ASTAR: ACCELERATING SPACETIME AUTOREGRESSIVE VIDEO GENERATION VIA APPEARANCE INHERITANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Spacetime autoregressive (STAR) models extend next-scale visual autoregression to video through coarse-to-fine spacetime prediction, yet their late high-resolution scales remain computationally expensive. We analyze how representations evolve during clip-wise generation and uncover Appearance Inheritance: much of the visual state established by the first clip persists as generation proceeds, while larger changes concentrate around evolving content. Building on this observation, we introduce ASTAR, a training-free acceleration framework that uses the first clip as a transformation reference for later clips. ASTAR reserves full Transformer computation for selected tokens and approximates the remaining positions with first-clip-derived layer transformations, while residual evidence from current computation constrains the approximation before multi-scale accumulation. ASTAR achieves $2.40\times$ and $1.93\times$ end-to-end speedup at 720p and 480p, respectively, with less than a 1% relative decrease in VBench Total at either resolution.

1 INTRODUCTION

Visual autoregressive (VAR) models replace raster-order next-token generation with next-scale prediction over coarse-to-fine token maps, enabling parallel prediction within each scale (Tian et al., 2024; Han et al., 2025). Spacetime autoregressive (STAR) modeling extends this hierarchy to video with clip-wise spacetime pyramids (Liu et al., 2025b). Compared with next-token autoregression, next-scale prediction requires fewer sequential prediction steps. Nevertheless, Transformer computation at the last several spacetime scales remains costly because their token maps contain many positions (Liu et al., 2025b; Yune et al., 2026).

Most existing acceleration methods for next-scale autoregressive models are designed for image generation and exploit redundancy during late-stage refinement. These approaches reduce computation through mechanisms such as token pruning, low-rank approximation, and structured sparsification (Guo et al., 2025; Li et al., 2026; Chen et al., 2026). However, video generation also introduces temporal redundancy across successive clips. Recent STAR-specific approaches exploit redundancy in the spacetime hierarchy: FastSTAR exploits spatial convergence and inter-clip temporal similarity to select tokens (Yune et al., 2026), while SparSTAR reduces late-scale attention cost while retaining all tokens and refinement scales (Lee et al., 2026). However, these mechanisms do not explicitly use first-clip layer transformations to update later-clip states. ASTAR exploits this cross-clip relationship by using the first clip as an appearance reference while adapting computation to evolving motion and newly refined content.

Our analysis reveals Appearance Inheritance: the first clip establishes persistent background, color, illumination, and coarse scene structure. Later representations retain much of this appearance even as motion, deformation, disocclusion, and newly refined details introduce differences. Figure 1 illustrates this persistence through token-level cosine-similarity and relative ℓ_2 feature-change distributions, together with example frames. The high-similarity concentration is consistent with the preserved scene layout, lighting, and object appearance in the examples, despite changes in viewpoint and object position. This agreement between representation-level similarity and visual persistence motivates ASTAR to prioritize full computation for changing content and constrain reused updates before multi-scale accumulation.

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.

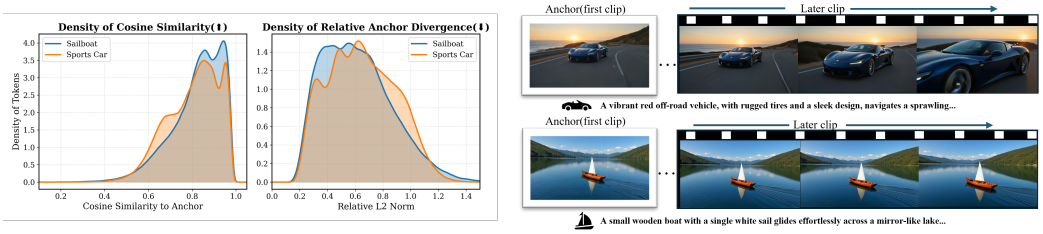


Figure 1: **Appearance Inheritance.** Left: distributions of cosine similarity and relative ℓ_2 feature change between corresponding first- and later-clip tokens. Right: example videos in which appearance persists across clips while content evolves.

Our contributions are summarized as follows:

- We identify and characterize Appearance Inheritance in STAR inference: much of the visual state established by the first clip persists during subsequent generation, providing a stable appearance reference for selective computation.
- We propose ASTAR, a unified training-free inference framework that uses the first clip as a transformation reference, routes full computation from current hidden states, and constrains reused updates with residual evidence from computed tokens.
- ASTAR achieves $2.40\times$ and $1.93\times$ speedup at 720p and 480p, with relative VBench Total decreases of 0.76% and 0.35%, respectively.

2 Related Work

2.1 Visual Autoregressive Generation

VAR replaces raster decoding with next-scale prediction over progressively finer token maps (Tian et al., 2024). Infinity scales this formulation with bitwise visual tokens (Han et al., 2025), and InfinityStar combines an initial image pyramid with later spacetime clip pyramids (Liu et al., 2025b). Next-scale prediction reduces the number of sequential steps relative to next-token autoregression, but the large token maps at fine video scales remain computationally expensive.

2.2 Efficient Visual Generative Inference

To reduce the cost of late, high-resolution refinement in next-scale image generation, FastVAR caches converged content and selects tokens (Guo et al., 2025); SparseVAR and SkipVAR reduce late-stage work through frequency-aware sparsification or scale skipping or guidance skipping (Chen et al., 2025a; Li et al., 2025a); ToProVAR coordinates token-, layer-, and scale-level sparsity (Chen et al., 2026); and FasterVAR studies stage-dependent redundancy and low-rank structure (Li et al., 2026). Complementary inference-efficiency techniques include dynamic token selection, merging, collaborative prediction, and KV-cache optimization (Rao et al., 2021; Liang et al., 2022; Ryoo et al., 2021; Bolya et al., 2022; Chen et al., 2025b; Li et al., 2025b; Qin et al., 2026). These approaches motivate selective computation but do not directly model STAR's first-to-later-clip relationship.

A parallel line of work accelerates diffusion-based image and video generation through feature reuse across denoising steps or sparse attention, including DeepCache, PAB, TeaCache, ERTACache, and training-free sparse video attention (Ma et al., 2024; Zhao et al., 2024; Liu et al., 2025a; Peng et al., 2026; Luo et al., 2026). Their reuse signals arise from denoising trajectories or generic attention redundancy rather than STAR's discrete cross-clip hierarchy.

Within the clip-wise spacetime hierarchy, FastSTAR exploits spatiotemporal token convergence (Yune et al., 2026), whereas SparSTAR sparsifies late-scale attention while preserving tokens and refinement scales (Lee et al., 2026). Their token-selection and sparse-attention mechanisms leave open how to use first-clip layer transformations to update a later clip while respecting its current state. ASTAR combines such transformation reuse with current-state routing and residual correction.

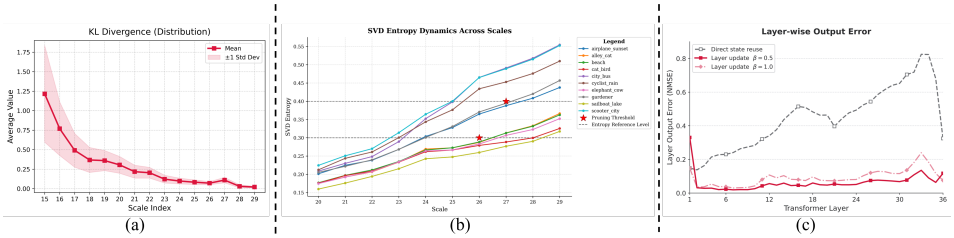


Figure 2: **Empirical signals behind ASTAR.** (a) The KL divergence between distributions derived from conditional and unconditional prediction-head logits decreases sharply at late scales, indicating diminishing branch discrepancy. (b) Temporally centered SVD entropy is higher for videos with stronger motion at matched late scales, so it serves as a video-conditioned signal for sparse onset. (c) Layer-wise output NMSE for direct first-clip state reuse and first-clip layer-update transfer with $\beta \in \{0.5, 1.0\}$; lower is better.

