate sustained long-duration quality, while also exposing motion and consistency failures. However, we observe a number of limitations with ATS, which limit its ability to be used broadly. Here, we discuss these limitations, then in Appendix B.2 we propose a concrete path forward to general-purpose long-content generation. We leave further development of the framework, including exploration of the proposed solutions, to future work.

1. A bad anchor poisons its entire subtree. Due to the hierarchical propagation of anchors with ATS, a single bad anchor results in degraded quality for all nodes which descend from it. Consequently, a poor root node result can seriously degrade an entire subtree. For static camera video, we have anecdotally observed this to not be a significant issue. However, when trying to apply ATS in its current form to dynamic camera video (or more complicated multi-shot video), generated guidance does not sufficiently adhere to the conditioning in weaker regimes (depth/Canny/pose).

2. Weak motion guidance demands video anchors, not keyframes. Unsurprisingly, when motion guidance is weak, keyframes are no longer sufficient to produce smooth motion continuation between chunks. In these instances, the anchors should consist of small chunks of video. The effects of this can be seen most prominently in the inpainting video featured in Figure 9. There, some chunks exhibit non-smooth motion continuation (e.g., unnatural movement given the previous chunk).

3. ATS does not necessarily solve consistency. ATS bounds dense generation between anchors, shortening the sequential paths along which rollout errors can accumulate. However, due to the independent rendering of all sibling nodes at each level of the tree, ATS does not guarantee consistency for entities outside the support of any bracketing anchor pair. Specifically, an entity that is out-of-frame in both bracketing anchors of a leaf but appears mid-leaf inherits no identity track from the tree, and the model is free to render it however the local conditioning permits. This can be seen in *all 5* generated videos. Each instance of an off-camera object or feature entering frame can result in it having a slightly different appearance (e.g., hands). This also leads to transient content, where spurious objects can enter frame, or vanish, within one subtree.

4. ATS does not provide a sparsity mechanism for non-V2V generation. Critically, ATS in its current form is not capable of generalizing to the more general class of non-V2V generation problems (e.g., T2V). The root and guidance nodes all leverage sparsity from a video conditioning signal. Without this, one cannot reliably generate a sparse representation of the desired video in order to leverage ATS-style inference.

B.1.1 MITIGATING THE LIMITATIONS OF ATS

The following untested mitigations are suggested by the structure of Equation (4), before the broader extensions in Appendix B.2.

1. *Adaptive anchor placement* replaces uniform level-0 spacing with placement biased toward regions of $\boldsymbol{c}_{1:T}$ with high local information content, shifting anchor density toward the regions of the horizon where transient events are most likely.
2. *Conditioning-residual gating* re-extracts the conditioning track from each leaf output and compares to the input track (e.g., re-extract Canny from the rendered leaf and compare to the input edge map). Large residuals flag intervals where the model has departed from $\boldsymbol{c}$, allowing one to re-render errant leaves or even entire corrupted subtrees.
3. *Cross-leaf consistency* adds a lightweight post-hoc pass that compares adjacent leaf outputs at their shared anchor and resolves residual discontinuities via a short bidirectional inpainting pass at the seams.

The initial generation can retain Equation (4); rejection, resampling, or seam repair would add operations and generally change the final output distribution. Their quality and cost require evaluation.

B.2 Potential extensions beyond static cameras

We propose three possible extensions; preserving parallelism depends on the additional conditioning and dependency structure. Limitations 1 and 4 both stem from the same fact: base video generators carry a continuity prior that resists sparse-anchor outputs, so the root call fails in dynamic and T2V regimes. A dedicated sparse-generation specialist alongside the dense generator (Section B.2.1) could address these failures. Limitation 3 (consistency) requires more: identity, environment, and object structure must be enforced across non-adjacent leaves, which is the natural fit for a directed acyclic graph (DAG) indexed by semantic rather than temporal subgraphs (Section B.2.2). The connecting thread is reference-based conditioning (Section B.2.3): anchors are already references in everything but name. Multi-frame anchors (Section 3.2) may help limitation 2 by supplying local motion context, but their effectiveness requires evaluation.

B.2.1 Applying ATS to T2V generation

Pretrained video models are trained on continuous video footage, so sparse outputs are out of distribution and the model cannot reliably produce them. This *continuity prior* is what makes generalizing ATS to T2V and non-static cameras non-trivial: the root call asks the model to emit content at sparsely chosen times with no continuous frames in between, but the prior insists on a continuous interpolant. One possible approach is a model trained to produce coherent content at specified levels of sparsity.

The natural sparsity ladder has three rungs. *Intra-shot* (dense): consecutive frames inside a single continuous take, where the continuity prior is exactly right. *Intra-scene* (semi-sparse): representative frames across multiple shots in a single setting and character configuration, where the prior is right at small scale and wrong at the cuts. *Intra-full-video* (very sparse): representative frames across the full horizon, spanning environment, identity, lighting, and shot changes. The evaluated dense generators are designed for the bottom rung; the full ladder is outside our demonstrated setting.

