FastSTAR: Spatiotemporal Token Pruning for Efficient Autoregressive Video Synthesis

Sungwoong Yune*, Suheon Jeong*, and Joo-Young Kim(✉)

Korea Advanced Institute of Science and Technology
{imwooong, sh.jeong, jooyoung1203}@kaist.ac.kr

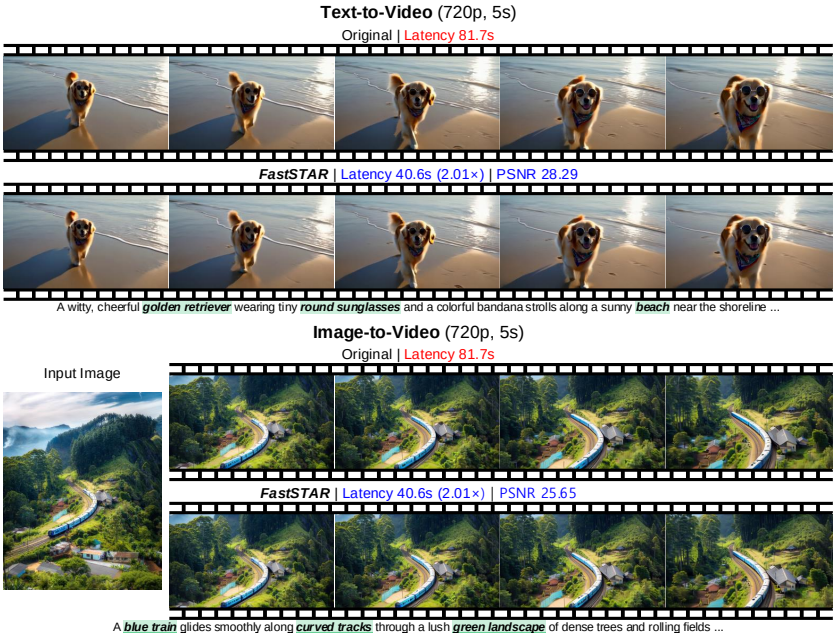


Fig. 1: FastSTAR achieves a 2.01× end-to-end speedup on a single H100 GPU for T2V and I2V tasks, maintaining high fidelity with PSNR of 28.29 and 25.65, respectively.

Abstract. Visual Autoregressive modeling (VAR) has emerged as a highly efficient alternative to diffusion-based frameworks, achieving comparable synthesis quality. However, as this paradigm extends to Space-time Autoregressive modeling (STAR) for video generation, scaling resolution and frame counts leads to a "token explosion" that creates a massive computational bottleneck in the final refinement stages. To address this, we propose **FastSTAR**, a training-free acceleration framework designed for high-quality video generation. Our core method, **Spatiotemporal Token Pruning**, identifies essential tokens by integrating two specialized terms: (1) *Spatial similarity*, which evaluates structural convergence across hierarchical scales to skip computations in regions where further refinement becomes redundant, and (2) *Temporal similarity*, which identifies active motion trajectories by assessing feature-level

* Equal contribution

variations relative to the preceding clip. Combined with a **Partial Update** mechanism, FastSTAR ensures that only non-converged regions are refined, maintaining fluid motion while bypassing redundant computations. Experimental results on InfinityStar demonstrate that FastSTAR achieves up to a 2.01× speedup with a PSNR of 28.29 and less than 1% performance degradation, proving a superior efficiency-quality trade-off for STAR-based video synthesis.

Keywords: Efficient Video Synthesis · Spacetime Autoregressive modeling · Training-free Acceleration · Token Pruning · Partial Update

1 Introduction

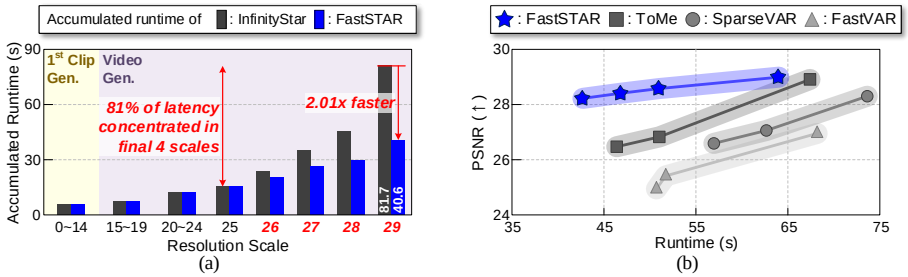


Fig. 2: (a) Latency breakdown for generating a 720p video (5s, 81 frames), comparing InfinityStar with FastSTAR. (b) Latency vs. PSNR trade-off curves for T2V synthesis. FastSTAR establishes the Pareto frontier, consistently surpassing existing baselines.