3 METHOD

3.1 PRELIMINARIES

3.1.1 STAR GENERATION AND GUIDED INFERENCE

A STAR model generates N clips $\{c^1, \ldots, c^N\}$, where N is the clip count. The first clip c^1 follows an image-style pyramid; each later clip c^c uses a spacetime pyramid (Liu et al., 2025b). Within clip c^c, let J_c be the number of ordered prediction steps and b_j^c the discrete residual token block generated at step j. Tokens within a block are predicted jointly, conditioned on earlier clips and previously generated blocks of the current clip. Given text prompt y and its encoded condition $\psi(y)$, the generation process is

$$p(\mathcal{B}^{1:N} \mid y) = \prod_{c=1}^{N} \prod_{j=1}^{J_c} p\big(b_j^c \mid \mathcal{B}^{<c}, b_{<j}^c, \psi(y)\big)\,. \tag{1}$$

Here $\mathcal{B}$ collects the predicted blocks over the indicated clips. STAR inference uses Adaptive Projected Guidance (APG) (Sadat et al., 2025), which evaluates conditional and unconditional prediction branches at guided scales.

3.1.2 MULTI-SCALE RESIDUAL REPRESENTATION

Each predicted block b_j^c encodes a residual. After sampling the predicted bit tokens, the Binary Spherical Quantization (BSQ)-based tokenizer used by InfinityStar converts them into a continuous residual code tensor $\mathbf{z}_j^c$ (Liu et al., 2025b). The residual code is then upsampled and accumulated with the representation from preceding prediction steps,

$$\mathbf{F}_j^c = \mathbf{F}_{j-1}^c + \mathrm{Up}\big(\mathbf{z}_j^c\big)\,. \tag{2}$$

Here $\mathbf{F}_j^c$ is the current clip's accumulated code after step j, and Up maps the decoded residual $\mathbf{z}_j^c$ to the accumulation grid. ASTAR corrects approximate residuals before the addition in Eq. (2).

3.2 EMPIRICAL MOTIVATION

We examine three signals that motivate ASTAR's selective inference path. First, the two APG branches become increasingly similar as refinement proceeds. Second, temporal SVD entropy differs across videos at the same late refinement scales. Third, we test whether a Transformer transformation observed in the first clip can approximate the transformation that would otherwise be computed densely in later clips.

Late guidance discrepancy. CausVid distills a bidirectional video diffusion model into a causal generator using distribution matching based on reverse KL divergence (Yin et al., 2025). Motivated by this use of KL to quantify distributional discrepancy, we measure the KL divergence between

distributions obtained by applying softmax to the conditional and unconditional prediction-head logits before APG combines the branches. Unlike the training objective in CausVid, our KL measurement is an inference-time diagnostic; ASTAR requires no distillation training. Figure 2(a) shows a substantial decrease at late scales. ASTAR therefore retains both branches at early scales and, after a fixed late switch k_g, continues with conditional prediction only. Table 3 includes a variant without late guidance reduction to test this quality–speed trade-off.

Temporal variation and sparse onset. A fixed sparse onset cannot adapt to differences in temporal variation across videos. To quantify these differences, we flatten each of the T latent frames of the current clip's accumulated representation at scale k into a vector in $\mathbb{R}^{D_k}$ and stack the vectors as rows of $\mathbf{M}_k \in \mathbb{R}^{T\times D_k}$. With temporal mean $\boldsymbol{\mu}_k = T^{-1}\mathbf{M}_k^\top \mathbf{1}_T$ and all-ones vector $\mathbf{1}_T$, we center the frames and take their thin singular value decomposition:

$$\widetilde{\mathbf{M}}_k = \mathbf{M}_k - \mathbf{1}_T\boldsymbol{\mu}_k^\top = \mathbf{U}_k \operatorname{diag}(\sigma_{k,1},\ldots,\sigma_{k,J})\mathbf{V}_k^\top, \qquad J = \min(T, D_k), \tag{3}$$

where $\mathbf{U}_k \in \mathbb{R}^{T\times J}$ and $\mathbf{V}_k \in \mathbb{R}^{D_k\times J}$ have orthonormal columns, and the nonnegative $\sigma_{k,j}$ are the singular values in descending order. Centering removes the time-invariant component. The squared singular values define an energy distribution and its entropy:

$$p_{k,j} = \frac{\sigma_{k,j}^2}{\sum_{m=1}^{J}\sigma_{k,m}^2}, \qquad \mathcal{H}_k = -\sum_{j=1}^{J} p_{k,j}\log p_{k,j}. \tag{4}$$

Lower entropy indicates that temporal differences are concentrated in fewer modes, without implying that motion vanishes. The main observation in Figure 2(b) is the separation across videos at matched late scales: videos with stronger motion exhibit higher temporal SVD entropy, whereas lower-motion videos exhibit lower entropy. ASTAR therefore uses entropy as a compact indicator of video dynamics.

ASTAR probes this dynamic indicator at two late scales. The first threshold crossed selects one of three nearby sparse onsets; if neither is crossed, the latest onset is used. The selected onset level and threshold-relative probe entropy determine a complexity value $q \in [0, 1]$: earlier onset levels occupy lower intervals of q, and larger entropy within a level retains more computation. The probe-to-budget calculation is described in Appendix A.3. At sparse scale k, the retained full-compute fraction is

$$\rho_k(q) = \operatorname{clip}(\bar{\rho}_k + \alpha(2q-1), \rho_{\min}, \rho_{\max}), \tag{5}$$

where $\bar{\rho}_k$ is the reference compute fraction, α controls adaptation, and $0 < \rho_{\min} \le \rho_{\max} \le 1$ bound the retained fraction; $\operatorname{clip}(u, a, b)$ restricts u to $[a, b]$. The probe scales, thresholds, and reference budgets are specified in Sec. 4.1.

Layer-wise approximation error. We compare two reuse rules against dense output $\mathbf{y}_i^\ell = \Phi^\ell(\mathbf{x}^\ell)_i$ at a later-clip position i and Transformer layer ℓ. Given aligned first-clip states $\mathbf{a}_i^\ell$ and $\mathbf{a}_i^{\ell+1}$, direct state reuse predicts $\widehat{\mathbf{y}}_{i,\text{direct}}^\ell = \mathbf{a}_i^{\ell+1}$, whereas layer-update transfer predicts $\widehat{\mathbf{y}}_{i,\beta}^\ell = \mathbf{x}_i^\ell + \beta(\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^\ell)$ for $\beta \in \{0.5, 1.0\}$. We evaluate each rule at positions $\mathcal{T}$ assigned to reuse by the production router using the layer-wise normalized squared error $E_\ell(\widehat{\mathbf{y}}) = \frac{\sum_{i\in\mathcal{T}}\|\widehat{\mathbf{y}}_i^\ell - \mathbf{y}_i^\ell\|_2^2}{\sum_{i\in\mathcal{T}}\|\mathbf{y}_i^\ell\|_2^2}$. Figure 2(c) shows the resulting error across Transformer depth at the final 720p refinement scale. Both rules are evaluated against the same dense trajectory without feeding their outputs back into that trajectory.