The closest existing primitive is the keyframe generator of DCARL (Ouyang et al., 2026), trained explicitly to emit a sparse keyframe sequence over a long horizon. DCARL generates a global keyframe track, then fills dense segments sequentially while conditioning on surrounding keyframes and preceding dense content. With ATS, the keyframe generator becomes the root call, and every inter-keyframe interval becomes a leaf or small subtree generated bidirectionally and in parallel. A suitable sparse generator could help limitations 1 and 4; ATS would contribute parallel sibling generation rather than dependence on preceding dense segments. DCARL itself currently assumes intra-shot sparsity (e.g., car-driving POV videos), so it does not demonstrate the full proposed ladder.

Even with a working keyframe generator, keyframes alone are unlikely to produce smooth motion continuity between neighboring chunks. We instead propose post-training controllable sparse *video* generators, triggered by prompt keywords and emitting fixed-duration sparse clips. This enables explicit modeling of different sparsity levels through text prompts; details are in Appendix C.

B.2.2 Semantic representation, not temporal

ATS's tree is a strictly temporal hierarchy: every node spans a contiguous time interval, and children partition the parent's interval. This structure alone does not express shared identity across scene cuts, character recurrences, or shared environments across shots—two non-adjacent leaves may share identity, environment, or object presence while two adjacent leaves share nothing. One candidate structure is a directed acyclic graph (DAG) whose nodes remain temporal spans but whose edges encode shared semantic structure rather than shared temporal nesting.

A *subgraph* is a set of nodes sharing a semantic constraint: an *appearance subgraph* containing every node in which entity X appears (regardless of when), a *shot subgraph* for every node within a continuous shot, a *scene subgraph* for every shot within a setting, and so on. Generation against such a DAG requires anchors that carry semantic identity, not just temporal-slot identity, so a leaf can be conditioned on its bracketing anchors *and* on the constraints active over its span—without those constraints needing to thread through a strictly temporal parent chain. The sparse-generation specialist of Section B.2.1 is the natural place to enforce these constraints, via structured prompts in the spirit of differential text-conditioning (e.g., "[ENVIRONMENT CHANGE: coffee shop $\rightarrow$ street]," "[CLOTHING CHANGE: shirt $\rightarrow$ dress]").

An appearance subgraph could supply references for entities missing from temporal anchors, but leaves must actually receive those references through a supported conditioning interface; otherwise the generator must be adapted. Sibling calls remain parallel only when all shared references are available beforehand. The critical path follows the DAG depth, which need not be logarithmic: a chain of sequential identity updates could grow linearly with the number of spans. Precomputing shared references and avoiding leaf-to-leaf dependencies is one possible way to retain parallelism.

B.2.3 Reference conditioning for continuity

The anchors that bracket every leaf in Equation (4) are, in effect, short references placed at chosen times: each pins character identity, environment, and object presence locally, and the leaf generator interpolates between bracketing pairs. References—explicit identity, environment, or appearance images and clips fed alongside the prompt—are a candidate mechanism for supporting the continuity dimensions of Section 1, because they sidestep the suffix-based memory of Equation (1) and replace it with a fixed conditioning term that does not drift.

The broader trend points the same way: recent systems are increasingly built around reference-based conditioning rather than pure prompt-based generation, with Seedance 2.0 (Seedance et al., 2026) illustrating this direction in T2V. Reference-based conditioning and ATS-style bidirectional composition are complementary, not competing—a model that accepts reference inputs serves as a stronger primitive for ATS to leverage in delivering consistency alongside coherence.

C TRAINING A SPARSE VIDEO GENERATOR

This appendix proposes a training strategy for a sparse-generation specialist intended to extend ATS beyond the static-camera regime (Section B.2.1). The specialist would generate both root and intermediate anchors; the dense base model would render within-shot leaves. This proposal has not been trained or evaluated, and is separate from the training-free experiments in this paper.

Why post-train rather than train from scratch? Pretrained video generators provide useful priors over appearance, motion, and short-range temporal structure. What they lack is the ability to emit *sparse* content on demand. We therefore propose a LoRA-based post-training framework on top of a strong base model such as LTX-2 (HaCohen et al., 2026). The goal is to teach the model to recognize a small vocabulary of *trigger tokens* in the prompt and to emit appropriately sparse output when those tokens appear. We would use fixed, model-compatible lengths for the sparse clips. Whether LoRA alone can learn discontinuous outputs while preserving identity and local motion is an empirical question; changes to positional encoding or attention may also be needed.

Trigger tokens for sparsity. The trigger tokens form a hierarchy of *how much carries over* between adjacent chunks of the sparse output. Four tokens cover the full range, plus one modifier:

- `[TIMESKIP]`: same shot, same setting, same subject; only time advances. Used for sparse temporal sampling within a single continuous take. This is the *intra-shot* level of the sparsity ladder in Section B.2.1.
- `[SHOT CHANGE]`: different shot, same scene. Camera angle, framing, and possibly subject focus shift; the setting persists. This is the *intra-scene* level.
- `[SCENE CHANGE]`: the same subject reappears in a different scene. Identity persists; location, time-of-day, and lighting change. The model is asked to render the same character recognizably in a new context.
- `[NEW SUBJECT]`: a fully new subject in a fully new scene. Nothing carries over visually. This is the *intra-full-video* level, and also the right token when the sparse generator opens a new narrative thread.
- `[STATIC CAMERA]`: a modifier (not a transition) indicating that the camera does not move during the segment. Provides explicit control over the regime where ATS already works today.