Visual Autoregressive modeling (VAR) [11, 31] has recently emerged as a powerful paradigm for visual generation, offering superior inference efficiency by redefining image synthesis as a coarse-to-fine next-scale prediction task. Unlike diffusion models [3–5, 27] that synthesize content through the iterative denoising of random noise [12, 23, 28], VAR-based frameworks progressively accumulate multi-scale residual feature maps. Building on this success, the recent Spacetime Autoregressive modeling (STAR) [22] utilizes a 3D-VAE [32] to tokenize video data into a spacetime pyramid, maintaining robust temporal consistency. However, the addition of the temporal dimension T escalates the computational complexity of the attention layer from $\mathcal{O}(H^2W^2)$ to $\mathcal{O}(T^2H^2W^2)$, where H and W denote the spatial resolution of the token map. Our profiling reveals that this quadratic growth manifests as a critical computational imbalance, where the last 4 out of the total resolution scales account for a disproportionate 81% of the total inference latency (Fig. 2(a)).

To address this "token explosion", existing reduction methods [1, 6, 10] suffer from distinct architectural limitations when applied to STAR. First, image-centric metric evaluations lead to the inaccurate identification of essential tokens, as they struggle to effectively detect the intricate temporal dynamics and motion trajectories inherent in video. Second, structural mismatches in token merging distort discrete feature distributions, triggering an error feedback loop that propagates across subsequent scales and spreads spatially as resolution increases. To

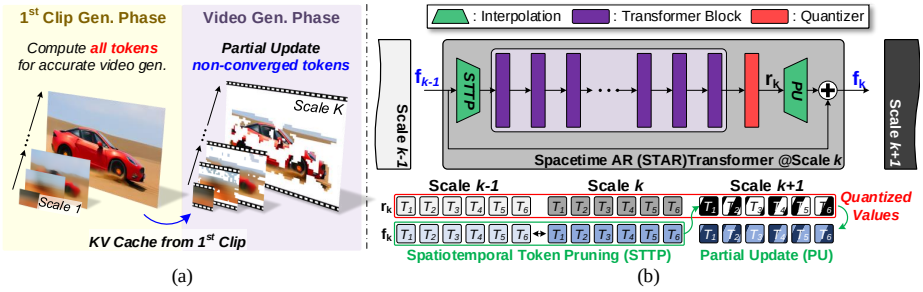


Fig. 3: (a) Overview of the FastSTAR framework. (b) Overall Mechanism of FastSTAR; Spatiotemporal Token Pruning identifies converged regions to completely skip transformer blocks, while Partial Update preserves structural integrity.

overcome these challenges, scale-wise spectral analysis (§ 3.2) demonstrates that while low-frequency global structures stabilize in the early stages, high-frequency details necessitate continuous updates until the final scale.

To leverage these insights, we propose **FastSTAR**, a training-free acceleration framework specifically tailored for high-quality video synthesis (Fig. 3). Unlike conventional token merging [1,20] or cached-token restoration [10] that often distort latent representations and degrade visual fidelity in the VAR paradigm, FastSTAR adopts a pruning-over-merging strategy to preserve structural integrity. Our framework consists of two core components:

- **Spatiotemporal Token Pruning (STTP):** It identifies non-converged tokens by integrating spatial and temporal similarity. The spatial term evaluates structural convergence across scales, while the temporal term captures motion dynamics via the causal consistency of STAR-based generation.
- **Partial Update (PU):** This mechanism prevents updates in pruned regions, confining transformer refinement exclusively to non-converged tokens. Restricting refinement to these active areas eliminates redundant processing while safeguarding the structural integrity of the cumulative feature map.

Experimental results on InfinityStar demonstrate that FastSTAR achieves an end-to-end speedup up to **2.01**× on a single NVIDIA H100 GPU while maintaining a high PSNR of 28.29. With less than 1% performance degradation, FastSTAR establishes a new Pareto frontier in Text-to-Video generation, consistently outperforming existing acceleration methods as shown in Fig. 2(b). By delivering robust gains across Image-to-Video and Video-to-Video tasks without any additional fine-tuning, our approach provides a practical, scalable solution for state-of-the-art autoregressive video synthesis.