3.3 Token Routing and Transformation Reuse

The schedule above supplies the sparse onset and compute budget. At those scales, ASTAR selects positions for full Transformer computation and uses first-clip layer transformations to update the rest. Figure 3 summarizes the path from token routing to residual correction.

Token routing. The first clip is generated by the original dense path. At sparse scales, ASTAR uses the current hidden representation to prioritize positions that differ from the inherited first-clip state or exhibit stronger local dynamics.

For token i, let $\mathbf{x}_i$ be its current hidden state, $\mathbf{a}_i$ the spatially corresponding first-clip reference repeated at the current temporal position, and $\mathbf{x}_{\pi(i)}$ the corresponding token in an adjacent frame.

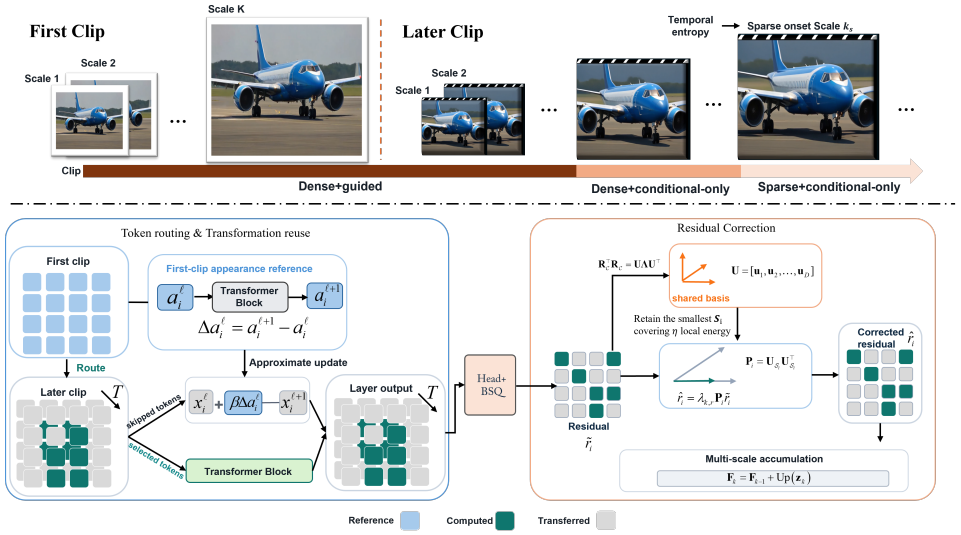


Figure 3: **ASTAR inference pipeline.** Appearance Inheritance provides a first-clip transformation reference. Temporal entropy measured from the current clip determines when sparse execution begins, while token-level routing scores determine which positions retain full Transformer computation. Positions outside the computed set receive the corresponding first-clip transformation, and residuals from the computed positions constrain the resulting residual representation update before it is accumulated.

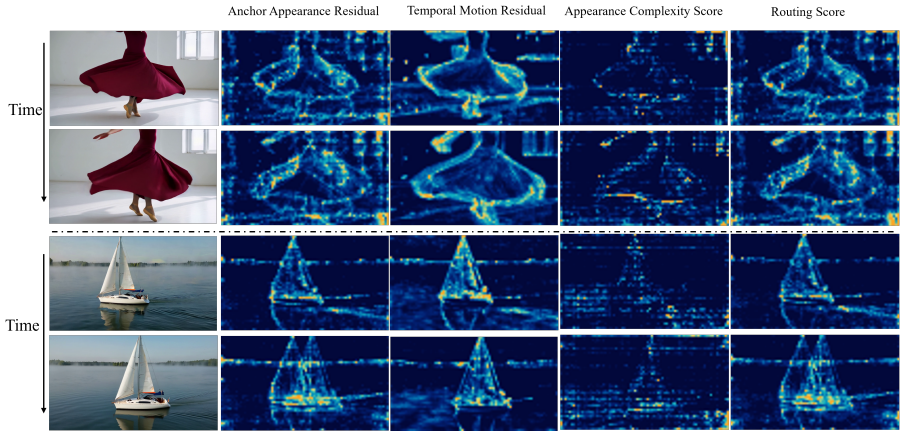


Figure 4: Token-routing signals and the resulting routing score. Anchor deviation and temporal change provide complementary cues about where current processing is most valuable, while channel-wise feature variation raises the priority of information-rich responses.

We define the Anchor Appearance Residual (AAR) and Temporal Motion Residual (TMR) as

$$d_i^{\mathrm{AAR}} = \frac{\|\mathbf{x}_i - \mathbf{a}_i\|_2}{\|\mathbf{a}_i\|_2}, \qquad d_i^{\mathrm{TMR}} = \frac{\|\mathbf{x}_i - \mathbf{x}_{\pi(i)}\|_2}{\|\mathbf{x}_{\pi(i)}\|_2}. \tag{6}$$

We use $d_i^{\mathrm{nov}} = \max(d_i^{\mathrm{AAR}}, d_i^{\mathrm{TMR}})$ so that either appearance change or short-range temporal change can promote full computation. The Appearance Complexity Score (ACS) normalizes each token's channel-wise feature standard deviation by its mean over the token positions $\mathcal{I}_k$ in the current scale step. Defining $\bar{v}_k = |\mathcal{I}_k|^{-1} \sum_{j \in \mathcal{I}_k} \mathrm{Std}_d(\mathbf{x}_j)$, we compute

$$a_i = \frac{\mathrm{Std}_d(\mathbf{x}_i)}{\bar{v}_k}, \qquad s_i = d_i^{\mathrm{nov}} \exp(a_i - 1). \tag{7}$$

Here Std_d denotes standard deviation across feature channels, $\mathcal{I}_k$ the positions of the current scale step, and s_i the routing score. For scale k, let r index its repeated prediction steps. At each step (k, r), the compute budget $\rho_k(q)$ specifies the fraction of positions in $\mathcal{C}_{k,r}$. We select positions in descending order of s_i, first meeting a minimum allocation per frame and then filling the remaining budget globally. The other positions form the transformation-reuse set $\mathcal{T}_{k,r}$. When routing is recomputed across layers, the selected sets have an intersection-over-union (IoU) of 0.94–0.99. ASTAR therefore selects $\mathcal{C}_{k,r}$ once per step and keeps it fixed across Transformer layers; Sec. 4.4 reports the corresponding ablation.

First-clip transformation reuse. Let $\mathbf{a}_i^\ell$ and $\mathbf{a}_i^{\ell+1}$ denote the hidden states at the spatially corresponding first-clip position before and after Transformer layer ℓ. Their difference represents the transformation produced by that Transformer layer on the inherited appearance. Positions in $\mathcal{C}_{k,r}$ follow the original Transformer computation, while $i \in \mathcal{T}_{k,r}$ applies the corresponding scaled first-clip transformation to its current hidden state:

$$\mathbf{x}_i^{\ell+1} = \mathbf{x}_i^\ell + \beta \left(\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^\ell \right), \qquad i \in \mathcal{T}_{k,r}. \tag{8}$$

This preserves the current-clip state and applies only a transformation derived from the inherited appearance, rather than replacing the token by a stale first-clip feature. Figure 2(c) and Table 3 compare this reuse rule with direct state substitution. Because the result is approximate, ASTAR constrains its decoded residual before accumulation.

3.4 Residual Correction

Selective Transformer execution leaves transfer positions with approximate hidden states, while STAR accumulates decoded residuals across scales as in Eq. (2). The transferred update may not fully match the residual directions and magnitude required by the current clip. ASTAR uses neighboring computed residuals as current evidence for which residual-space directions should be written back.