A multi-segment prompt is then a sequence of captions separated by transition tokens, of the form `[1]` ⟨`caption`$_1$⟩ `[TIMESKIP] [2]` ⟨`caption`$_2$⟩ `[SHOT CHANGE] [3]` ⟨`caption`$_3$⟩ $\cdots$. The token sequence specifies semantic transitions, not elapsed time or anchor spacing. A prompt with mostly `[TIMESKIP]` tokens asks for a long single take seen through a handful of representative frames, a prompt with mostly `[SCENE CHANGE]` tokens asks for representative frames spanning multiple scenes of the same character, and a mix yields a multi-scene, multi-shot, multi-subject sparse layout.

Grounding anchors in time. A planner would separately assign each output clip an explicit window of integer frame indices in $1{:}T$ before generation; at frame rate f, frame t corresponds to time $(t-1)/f$. These indices, clip lengths, and captions would form the per-node input, with timing represented as explicit numeric fields alongside the transition tokens. Generated clips would then occupy the planned windows; `[TIMESKIP]` or `[SCENE CHANGE]` would not determine their timestamps. The planner would refine gaps until every dense within-shot leaf has length in $[M_{\min}, M_{\max}]$. This enforces the numerical span budget, but does not guarantee that the learned generator respects the intended timing or semantic content.

Modeling sparsity through text rather than architecture. We propose using prompts to select semantic transitions, while the planner controls temporal spacing. The same specialist would be trained to produce sparse intra-shot content (`[TIMESKIP]`), intra-scene content (mixed `[TIMESKIP]` and `[SHOT CHANGE]`), or multi-scene content (`[SCENE CHANGE]` and `[NEW SUBJECT]`). The token mix would follow the planned content rather than a fixed mapping from tree level to token type:

even the root of a long single-shot video could use only [TIMESKIP] transitions. Dense leaves would use the base model without transition tokens.

This is also the natural place to host the differential text-conditioning constructs of Section B.2.2. Compound prompts of the form [SCENE CHANGE: coffee shop → street] or [SHOT CHANGE: trailing person → over-the-shoulder] attach the specific *delta* to the transition token, specifying the intended change and the attributes to preserve.

Training data: matching each rung of the ladder to a corpus. Each transition token corresponds to a different pairing structure in the training data, and the natural data source differs by rung:

- [TIMESKIP] pairs can be constructed by sampling short clips at separated times within a single shot. Large-scale clip datasets such as OpenVid-1M (Nan et al., 2025) provide candidate data for this rung.
- [SHOT CHANGE] pairs require within-scene multi-shot data: the same setting, a different camera, sometimes a different subject focus. Shot-segmented documentary and film footage covers this.
- [SCENE CHANGE] pairs are the hardest to source: the same subject must reappear across genuinely different scenes (location, time-of-day, lighting), with identity preserved. This is the rung where long-form narrative data becomes essential.
- [NEW SUBJECT] pairs are trivially abundant: any two unrelated clips form a valid pair.

The natural choice for the upper rungs of the ladder is MovieBench (Wu et al., 2025), a hierarchical movie-level dataset for long-video generation that explicitly indexes recurring characters across scenes within feature-length narratives. MovieBench supplies precisely the same-subject / different-scene structure that [SCENE CHANGE] needs to learn, and the multi-scene structure of feature-length films naturally exposes the full sparsity ladder *within a single source*. Temporally related content (recurring characters, persistent environments, intra-scene shot variety) and temporally unrelated content (cuts between unrelated scenes) live side by side in the same corpus. OpenVid and MovieBench are candidate sources for these regimes, subject to validating the required temporal and identity annotations. Training examples would include their planned frame indices and clip lengths; refinement examples would additionally condition on retained parent anchors.

Composition with ATS. The proposed specialist would replace the base model in both the root and sparse refinement calls of Section 3.2. At the root it would generate the planned coarse anchors; at refinement levels it would insert new anchors conditioned on the enclosing parent anchors, retaining the latter. Each call would receive only its local captions, timing fields, and relevant shared references, with the planner limiting both packed-video length and text-token count to the respective model budgets. Thus the full narrative need not fit in one text-encoder input.

Shot and scene cuts would have explicit frame boundaries in the plan. Dense infilling would stay within continuous shots; cuts would be assembled as explicit transitions between shots rather than asking the dense model to interpolate continuously across them. Leaves could use an unmodified base model where its existing conditioning interface suffices; enforcing identities absent from both temporal anchors would additionally require the reference inputs discussed in Section B.2.3. Conditionally independent siblings could still run in parallel, but runtime would depend on the planned dependency depth and per-call costs; the latency and coherence of this proposed extension remain to be evaluated.