2 Related Work

Autoregressive Video Generation. Early autoregressive (AR)-based video generators treat synthesis as sequential next-token prediction over discrete visual representations [34, 35, 38], which inherently discards spatial structure by

serializing 3D content into a 1D sequence. To address this, some approaches introduce spatiotemporal-aware positional embeddings [40] or remove vector quantization [39, 45] entirely with spatial set-by-set prediction [8]. Recent research also explores spatiotemporal cubes as a more tightly coupled prediction unit [26]. In parallel, VAR-based models reformulate video synthesis as a coarse-to-fine next-scale prediction [14, 22], achieving synthesis quality comparable to diffusion-based models [15, 36] at a fraction of the inference cost.

Efficient Image Synthesis. A diverse landscape of efficiency-oriented strategies has been established for both VAR-based and diffusion-based frameworks. Within the VAR-based framework, existing approaches reduce computational cost through (i) scale-wise token selection with cached-token restoration (FastVAR [10]), (ii) frequency-aware schemes that skip generation steps [16] and stage-aware schemes that prune or approximate computation in later stages [19], and (iii) low-frequency token exclusion in high-resolution stages while preserving fidelity via anchor tokens (SparseVAR [6]). Other directions improve efficiency without explicitly dropping tokens, such as compressing KV cache in a scale-aware manner [17], or modifying decoding to share computation across candidates [7]. In parallel, diffusion-based image synthesis has developed efficiency techniques: step-reduction via fast solvers [23], as well as runtime optimizations that reduce denoising compute through feature caching [47], similarity-based token pruning [42, 43], and diffusion process-aware token merging [2, 9, 24].

Efficient Video Synthesis. The transition from image to video synthesis significantly amplifies computational demands, necessitating specialized acceleration efforts across both autoregressive and diffusion paradigms. On the AR side, training-free methods exploit spatiotemporal redundancy during sequential decoding [37] or mitigate the growing KV-cache bottleneck via compact cache management [18]. For diffusion-based video synthesis, efficiency has been extensively studied through step-reduction via distillation or consistency training [21,33,41], as well as optimizations including feature caching across denoising steps [25, 46], token-level reduction via asymmetric attention pruning [30] and cross-frame token merging [20], and pruning of redundant temporal attention [29]. Despite their effectiveness, these accelerators are structurally mismatched with STAR. AR-based methods presuppose flat sequential decoding, such as compacting a linearly growing KV-cache [18] or parallelizing raster-scan generation [37], a structure absent in STAR's parallel, next-scale refinement. Diffusion-centric methods rely on cross-step redundancy within denoising trajectories [29, 30], a property absent in discrete-index generation, rendering them not directly transferable.

3 Method

3.1 Preliminary

Visual Autoregressive Modeling. Visual Autoregressive modeling (VAR) reformulates image synthesis as a next-scale prediction task. Within this framework, generation is performed through a hierarchy of discrete token maps $r_1, r_2,$

$\ldots, r_K$ with progressively increasing resolutions (h_k, w_k). Each token map r_k at a given scale follows a joint probability distribution conditioned on the results of all preceding scales:

$$p(r_1, \ldots, r_K) = \prod_{k=1}^{K} p(r_k \mid r_1, \ldots, r_{k-1}) \tag{1}$$

This hierarchical structure enables the model to establish global context at lower scales and refine intricate details at higher resolutions.

Spatiotemporal Pyramid Modeling. Following recent spacetime autoregressive frameworks, a video is decomposed into N sequential clips $\{c_1, c_2, \ldots, c_N\}$. The initial clip c_1 (at $T = 1$) encodes static appearance cues and establishes the spatiotemporal foundation for the sequence. Subsequent clips where $T > 1$ are modeled as 3D volume pyramid with dimensions (T, h_k, w_k) and are generated:

$$p(c_1, \ldots, c_N) = \prod_{c=1}^{N} \prod_{k=1}^{K} p(r_k^c \mid r_1^1, \ldots, r_{k-1}^c, \psi(t)) \tag{2}$$

where $\psi(t)$ represents the text conditioning.