Current-supported residual subspace. Let D be the decoded residual-code dimension, distinct from the Transformer hidden dimension, and let $\mathbf{r}_j \in \mathbb{R}^D$ be the residual at computed position j. Stack these vectors as rows of $\mathbf{R}_\mathcal{C} \in \mathbb{R}^{|\mathcal{C}_{k,r}| \times D}$. For each routed step, ASTAR forms the uncentered residual second-moment matrix and eigendecomposes it:

$$\mathbf{G}_{k,r} = \mathbf{R}_\mathcal{C}^\top \mathbf{R}_\mathcal{C} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^\top. \tag{9}$$

The columns of $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_D]$ provide a shared orthonormal coordinate system for the residual space. For an approximated position i, we use only the computed residuals in its radius-one spacetime neighborhood $\mathcal{N}(i)$, excluding i itself, to determine which of these directions are locally supported. Specifically, the support assigned to direction $\mathbf{u}_m$ is

$$e_{i,m} = \sum_{j \in \mathcal{N}(i) \cap \mathcal{C}_{k,r}} \langle \mathbf{r}_j, \mathbf{u}_m \rangle^2. \tag{10}$$

We rank directions by $e_{i,m}$ and let $\mathcal{S}_i \subseteq \{1, \ldots, D\}$ be the smallest set whose cumulative energy reaches fraction $\eta \in (0, 1]$ of the total local energy. If no computed neighbor provides support, ASTAR skips the residual write-back at that position for this step; subsequent refinement steps can still update it. The chosen η is given in Sec. 4.1. The candidate residual $\widetilde{\mathbf{r}}_i$ decoded from the approximated hidden state is then restricted to the subspace spanned by the supported directions:

$$\mathbf{P}_i = \mathbf{U}_{\mathcal{S}_i} \mathbf{U}_{\mathcal{S}_i}^\top, \qquad \mathbf{p}_i = \mathbf{P}_i \widetilde{\mathbf{r}}_i. \tag{11}$$

This operation preserves the candidate's coefficients along directions supported by current computation and suppresses components along unsupported directions.

Why local support is informative. The projection is useful only if directions inferred from neighboring computed tokens are also relevant to what full computation would produce at the omitted position. For analysis only, we replay the same generation prefix with full computation at an approximated position i and denote its residual by $\mathbf{r}_i^\star$. The projector $\mathbf{P}_i$ is still built only from neighboring computed residuals. We then measure the full-residual subspace fit $A_i = \|\mathbf{P}_i \mathbf{r}_i^\star\|_2^2 / \|\mathbf{r}_i^\star\|_2^2$. A high



Figure 5: **Analysis of ASTAR's residual correction.** (a) Mean dense-reference subspace fit for directions selected only from neighboring computed residuals, compared with the expected fit of a randomly oriented subspace of the same dimensionality. (b) Mean analysis-only optimal commitment scale $\lambda^\star$ after support projection; the dashed line at $\lambda = 1$ denotes full write-back. (c) Residual error relative to transformation reuse alone after support projection and after projection plus scaling.

A_i means that the dense-reference residual is largely representable within the subspace identified from nearby computed tokens.

To control for subspace capacity, let $m_i = |\mathcal{S}_i|$. A uniformly oriented m_i-dimensional subspace, sampled independently of the nonzero residual, has expected fit m_i/D. Appendix A.5 derives both the fit identity and this dimension-matched expectation. Figure 5(a) reports the mean fit at each analyzed scale and compares the locally supported subspace with this dimension-matched expectation. The gap at the first analyzed scale indicates that nearby computed positions identify directions aligned with the omitted residual beyond what is explained by subspace size alone. At finer scales the selected support becomes broader and the gap narrows, so the projection becomes less restrictive.

Residual commitment calibration. Projection determines which residual directions are retained, but not how strongly the projected candidate should be written back. To isolate this issue, we compute the analysis-only least-squares scale $\lambda_i^\star = \mathrm{clip}_{[0,1]} \frac{\langle \mathbf{P}_i \widetilde{\mathbf{r}}_i, \mathbf{r}_i^\star \rangle}{\|\mathbf{P}_i \widetilde{\mathbf{r}}_i\|_2^2}$. Figure 5(b) shows that $\lambda_i^\star$ remains well below the full-write reference $\lambda = 1$, so full-strength write-back is overly aggressive even after projection.

During inference, ASTAR estimates the write-back scale from computed probe positions near the routing boundary; the cap is specified in Sec. 4.1. Let $\mathcal{B}_{k,r} \subseteq \mathcal{C}_{k,r}$ denote these probes. The normal computation provides $\mathbf{r}_j$, while a matched approximation path at the probe provides $\mathbf{p}_j^{\mathrm{probe}}$. We estimate one scale for the routed step:

$$\lambda_{k,r} = \mathrm{clip}_{[0,1]} \frac{\sum_{j \in \mathcal{B}_{k,r}} \langle \mathbf{p}_j^{\mathrm{probe}}, \mathbf{r}_j \rangle}{\sum_{j \in \mathcal{B}_{k,r}} \|\mathbf{p}_j^{\mathrm{probe}}\|_2^2}, \tag{12}$$

and commit

$$\widehat{\mathbf{r}}_i = \lambda_{k,r} \mathbf{P}_i \widetilde{\mathbf{r}}_i. \tag{13}$$

Computed positions retain their original residuals. Transfer positions without local support simply receive no residual write-back at this scale.

Figure 5(c) directly measures the effect of the completed correction. For each analyzed approximated position, we compute squared residual error to the same-prefix dense-reference residual and normalize it by the error of transformation reuse alone; the Transfer bar is therefore 1 by construction. Support projection reduces this normalized error by 11.5%, while projection followed by scaling reduces it by 39.7%. Together, the three panels separate the two roles of residual correction: neighboring computed residuals identify supported directions, and online calibration controls how strongly the supported update is accumulated.

Table 1: Five-second text-to-video comparison at 480p and 720p. PSNR, SSIM, and LPIPS measure paired fidelity to dense InfinityStar outputs. Among accelerated methods within each resolution, **bold** and underlined indicate the best and second-best reported values per column. Dashes denote unreported values; external quality scores follow the cited studies.

Method	Speedup ↑	PSNR ↑	SSIM ↑	LPIPS ↓	Quality ↑	Semantic ↑	Total ↑
480p text-to-video							
InfinityStar	1.00×	–	–	–	84.03	83.30	83.89
FastSTAR (Yune et al., 2026)	1.47×	22.96	0.697	0.262	–	–	–
FasterVAR (Li et al., 2026)	1.76×	–	–	–	83.43	82.93	83.33
SparSTAR (Lee et al., 2026)	1.30×	**25.56**	**0.793**	**0.123**	–	–	83.48
ASTAR (ours)	**1.93**×	25.07	0.790	0.124	**83.76**	**82.94**	**83.60**
720p text-to-video							
InfinityStar	1.00×	–	–	–	84.08	82.75	83.82
SparseVAR	1.44×	26.59	0.789	0.228	82.70	79.86	82.13
FastVAR	1.61×	25.00	0.757	0.266	81.20	81.18	81.19
ToMe	1.76×	26.47	0.790	0.227	81.98	80.10	81.60
FastSTAR (Yune et al., 2026)	2.01×	28.29	0.828	0.185	**83.59**	81.51	83.18
FasterVAR (Li et al., 2026)	**2.53**×	–	–	–	83.15	**82.43**	83.00
SparSTAR (Lee et al., 2026)	1.60×	**31.22**	**0.898**	**0.065**	–	–	**83.83**
ASTAR (ours)	2.40×	28.20	0.854	0.108	83.39	82.34	83.18

4 Experiments

4.1 Experimental Setup

Backbone and evaluation. We evaluate ASTAR on the 8B InfinityStar text-to-video model (Liu et al., 2025b) without retraining. We report 5-second, 81-frame generation at 720p and 480p. We evaluate the 16 VBench dimensions (Huang et al., 2024) with five generations per prompt and report the official Quality, Semantic, and Total aggregates; prompt provenance is detailed in Appendix A. We report end-to-end latency from text encoding through autoregressive generation and VAE decoding.