D LTX-2.3 FULL-DURATION REELS

We evaluated the same five videos considered in Section 4 for the same tasks with LTX-2.3 (HaCohen et al., 2026). This appendix collects the LTX-2.3 per-modality reels at their full ≥ 40-minute duration, complementing the hero figure (Figure 1). Where the body figures in Section 4 are the 30-minute Wan 2.1 + VACE head-to-head comparisons against the autoregressive baselines, the figures below show longer LTX-2.3 reels used to assess visual quality beyond the 30-minute quantitative evaluation.



Figure 15: Long-form video inpainting with ATS on LTX-2.3, full duration.

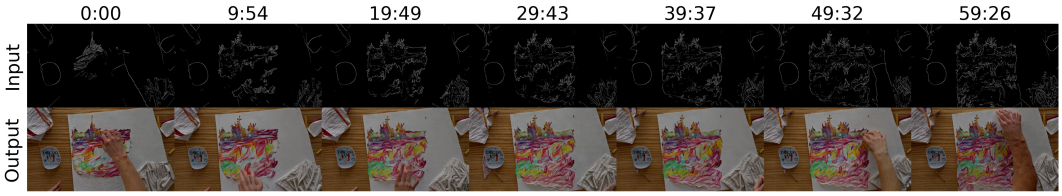


Figure 16: Long-form edge-conditioned (Canny) generation with ATS on LTX-2.3, full duration.



Figure 17: Long-form depth-conditioned generation with ATS on LTX-2.3, full duration.

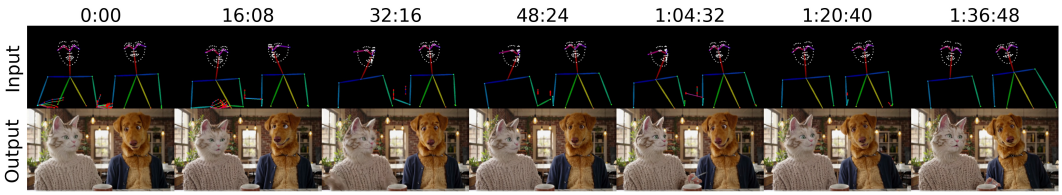


Figure 18: Long-form pose-conditioned generation with ATS on LTX-2.3, full duration.



Figure 19: Long-form video outpainting with ATS on LTX-2.3, full duration.

E SOURCE VIDEOS

The five source videos used for the qualitative reels (Section 4, Appendix D) and the head-to-head VBench comparison (Table 1) are listed below. Each clip is trimmed to a 30-minute window for the Wan 2.1 + VACE evaluation in the body, and to a $\geq$ 40-minute window for the LTX-2.3 reels in Appendix D. The table identifies the public source videos. Wan inputs use 16 fps. The documented temporal crops are $[0, 1800)$ s for depth and pose, $[40, 1840)$ s for Canny, and $[0, 1800)$ s of the pretrimmed sources for inpainting and outpainting. For the latter two, these are offsets within prepared clips, not original-video timestamps. Motion controls use DepthAnything-3 Metric-Large (Lin et al., 2025), OpenCV Canny (Canny, 1986), and DWPose body and hands with face disabled, rendered at 540p (Yang et al., 2023). Controls were extracted offline, before timing. No start image is supplied. Motion-control and inpainting mask canvases have width 832 and height 480; the outpainting mask canvas has width 480 and height 832. Table 3 gives the recorded mask geometry without assuming a shared canvas orientation. The inpainting source is stretched from 1920×1080 to its 832×480 canvas. Outpainting uses a 480×480 center crop of the actual 1280×720 cooking source, starting at $(x, y) = (400, 120)$, with 176-row padding above and below. Depth conditioning uses two-pass global normalization, with lighter foreground and darker background, and repeats the grayscale signal across RGB channels. Canny is extracted at 540p with recorded low/high settings of $0.1/0.4$ before RGB conversion and resizing to 832×480. Pose controls use body and hand skeletons, with face disabled, and a BGR-to-RGB color conversion before resizing to 832×480. The five positive prompts are listed in Appendix H. Sampling and packing settings are in Appendix G; upstream checkpoint/repository revision hashes were not recorded. We do not redistribute the source media.

Table 2: Source videos. One clip per VACE conditioning modality.

Modality	URL
Inpainting	`https://www.youtube.com/watch?v=4rgSzQwe5DQ`
Outpainting	`https://www.youtube.com/watch?v=wfYHoSzxk_o`
Edge (Canny)	`https://www.youtube.com/watch?v=e84sUd_QM6E`
Depth	`https://www.youtube.com/watch?v=61P9wlLyME8`
Pose	`https://www.youtube.com/watch?v=LeYIndII13w`

Table 3: Recorded mask geometry. Canvas sizes are height $\times$ width. Coordinates are pixel indices: (x, y, w, h) for boxes and (y, x, h, w) for placement.

Task	Canvas	Geometry
Inpainting	480×832	Mask box $(32, 160, 768, 192)$; original-source box $(80, 360, 1760, 420)$ on a 1080×1920 source.
Outpainting	832×480	Source placement $(176, 0, 480, 480)$; the remaining canvas is generated.
Motion controls	480×832	All-ones mask; the full frame is regenerated under control guidance.

F COMPUTE ACCOUNTING AND EVALUATION SCOPE

Fixed-worker latency and work. The critical-path approximation in Equation (5) assumes enough workers and comparable call costs. For P workers, n_ℓ calls at level ℓ, and representative call duration τ_ℓ, a level-synchronous schedule has the approximate generation latency and work

$$\mathcal{T}_P \approx \sum_{\ell=0}^{L} \left\lceil \frac{n_\ell}{P} \right\rceil \tau_\ell, \qquad W \approx \sum_{\ell=0}^{L} n_\ell \tau_\ell. \tag{6}$$

These expressions assume one GPU per call, similar durations within each level, and exclude orchestration overhead. Root and refinement work must be included, and their costs need not match those of distilled leaves. For a fixed dense-generation budget per output frame, logarithmic dependency depth does not imply logarithmic wall time at fixed P, or a reduction in total work. The reported runtime result is a latency comparison; total FLOPs or active GPU time cannot be inferred from the number of calls alone. Allocated GPU-hours for P reserved GPUs over t seconds are $Pt/3600$; this differs from active-device time when GPUs wait between levels.

F.1 SCORING PROTOCOL AND ROBUSTNESS

AQ uses L2-normalized CLIP ViT-L/14 embeddings with the LAION linear aesthetic prediction head, divided by 10. IQ uses MUSIQ-SPAQ on tensor-converted frames, divided by 100. Both are reported on a $\times 100$ scale and are full-frame single-image predictors; preserved pixels contribute to inpainting/outpainting scores. Global scores average 60 uniformly spaced keyframes independently within each output. Rescoring two generated videos with both scoring pipelines reproduced the stored metric fields exactly; this verifies scoring repeatability, not experimental replication.

Window endpoints are $s_i = (i-1)192 + 5$ and $e_i = i192 - 5$ seconds for $i = 1, \ldots, 10$. The nominal final endpoint $e_{10} = 1915\,\text{s}$ exceeds the horizon and is clamped to each output's last available frame. At 16 fps, ATS, periodic-reset AR, and no-reset AR have 28801, 28773, and 28800 frames, respectively. Their final sampled timestamps are 1800, 1798.25, and 1799.9375 s, so the tenth comparison spans 67, 65.25, and 66.9375 s from $s_{10} = 1733\,\text{s}$, versus 182 s in a complete window. All nine interior boundaries are complete; the remaining nominal timestamps are matched across samplers. Neither ATS nor the no-reset control has a cache reset at the matched window boundaries. We therefore use within-window and cross-window quality variation consistently for all samplers.

Table 4: Removing the final partial window changes no within-window ordering. All values are equal-weight five-source means.

		$\lvert\Delta\text{AQ}\rvert_c$		$\lvert\Delta\text{IQ}\rvert_c$	
Checkpoint	Sampler	10 windows	9 complete	10 windows	9 complete
LongLive	AR	3.86	3.52	5.92	6.04
	Baseline	4.53	4.35	3.98	4.16
	ATS	3.05	2.77	3.08	3.13
Reward Forcing	AR	3.60	3.43	3.46	3.48
	Baseline	4.66	4.65	5.36	5.59
	ATS	3.49	3.25	2.77	2.83

Table 5: Paired ATS-minus-control differences. Parentheses count the five sources favoring ATS, with lower variation preferred. Differences use unrounded scores. Counts are descriptive, not significance tests.

Checkpoint	Control	AQ	IQ	$\lvert\Delta\text{AQ}\rvert_c$	$\lvert\Delta\text{IQ}\rvert_c$	$\lvert\Delta\text{AQ}\rvert_r$	$\lvert\Delta\text{IQ}\rvert_r$
LongLive	AR	$+3.25$ (4)	$+5.44$ (4)	-0.81 (3)	-2.83 (5)	-2.03 (4)	-3.94 (5)
	Baseline	$+5.34$ (4)	$+7.48$ (3)	-1.48 (4)	-0.90 (2)	-2.20 (5)	-1.96 (3)
Reward Forcing	AR	$+4.69$ (4)	$+4.65$ (4)	-0.11 (4)	-0.69 (4)	-1.17 (4)	-2.89 (4)
	Baseline	$+6.48$ (5)	$+6.55$ (3)	-1.17 (4)	-2.59 (5)	-1.66 (4)	-3.38 (4)

Quality trajectories. The mean paired IQ gap favors ATS at all 60 sampled positions for LongLive against either AR control, at 58 of 60 for Reward Forcing against periodic-reset AR, and at 56 of 60 against its no-reset control. For AQ the corresponding counts are 55, 58, 58, and 59 of 60. These are descriptive counts of correlated sample positions, not independent replications or significance tests. The trajectories support a sustained mean quality-level advantage; they do not show monotonic AQ/IQ degradation for any sampler. Visual observations of content drift and reset jumps in the reels concern different properties from these scalar predictors. Since output lengths differ slightly, global trajectories compare equal relative positions within each clip on a nominal 30-minute time axis. The paired bands display across-source SD divided by $\sqrt{5}$; the five sources are not independent trials, so the bands are descriptive and do not quantify inferential uncertainty.