ASTAR configuration. At 720p, ASTAR uses conditional-only guidance from $S25$. Temporal SVD entropy at $S25$ and $S26$ selects sparse onset from $\{S26, S27, S28\}$: $\mathcal{H}_{25} < 0.3$ selects $S26$, otherwise $\mathcal{H}_{26} < 0.4$ selects $S27$, and the remaining videos begin at $S28$. The base full-compute ratios for $S26$–$S29$ are $(0.98, 0.85, 0.75, 0.55)$ with adaptation amplitude $\alpha = 0.2$. We use $\beta = 0.5$, a per-frame routing floor factor of 0.8, residual-support energy retention $\eta = 0.9$, and at most 512 computed positions for calibrating each routed scale step. The high-compute operating point changes the late-scale ratios to $(0.980, 0.865, 0.775, 0.600)$; the low-compute point uses $(0.976, 0.820, 0.680, 0.480)$. All other settings remain fixed.

At 480p, conditional-only guidance begins at $S23$. Entropy probes at $S24$ and $S25$ use thresholds 0.4 and 0.5 to select sparse onset from $\{S25, S26, S27\}$, and the base ratios for $S25$–$S27$ are $(0.98, 0.80, 0.60)$. Other method parameters follow the 720p setting.

Baselines. The SparseVAR, FastVAR, ToMe, and FastSTAR results follow FastSTAR (Yune et al., 2026); SparSTAR results follow SparSTAR (Lee et al., 2026); FasterVAR quality scores follow FasterVAR (Li et al., 2026).

4.2 Quality–Efficiency Comparison

Table 1 compares both resolutions. At 480p, ASTAR has the highest speedup and VBench Total among the listed accelerated methods. At 720p, it achieves $2.40\times$ speedup with a Total of 83.18. Results reported in different studies reflect their respective evaluation settings.

Table 2: 720p ASTAR operating points obtained by varying only the late-scale full-compute ratios. PSNR, SSIM, and LPIPS measure fidelity to matched dense InfinityStar outputs.

720p setting	Speedup ↑	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	Quality ↑	Semantic ↑	Total ↑
ASTAR (high-compute)	$2.31\times$	28.27	0.856	0.106	83.44	82.42	83.23
ASTAR (default)	$2.40\times$	28.20	0.854	0.108	83.39	82.34	83.18
ASTAR (low-compute)	$2.55\times$	28.04	0.850	0.114	83.13	82.22	82.95

Table 3: Component ablation at 720p. Only the first row retains both late APG branches; all other rows use late guidance reduction.

Variant	Speedup ↑	Quality ↑	Semantic ↑	Total ↑
ASTAR without late guidance reduction	$1.54\times$	83.62	82.58	83.41
Direct state transfer	$2.46\times$	82.25	81.71	82.14
First-clip transformation reuse	$2.39\times$	82.39	81.88	82.29
+ Layer-consistent routing	$2.43\times$	82.90	82.17	82.76
ASTAR	$2.40\times$	83.39	82.34	83.18

4.3 Operating-Point Sweep

Table 2 varies only the late-scale compute budget. Increasing the retained compute raises VBench Total from 82.95 to 83.23, while the end-to-end speedup varies from $2.55\times$ to $2.31\times$. This exposes the expected Pareto trade-off without changing the routing criterion, transfer rule, or residual correction.

4.4 Component Ablation

We compare five variants with the same sparse-compute budgets in Table 3. The first row disables only late guidance reduction, retaining both APG branches while keeping transformation reuse, layer-consistent routing, and residual correction. All other rows use conditional-only prediction at late scales. Direct state transfer substitutes the aligned first-clip output for each reused state; first-clip transformation reuse instead adds the layer update in Eq. (8) to the current state. These two variants omit layer-consistent routing and residual correction. The next row fixes the computed set across layers within each repetition, and the last also applies residual correction.

Retaining late guidance raises Total from 83.18 to 83.41 but reduces speedup from $2.40\times$ to $1.54\times$. The remaining rows isolate the effects of transfer, routing, and correction. Transformation reuse improves over direct state substitution because it preserves the current hidden state instead of replacing it with an older first-clip feature. Layer-consistent routing further raises Total from 82.29 to 82.76, and residual correction reaches 83.18 by constraining approximate residual updates with evidence from current computation.

5 Conclusion

ASTAR accelerates spacetime autoregressive video generation by exploiting Appearance Inheritance as a cross-clip transformation reference. It retains full Transformer computation at selected positions, approximates the remaining positions with first-clip-derived layer updates, and uses residual evidence from current computation to constrain these approximations before multi-scale accumulation. On 720p and 480p text-to-video generation, ASTAR reaches $2.40\times$ and $1.93\times$ speedup, respectively, with relative VBench Total decreases of 0.76% and 0.35%, without retraining.

AI USE STATEMENT

We used generative AI tools to assist with drafting and polishing the manuscript, as well as developing and debugging research code. The authors reviewed the AI-assisted work and take responsibility for the final content of this paper.

REFERENCES

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

Jiayu Chen, Ruoyu Lin, Zihao Zheng, Jingxin Li, Maoliang Li, Guojie Luo, and Xiang Chen. Toprovar: Efficient visual autoregressive modeling via tri-dimensional entropy-aware semantic analysis and sparsity optimization. In *International Conference on Learning Representations*, 2026.

Zhuokun Chen, Jugang Fan, Zhuowei Yu, Bohan Zhuang, and Mingkui Tan. Frequency-aware autoregressive modeling for efficient high-resolution image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17140–17149, 2025a.

Zigeng Chen, Xinyin Ma, Gongfan Fang, and Xinchao Wang. Collaborative decoding makes visual auto-regressive modeling efficient. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 23334–23344, 2025b.

Hang Guo, Yawei Li, Taolin Zhang, Jiangshan Wang, Tao Dai, Shu-Tao Xia, and Luca Benini. Fastvar: Linear visual autoregressive modeling via cached token pruning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19011–19021, 2025.

Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15733–15744, 2025.

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*, pp. 21807–21818, 2024.

Jongbeom Lee, Hyunwoo Yu, Jincheol Yang, Jaemin Choi, and Suk-Ju Kang. SparSTAR: Sparse attention for SpaceTime AutoRegressive video synthesis, 2026. URL https://arxiv.org/abs/2608.10519.

Jiajun Li, Yue Ma, Xinyu Zhang, Qingyan Wei, Songhua Liu, and Linfeng Zhang. Skipvar: Accelerating visual autoregressive modeling via adaptive frequency-aware skipping. *arXiv preprint arXiv:2506.08908*, 2025a.

Kunjun Li, Zigeng Chen, Cheng-Yen Yang, and Jenq-Neng Hwang. Memory-efficient visual autoregressive modeling with scale-aware kv cache compression. *arXiv preprint arXiv:2505.19602*, 2025b.

Senmao Li, Kai Wang, Salman Khan, Fahad Shahbaz Khan, Jian Yang, and Yaxing Wang. Fastervar: Plug-and-play acceleration for visual autoregressive models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2512.16483.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*, 2022.