Evaluation scope. The five sources, one per modality, and their sampled frames/windows are not independent trials. All reported runs use seed 40, so multi-seed variation is unknown. The five-source means favor ATS in all six metrics for each checkpoint family, but individual-source advantages are not uniform. The predictors do not establish semantic identity preservation or motion continuity. The inpainting/outpainting task produces a composite in which generated content is contextualized by the retained source region, so full-frame scoring evaluates the complete rendered output. Identity and appearance continuity are assessed through the qualitative reels (Figure 3, Figure 4, and Appendix A); scalar quality scores do not replace that visual evidence. Source correspondence and structural adherence are additional evaluation dimensions, as emphasized by V2V-Bench (Liu et al., 2026). Root-checkpoint sensitivity was not evaluated here.

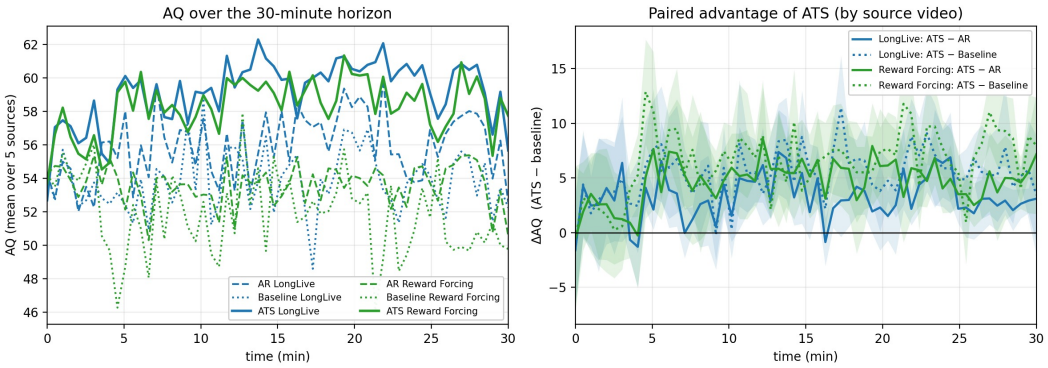


Figure 20: AQ trajectories and paired ATS-minus-control differences, using the same six configurations and descriptive band convention as Figure 5. Positive mean gaps occur at most, not all, sampled positions. Full-frame scores include preserved source pixels in editing tasks.

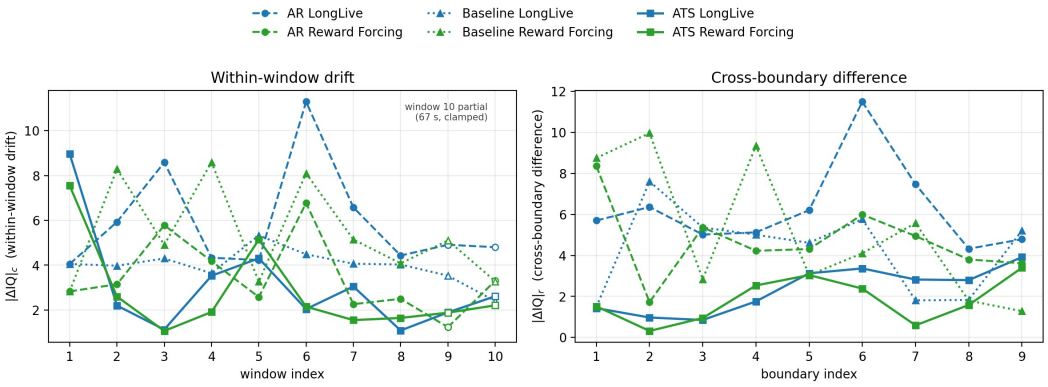


Figure 21: Within-window (left) and cross-window (right) IQ variation for six configurations, averaged over five sources. AR uses periodic resets; Baseline uses no resets. Only window 10 is partial; all nine interior boundaries are complete. Lines cross: the aggregate ATS advantage is not uniform at every window.

Table 6: Per-source LongLive configuration scores. Baseline denotes the no-reset control. All runs use seed 40; these are source-specific observations, not independent trials.

Source	Sampler	AQ	IQ	$\lvert\Delta\text{AQ}\rvert_c$	$\lvert\Delta\text{IQ}\rvert_c$	$\lvert\Delta\text{AQ}\rvert_r$	$\lvert\Delta\text{IQ}\rvert_r$
Inpainting	AR	51.07	61.98	2.34	6.97	2.18	7.20
	Baseline	47.59	62.61	3.62	4.58	2.38	3.69
	ATS	50.36	56.89	1.02	1.25	0.67	0.90
Outpainting	AR	56.79	66.95	4.41	5.83	2.79	5.23
	Baseline	57.20	68.10	3.41	4.36	3.41	3.09
	ATS	57.05	67.57	5.21	4.81	2.84	3.61
Edge	AR	48.55	57.31	3.18	9.58	4.68	10.55
	Baseline	49.03	47.26	4.32	4.67	3.30	8.47
	ATS	49.03	73.79	3.72	5.30	2.86	1.98
Depth	AR	51.17	51.95	7.00	4.20	8.24	5.18
	Baseline	46.12	53.64	8.40	2.03	8.79	2.85
	ATS	63.31	63.45	3.80	3.32	2.59	4.53
Pose	AR	71.98	76.04	2.40	3.00	2.17	3.22
	Baseline	69.14	72.42	2.88	4.26	3.05	3.38
	ATS	76.04	79.73	1.51	0.72	0.97	0.66

Table 7: Per-source Reward Forcing configuration scores. Baseline denotes the no-reset control. All runs use seed 40; these are source-specific observations, not independent trials.

Source	Sampler	AQ	IQ	$\lvert\Delta\text{AQ}\rvert_c$	$\lvert\Delta\text{IQ}\rvert_c$	$\lvert\Delta\text{AQ}\rvert_r$	$\lvert\Delta\text{IQ}\rvert_r$
Inpainting	AR	54.95	61.41	2.36	3.02	1.88	4.15
	Baseline	49.86	65.43	3.96	3.53	3.88	5.19
	ATS	57.19	65.15	1.32	2.10	0.68	0.89
Outpainting	AR	56.94	68.55	4.53	5.89	2.42	4.54
	Baseline	57.04	67.81	4.39	4.94	1.95	3.06
	ATS	57.46	67.19	3.67	3.89	3.64	2.39
Edge	AR	42.19	60.73	3.93	4.65	5.46	9.57
	Baseline	40.45	47.24	6.77	13.42	5.35	13.75
	ATS	40.92	72.68	3.40	4.16	1.73	1.05
Depth	AR	43.11	54.46	4.86	2.74	5.14	3.21
	Baseline	43.54	54.78	5.52	3.15	5.80	1.93
	ATS	61.58	61.82	7.08	2.61	3.56	3.70
Pose	AR	70.88	77.33	2.30	1.00	2.94	2.05
	Baseline	68.20	77.71	2.65	1.77	3.31	2.02
	ATS	74.37	78.91	1.96	1.10	2.37	1.04

G WAN IMPLEMENTATION DETAILS

Checkpoint and sampling. Each ATS run uses one fused AR-distilled-base/VACE checkpoint across root, refinement, and leaves, with no additional root model or LoRA. Reward Forcing AR and ATS share identical learned weights and use no LoRA. Both LongLive AR variants load the released LoRA (rank 256, alpha 256, bf16), whereas LongLive ATS uses the SF/DMD base with the shared VACE module and no LoRA. The LongLive comparison is therefore not weight-matched. The missing LoRA's effect under tree inference is untested; neither a conservative bias nor isolated scheduler gains follow from this mismatch. All levels use four Euler denoising steps with timestep ladder $(1000, 750, 500, 250)$, guidance scale 1.0 (no unconditional CFG pass), and VACE context scale 1.5. The configured planner limits are 49–81 frames per leaf span and 81 frames for a packed root call. The realized plan has one root, 20 refinement nodes, and 360 dense leaves. The 359 interior leaves have planned spans of 80 frames and the final leaf a span of 81; every decoded leaf has 81 frames. Upstream checkpoint/repository hashes were not recorded. The prepared root-checkpoint pilot was not run; sensitivity to replacing the root checkpoint remains untested.

Sparse packing and anchors. Root and refinement use the same source-frame packer. With default zero warm-up, it concatenates one lead frame at the node start, four-frame windows around interior child boundaries (starting at $\max(\text{start}, \text{boundary} - 2)$), and a trailing four-frame block when the terminal-block flag is set. The default root layout with n children uses $1 + 4n$ frames; its 81-frame budget allows at most 20 such blocks. The conditioning operator is applied after packing. The Wan VAE temporal stride is four; the planner aligns boundaries to this stride and uses $1 + 4k$ clip budgets. Refinement and dense leaves receive the parent anchors as clean RGB at the first and last input slots, with the VACE mask zero at those slots. This is first/last-frame input conditioning, not clean-latent insertion or output pixel clamping. Leaves otherwise use contiguous source conditioning.

For a lead block of b frames and zero-based decoded indices, extraction retains frame 0 for the first child's left anchor, internal anchors at indices $b + 4i + 2$ for internal-boundary block index i, and the final decoded frame for the last child's right anchor. Assembly keeps the first decoded leaf whole and discards frame 0 of each subsequent leaf. There is no cross-fade, blending, or hard replacement of decoded endpoints by anchor pixels. For a planned span of s frames, an interior leaf is generated with $F = s + 1$ frames to include the shared right endpoint; the final leaf uses $F = s$. For J leaves the stitched length is

$$T_{\text{out}} = F_0 + \sum_{j=1}^{J-1} (F_j - 1). \tag{7}$$

The planner rounds the nominal 28800-frame target up to the nearest $4k + 1$ count. For all ten ATS runs, $T_{\text{out}} = 81 + 359 \times 80 = 28801$, matching the planner total, decoded array, and MP4 frame count. At 16 fps this is 1800.0625 s of encoded duration, with final-frame timestamp 1800 s.