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

Jinlai Liu, Jian Han, Bin Yan, Hui Wu, Fengda Zhu, Xing Wang, Yi Jiang, Bingyue Peng, and Zehuan Yuan. Infinitystar: Unified spacetime autoregressive modeling for visual generation. In *Advances in Neural Information Processing Systems*, 2025b.

Jiayi Luo, Jiayu Chen, Jiankun Wang, Cong Wang, Hanxin Zhu, Qingyun Sun, Chen Gao, Zhibo Chen, and Jianxin Li. Training-free sparse attention for fast video generation via offline layer-wise sparsity profiling and online bidirectional co-clustering. *arXiv preprint arXiv:2603.18636*, 2026.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15762–15772, 2024.

Xurui Peng, Chenqian Yan, Hong Liu, Rui Ma, Fangmin Chen, Xing Wang, Zhihua Wu, Songwei Liu, and Mingbao Lin. Ertacache: Error rectification and timesteps adjustment for efficient diffusion. In *International Conference on Learning Representations*, 2026.

Ziran Qin, Youru Lv, Mingbao Lin, Hang Guo, Zeren Zhang, Danping Zou, and Weiyao Lin. Head-aware kv cache compression for efficient visual autoregressive modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 24982–24990, 2026.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

Michael Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: Adaptive space-time tokenization for videos. *Advances in neural information processing systems*, 34:12786–12797, 2021.

Seyedmorteza Sadat, Otmar Hilliges, and Romann M. Weber. Eliminating oversaturation and artifacts of high guidance scales in diffusion models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=e2ONKX6qzJ.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Frédo 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.

Sungwoong Yune, Suheon Jeong, and Joo-Young Kim. Faststar: Spatiotemporal token pruning for efficient autoregressive video synthesis. In *European Conference on Computer Vision*, 2026.

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

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

A Reproducibility and Evaluation Details

A.1 VBench protocol and prompts

We use the original 16-dimension VBench protocol for T2V: subject consistency, background consistency, temporal flickering, motion smoothness, dynamic degree, aesthetic quality, imaging quality, object class, multiple objects, human action, color, spatial relationship, scene, appearance style, temporal style, and overall consistency. Generation uses the `refined_prompt` entries in the evaluation JSON distributed with InfinityStar (Liu et al., 2025b). VBench evaluation uses five seeded videos per prompt in each dimension's designated subset.

A.2 Hardware and software

Generation experiments use an NVIDIA H100 GPU with CUDA 12.1. Metric evaluation uses Python 3.10, PyTorch 2.5.1+cu121, torchvision 0.20.1+cu121, NumPy 1.26.4, scikit-image 0.25.2, OpenCV 4.10.0, and Pillow 12.3.0.

A.3 Entropy probes and adaptive budget

Equation (4) gives the idealized temporally centered entropy; $\mathcal{H}_k$ uses natural logarithms and is not divided by $\log J$. Numerical safeguards are specified below. Let (k_a, k_b) be the two probe scales, (τ_a, τ_b) their thresholds, and (s_0, s_1, s_2) the three candidate onsets. At 720p these are $(25, 26)$, $(0.3, 0.4)$, and $(26, 27, 28)$; at 480p they are $(24, 25)$, $(0.4, 0.5)$, and $(25, 26, 27)$. Writing $H_a = \mathcal{H}_{k_a}$ and $H_b = \mathcal{H}_{k_b}$, the implementation maps the probe result to

$$(k_s, q) = \begin{cases} \left(s_0, \dfrac{H_a}{3\tau_a}\right), & H_a < \tau_a, \\ \left(s_1, \dfrac{1 + H_b/\tau_b}{3}\right), & H_a \geq \tau_a,\ H_b < \tau_b, \\ \left(s_2, 1 - \dfrac{\tau_b}{3H_b}\right), & H_a \geq \tau_a,\ H_b \geq \tau_b. \end{cases} \tag{14}$$

The three onset levels occupy successive thirds of $[0, 1]$. Within a threshold-crossing level, the position is proportional to entropy divided by that level's threshold; the fallback uses a saturating excess-entropy mapping. The implementation clips the within-level value and q to $[0, 1]$, and its complexity-mapping helper returns $q = 1$ for non-finite input entropy or a non-positive threshold. Before a probe decision, the latest onset and $q = 1$ are used. A probe is evaluated only after the last repetition of its scale, and its result affects subsequent scheduled steps. If the first probe does not cross its threshold, the defaults remain in force until the second probe; H_b is not evaluated prematurely. The resulting q is inserted into Eq. (5).

A.4 Numerical and implementation conventions

Routing and entropy. The denominators in Eqs. (6) and (7) are floored at 10^{-6}. For TMR, the first frame uses the second frame as its reference, later frames use their predecessor, and a single-frame input has zero TMR. The retained fraction is bounded by $(\rho_{\min}, \rho_{\max}) = (0.30, 0.98)$; the token count is rounded and clipped to the valid range. The per-frame quota is the smaller of $\lfloor 0.8\rho_k HW \rfloor$ and the total budget divided by T, rounded down; the remaining positions are selected globally. For entropy, the implementation adds 10^{-9} to the total squared singular-value energy and includes only normalized energies above 10^{-9} in the logarithmic sum. Zero temporal energy gives zero entropy; fewer than three frames or a failed SVD also returns zero in the implementation.

Residual grid and local support. The implementation converts sampled tokens to residual codes and aligns them to the accumulation grid before residual correction. Computed-position and probe masks are lifted to that grid by nearest-neighbor resizing. The cap of 512 applies to computed Transformer-token probes before this lifting. The residual indices and sets in Sec. 3.4 refer to the aligned code grid. A radius-one neighborhood includes at most 26 valid neighbors and excludes the center, including when constructing a probe projector. The shared basis still uses all computed residuals. Local energy at or below 10^{-12} gives an empty support and $\mathbf{P}_i = \mathbf{0}$.

Calibration and degenerate cases. For online calibration, the denominator in Eq. (12) is floored at 10^{-12}, and an empty supported calibration set gives $\lambda_{k,r} = 0$. A zero projected candidate therefore contributes no residual update. The analysis-only optimum $\lambda_i^\star$ is defined as zero when its projected candidate is zero, since every scale then gives the same residual. The shadow path replaces probe logits in a copy of the main logit grid before sampling and code-grid alignment. Shadow sampling replays the main path's random-generator state and restores the post-sampling state, so calibration does not advance the production random stream. Online computed residuals are not same-prefix dense-reference targets.

A.5 Derivation and interpretation of the residual subspace fit

For an analyzed transfer position with $\|\mathbf{r}_i^\star\|_2 > 0$, $\mathbf{P}_i = \mathbf{U}_{\mathcal{S}_i}\mathbf{U}_{\mathcal{S}_i}^\top$ is an orthogonal projector, so $\mathbf{P}_i^\top = \mathbf{P}_i$ and $\mathbf{P}_i^2 = \mathbf{P}_i$. The dense-reference residual decomposes as

$$\mathbf{r}_i^\star = \mathbf{P}_i\mathbf{r}_i^\star + (\mathbf{I} - \mathbf{P}_i)\mathbf{r}_i^\star, \tag{15}$$

where the two terms are orthogonal. The Pythagorean identity therefore gives

$$\|\mathbf{r}_i^\star\|_2^2 = \|\mathbf{P}_i\mathbf{r}_i^\star\|_2^2 + \|\mathbf{r}_i^\star - \mathbf{P}_i\mathbf{r}_i^\star\|_2^2. \tag{16}$$

Dividing by $\|\mathbf{r}_i^\star\|_2^2$ yields $A_i = 1 - \frac{\|\mathbf{r}_i^\star - \mathbf{P}_i\mathbf{r}_i^\star\|_2^2}{\|\mathbf{r}_i^\star\|_2^2} = \frac{\|\mathbf{P}_i\mathbf{r}_i^\star\|_2^2}{\|\mathbf{r}_i^\star\|_2^2}$. Thus A_i is the fraction of dense-reference residual energy captured by the locally supported subspace: $A_i = 1$ means the residual lies entirely in that subspace, whereas $A_i = 0$ means it is orthogonal to it.