AR conditioning and cache configuration. Both AR families use 12-frame chunks with three-frame overlap, block size 3, local attention size 12, sink size 3, and enabled global sinks. The maximum RoPE frequency-table sequence length is 1024 and the timestep shift is 5.0. The periodic-reset path refreshes the cache at approximately 192 s intervals. In the no-reset control, when the temporal position counter approaches the precomputed RoPE table limit, cached keys are re-rotated by $R(-N)$ on their temporal channels and the counter is decremented by N, retaining cache contents and relative positions. This control is not presented as a reimplementation of any other training-free method. Both use VACE context scale 1.5, seed 40, a nominal 28800-frame target, and an FP8-quantized UMT5-XXL text encoder. The LongLive adapter settings above apply to both AR variants; Reward Forcing has no adapter.

Temporal positions and randomness. Each node uses local temporal latent-grid RoPE indices $0, \ldots, f - 1$, where f is that call's latent temporal length. Absolute source-frame indices select conditioning only; no global-time or FPS conditioning reaches the model. All tree calls use full bidirectional attention. Each node reinitializes its random generator with seed 40 before drawing noise, so nodes with identical latent shapes receive identical noise. This is independent of execution order but couples the random draws: the independent-noise product in Equation (4) describes the idealized sampler, not the joint noise coupling of these fixed-seed runs. The calls remain executable in parallel once their parent anchors are available.

H EVALUATION PROMPTS

The following positive prompts are reproduced verbatim for the five source videos in the Wan evaluation.

H.1 DEPTH: BIRD FEEDER

A charming hand-drawn 2D animated scene of a wooden garden bird feeder with little songbirds flitting in to eat seeds. Classic frame-by-frame animation with bold confident ink outlines and flat cel-shaded color fills. Painterly brush-textured backgrounds with soft watercolor washes of leaves and dappled foliage. Warm sunny morning light. Cinematic palette of sage green, cream, soft amber, and tender rose. Polished hand-drawn feature animation aesthetic in the spirit of classic Disney or modern Cartoon Saloon.

H.2 EDGES: PASTEL LANDSCAPE

A real-time soft pastel landscape drawing video. An artist's hand uses chalky soft pastels to gradually build up a beautiful landscape scene on white paper resting on a warm wooden table, accompanied by an assortment of colorful pastel sticks and a soft cloth towel for blending. The drawing evolves from rough blocked-in shapes through layered colors into a richly detailed finished pastel painting. Soft chalky pastel strokes blend and overlap with painterly texture, fingers smudging and softening edges. Warm focused workspace lighting on the wood grain, intimate cinematic atmosphere, hand-drawn artisan aesthetic.

H.3 POSE: CONVERSATION

A Pixar-style 3D animated scene of two cartoon animal friends chatting in a cozy artisan coffee shop on a warm afternoon. The female character is an expressive tabby cat with soft fluffy fur, big sparkling green eyes, gentle whiskers, and subtle pink nose; she wears a soft cream cable-knit sweater. The male character is a cheerful golden retriever with floppy ears, warm caramel fur, kind brown eyes, and a tiny wagging tail; he wears a navy blue cardigan. Both are rendered with the rich subsurface scattering, detailed fur, and gentle facial expressions characteristic of modern Pixar animation. The cafe behind them has hanging Edison-bulb pendant lights casting an amber glow, exposed brick walls, leafy green plants on shelves, vintage books, and a marble-topped table with steam rising from ceramic mugs. Warm golden sunlight streams through tall windows, gentle dust motes in the air. Cinematic depth of field, lush color palette of cream, caramel, forest green, and warm amber, locked-off static camera framing the two characters in a clean two-shot, polished feature-film 3D animation aesthetic.

H.4 INPAINTING: ORCHESTRA

A string instrument orchestra actively performing on the concert hall stage. Bows sweep dynamically back and forth across taut strings. The first and second violinists lean in with their violins, arms in vigorous motion, fingers shifting up and down the fingerboard. The violists sway with the music. The cellists bow energetically with their right arms in continuous motion. The double bassists rock with the deep low notes. The conductor waves the baton through expressive arcs, arms cutting through the air, body shifting weight between sections. All musicians wear formal black concert attire. Warm theatrical stage lighting illuminates the wooden stage and the polished wood of the string instruments, against the dark concert hall background with the audience visible in silhouette below.

H.5 OUTPAINTING: COOKING

A professional cooking demonstration filmed in a modern test kitchen. A chef in a white chef's coat prepares food on a dark countertop, working with fresh ingredients, knives, cutting boards, and various cooking tools. Above the counter, warm overhead lights, a stainless steel range hood, and neutral upper cabinets are visible against a clean ceiling. Below the counter, the front of a sleek kitchen island with neutral cabinetry and drawers is in view. The atmosphere is calm, focused, and educational, with soft cinematic lighting throughout.