Expected fit of a uniformly oriented subspace. Fix m_i and a nonzero residual $\mathbf{r}_i^\star$. Let $\mathbf{Q}$ be uniformly distributed over the orthogonal group in $\mathbb{R}^D$, independently of the residual, and define

$$\mathbf{P}_{\mathrm{rand}} = \mathbf{Q}\,\mathrm{diag}(\mathbf{I}_{m_i}, \mathbf{0}_{D-m_i})\mathbf{Q}^\top. \tag{17}$$

This is a uniformly oriented rank-m_i orthogonal projector. For every fixed orthogonal matrix $\mathbf{O}$, rotational invariance gives $\mathbf{O}\,\mathbb{E}[\mathbf{P}_{\mathrm{rand}}]\,\mathbf{O}^\top = \mathbb{E}[\mathbf{P}_{\mathrm{rand}}]$. Consequently, $\mathbb{E}[\mathbf{P}_{\mathrm{rand}}] = c\mathbf{I}_D$ for some scalar c: sign changes force its off-diagonal entries to vanish, and coordinate permutations make its diagonal entries equal. Since every realization has trace m_i, taking traces gives $cD = m_i$. Using symmetry and idempotence of the projector,

$$\begin{aligned}\mathbb{E}\left[\frac{\|\mathbf{P}_{\mathrm{rand}}\mathbf{r}_i^\star\|_2^2}{\|\mathbf{r}_i^\star\|_2^2}\,\middle|\, m_i, \mathbf{r}_i^\star\right] &= \frac{(\mathbf{r}_i^\star)^\top\mathbb{E}[\mathbf{P}_{\mathrm{rand}}]\mathbf{r}_i^\star}{\|\mathbf{r}_i^\star\|_2^2} \\ &= \frac{m_i}{D}.\end{aligned} \tag{18}$$

Thus the dimension-matched reference for a collection of positions is the corresponding mean of m_i/D. Uniform orientation is essential: independence alone does not guarantee this expectation. This reference controls for the energy captured solely by using m_i dimensions.

In Figure 5(a), the larger gap at the first analyzed scale indicates that locally supported directions capture more dense-reference residual energy than a same-size random subspace; at finer scales the selected support broadens and the gap correspondingly narrows.

B ASTAR INFERENCE

Algorithm 1 summarizes clip-wise ASTAR inference. The accumulated code belongs to the current clip, and completed clips provide separate conditioning context. The guidance switch and sparse onset are independent. $\mathrm{ResidualDecode}$ includes the prediction head, APG when enabled, sampling, code conversion, and alignment to the accumulation grid, but not final VAE decoding. The correction therefore acts after grid alignment and before accumulation; no second upsampling of the corrected codes is needed. Superscript sh denotes the cheap calibration path. Grid lifting and numerical conventions are detailed in Appendix A.4.

Algorithm 1 ASTAR inference with Appearance Inheritance

Require: Text condition $\psi(y)$, N-clip schedule, L layers $\{\Phi^\ell\}$, guidance switch k_g, entropy probes/onsets, and parameters in Sec. 4.1.
Ensure: Video $\widehat{\mathbf{V}}$.

$(\mathbf{F}^1_{J_1}, \{\mathbf{a}^\ell_{k,r}\}, \mathcal{Z}) \leftarrow \mathrm{DenseFirstClip}(\psi(y))$
▷ Cache aligned first-clip states and retain conditioning context $\mathcal{Z}$.
for $c = 2, \dots, N$ **do**
  $\mathbf{F}^c_0 \leftarrow \mathrm{InitClip}(\mathcal{Z})$; $k_s \leftarrow k_{\mathrm{fallback}}$; $q \leftarrow 1$.
  for $j = 1, \dots, J_c$ in scheduled order **do**
    $(k, r) \leftarrow \mathrm{Schedule}(j)$; $\mathcal{C}_{k,r} \leftarrow \mathcal{I}_k$; $\mathcal{T}_{k,r}, \mathcal{B}_{k,r} \leftarrow \varnothing$.
    $g \leftarrow \mathbf{1}[k < k_g]$; $\mathbf{x}^0 \leftarrow \mathrm{BuildInput}(\mathbf{F}^c_{j-1}, \mathcal{Z}, \psi(y), g)$.
    ▷ $g = 1$: guided branches; $g = 0$: conditional branch.
    if $k \geq k_s$ **then**
      Compute s_i from $\mathbf{x}^0$ by Eqs. (6)–(7).
      $\mathcal{C}_{k,r} \leftarrow \mathrm{FrameBalancedTop}(\{s_i\}_{i \in \mathcal{I}_k}, \rho_k(q))$; $\mathcal{T}_{k,r} \leftarrow \mathcal{I}_k \setminus \mathcal{C}_{k,r}$.
      Select at most 512 computed boundary-token probes $\mathcal{B}_{k,r}$; $\mathbf{x}^{0,\mathrm{sh}}_{\mathcal{B}} \leftarrow \mathbf{x}^0_{\mathcal{B}}$.
      ▷ Keep the computed set fixed; use the aligned cache $\mathbf{a}^\ell = \mathbf{a}^\ell_{k,r}$.
      for $\ell = 0, \dots, L-1$ **do**
        $\mathbf{x}^{\ell+1}_{\mathcal{C}} \leftarrow \Phi^\ell_{\mathcal{C}}(\mathbf{x}^\ell, g, \mathcal{Z})$; $\mathbf{x}^{\ell+1}_i \leftarrow \mathbf{x}^\ell_i + \beta(\mathbf{a}^{\ell+1}_i - \mathbf{a}^\ell_i)$ for $i \in \mathcal{T}_{k,r}$.
        $\mathbf{x}^{\ell+1,\mathrm{sh}}_j \leftarrow \mathbf{x}^{\ell,\mathrm{sh}}_j + \beta(\mathbf{a}^{\ell+1}_j - \mathbf{a}^\ell_j)$ for $j \in \mathcal{B}_{k,r}$.
      end for
    else
      $\mathbf{x}^L \leftarrow \mathrm{DenseLayers}(\mathbf{x}^0, g, \mathcal{Z})$.
    end if
    Decode and align residual codes with $\mathrm{ResidualDecode}(\mathbf{x}^L, g)$.
    Lift $\mathcal{C}_{k,r}, \mathcal{T}_{k,r}, \mathcal{B}_{k,r}$ to the code grid; denote the candidates by $\mathbf{r}_i$ on $\mathcal{C}_{k,r}$ and $\widetilde{\mathbf{r}}_i$ on $\mathcal{T}_{k,r}$.
    if $\mathcal{T}_{k,r} \neq \varnothing$ **then**
      Eigendecompose $\mathbf{R}^\top_{\mathcal{C}} \mathbf{R}_{\mathcal{C}}$; form $\mathbf{P}_i$ for $i \in \mathcal{T}_{k,r} \cup \mathcal{B}_{k,r}$ by Eqs. (10)–(11), with center excluded.
      Decode and align shadow candidates with matched randomness; $\mathbf{p}^{\mathrm{probe}}_j \leftarrow \mathbf{P}_j \widetilde{\mathbf{r}}^{\mathrm{sh}}_j$.
      Compute $\lambda_{k,r}$ by Eq. (12) from supported probes; use 0 if none exist.
      $\widehat{\mathbf{r}}_i \leftarrow \lambda_{k,r} \mathbf{P}_i \widetilde{\mathbf{r}}_i$ for $i \in \mathcal{T}_{k,r}$ ($\mathbf{P}_i = \mathbf{0}$ without support); keep $\widehat{\mathbf{r}}_i = \mathbf{r}_i$ on $\mathcal{C}_{k,r}$.
    else
      $\widehat{\mathbf{r}}_i \leftarrow \mathbf{r}_i$ for every aligned code position $i \in \mathcal{C}_{k,r}$.
    end if
    $\widehat{\mathbf{z}}^c_j \leftarrow \mathrm{Assemble}(\{\widehat{\mathbf{r}}_i\})$; $\mathbf{F}^c_j \leftarrow \mathrm{Align}(\mathbf{F}^c_{j-1}) + \widehat{\mathbf{z}}^c_j$.
    if the last repetition of k_a has completed and $k_a < k_s$ **then**
      Compute H_a; if $H_a < \tau_a$, use the first branch of Eq. (14).
    else if the last repetition of k_b has completed and $k_b < k_s$ **then**
      Compute H_b; use the second or third branch of Eq. (14).
    end if
  end for
  $\mathcal{Z} \leftarrow \mathrm{UpdateContext}(\mathcal{Z}, \mathbf{F}^c_{J_c})$.
end for
$\widehat{\mathbf{V}} \leftarrow \mathrm{VAEDecode}(\{\mathbf{F}^c_{J_c}\}_{c=1}^N)$;
return $\widehat{\mathbf{V}}$.

ResidualCorrect. At each routed step, computed residuals define the shared eigenspace in Eq. (9). Each transfer position keeps the smallest locally supported direction set reaching η energy, giving $\mathbf{P}_i$. A bounded set of computed boundary probes supplies the online least-squares commitment scale

$\lambda_{k,r}$ in Eq. (12); the corrected candidate is then $\lambda_{k,r}\mathbf{P}_i\widetilde{\mathbf{r}}_i$. No dense target residual or oracle scale is used in production inference.

C Additional Experimental Details

C.1 Paired dense-reference fidelity

We compute PSNR, SSIM, and AlexNet-based LPIPS (Zhang et al., 2018) on paired RGB frames from ASTAR and dense InfinityStar at matching frame indices and native resolution. Scores are averaged over frames and videos.

C.2 Detailed VBench scores

Table 4 lists the 16 VBench dimensions.

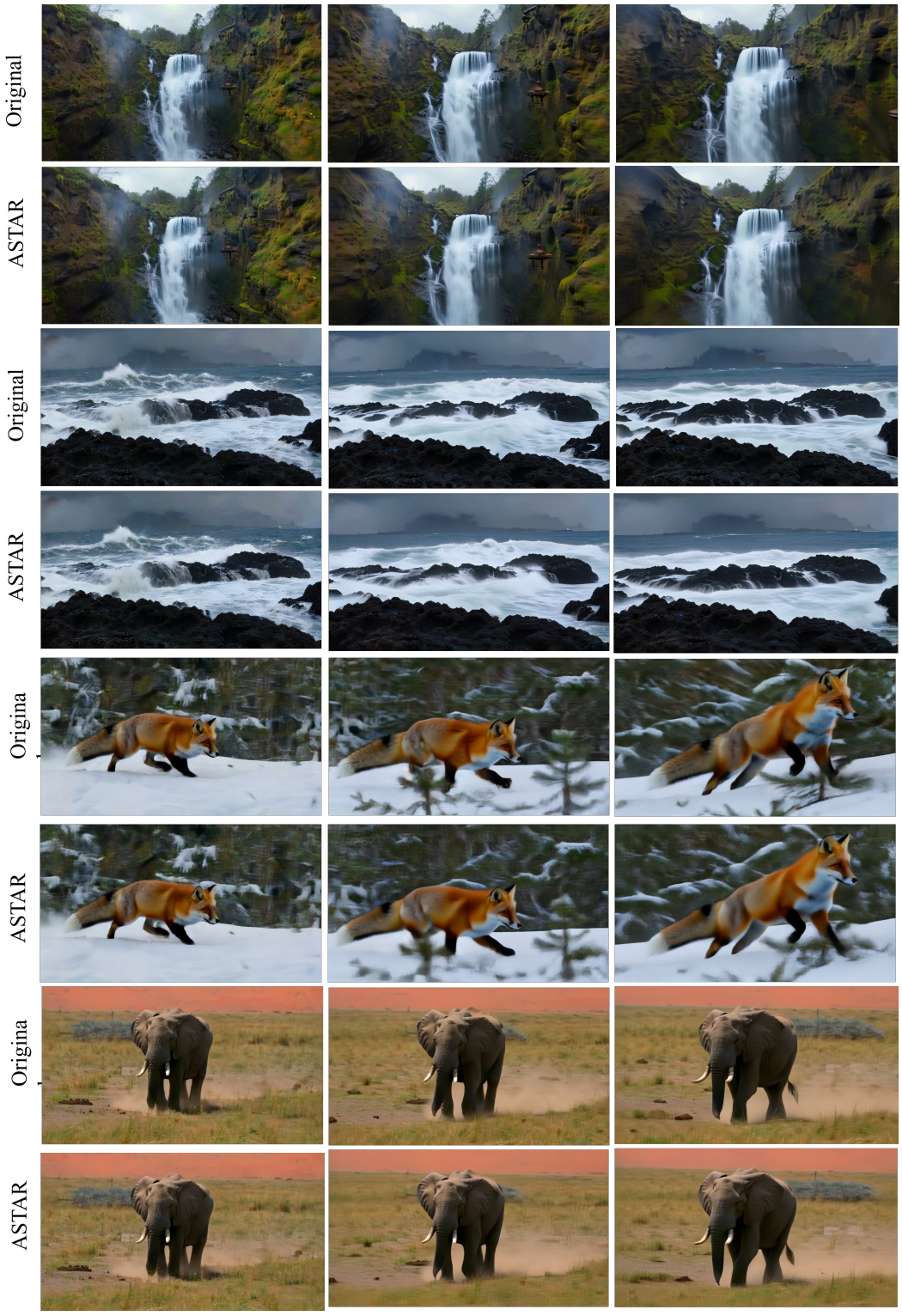
Table 4: VBench dimension scores for the final ASTAR configurations.

Dimension	720p ASTAR	480p ASTAR
Subject consistency	0.9472	0.9359
Background consistency	0.9558	0.9553
Temporal flickering	0.9860	0.9813
Motion smoothness	0.9886	0.9842
Dynamic degree	0.6139	0.7472
Aesthetic quality	0.6378	0.6562
Imaging quality	0.6653	0.6459
Object class	0.9628	0.9733
Multiple objects	0.8688	0.8732
Human action	0.9820	0.9760
Color	0.9103	0.8744
Spatial relationship	0.8493	0.8636
Scene	0.5403	0.5501
Appearance style	0.2075	0.2212
Temporal style	0.2544	0.2558
Overall consistency	0.2749	0.2761

C.3 QUALITATIVE EXAMPLES AT 720P, FIVE SECONDS

The following examples compare dense InfinityStar and ASTAR at 720p over five seconds using matched prompts and seeds.




Original
ASTAR
Original
ASTAR
Origina
ASTAR
Origina
ASTAR