Residual Feature Accumulation. At each scale k, the transformer predicts a distribution over tokens which is then quantized to produce the discrete token map r_k. This token map represents the residual information needed to refine the video at the current resolution. The feature map f_k is updated by integrating the interpolated results of the current token map r_k with the feature state from the previous scale f_{k-1} via element-wise addition:

$$f_k = \text{Interpolate}(r_k, (h_K, w_K)) + f_{k-1} \tag{3}$$

This recursive relationship ensures that f_k encapsulates the integrated information from all preceding scales, providing a stable foundation for further refinement. To further enhance quality, certain scales repeat this process multiple times at the same resolution before advancing to the next scale. The accumulated feature map is interpolated to the next scale resolution (h_{k+1}, w_{k+1}):

$$r_{k+1} = \text{Interpolate}(f_k, (h_{k+1}, w_{k+1})) \tag{4}$$

The resulting token map r_{k+1} then functions as the primary input for the transformer block at the subsequent scale $k + 1$.

3.2 Motivation

This work presents an acceleration framework for STAR-based video synthesis by exploiting the scale-wise information redundancy inherent in hierarchical video feature maps. Specifically, we characterize the model through the spectral convergence of information across resolutions, the spatiotemporal duality of feature maps, and the structural justification for prioritizing token pruning over merging in the context of cumulative refinement.

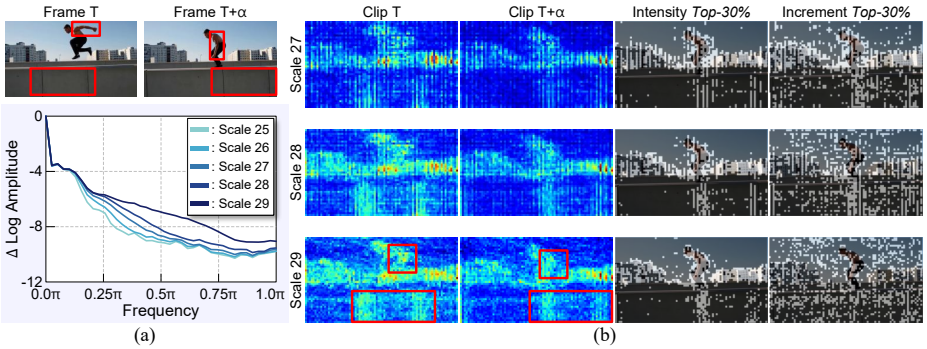


Fig. 4: (a) Scale-wise Fourier-based spectral analysis demonstrates early low-frequency convergence alongside sustained high-frequency updates. (b) High-frequency energy maps show intense localization. Comparing high-intensity locations with large incremental updates reveals significant opportunities for token pruning in converged regions.

Observation 1: Spectral Convergence of Video Feature Maps. Scale-specific information density in the STAR-based pipeline reveals a non-uniform distribution. Fourier analysis (Fig. 4a) demonstrates that low-frequency global structures converge early, while high-frequency details require continuous late-scale updates. Inverse Fourier Transform (IFT) heatmaps (Fig. 4b) confirm this high-frequency energy is intensely localized on salient regions. In contrast, residual increments between scales are scattered and patternless, failing to yield meaningful high-frequency refinements in low-frequency areas. This localized stability provides a significant opportunity to prune converged regions and concentrate computation exclusively on high-frequency areas.

Observation 2: Spatiotemporal Duality of Video Feature Maps. Unlike static images, tokens in the STAR framework embody a spatiotemporal duality, acting as integrated carriers for both decomposed visual textures and motion evolution. High-frequency tokens densely populate motion trajectories (Fig. 4b). Consequently, redundancy in STAR is fundamentally distinct from static image models. Pruning criteria must integrate both spatial and temporal characteristics to accurately identify tokens requiring active updates while bypassing spatiotemporally converged regions.

Observation 3: Structural Compatibility of Pruning over Merging. Because STAR accumulates discrete features via quantization, conventional token merging faces fundamental structural limitations. Averaging similar tokens distorts discrete latent representations, triggering a compounding error loop across scales (Fig. 5a). Quantitative analysis (Fig. 5b) demonstrates that merging suffers from a significantly higher Mean Squared Error (MSE) and rapidly expanding variance. Conversely, pruning preserves spatial integrity, maintains tighter error bounds, and ensures stable residual updates, structurally justifying its superiority for cumulative refinement.

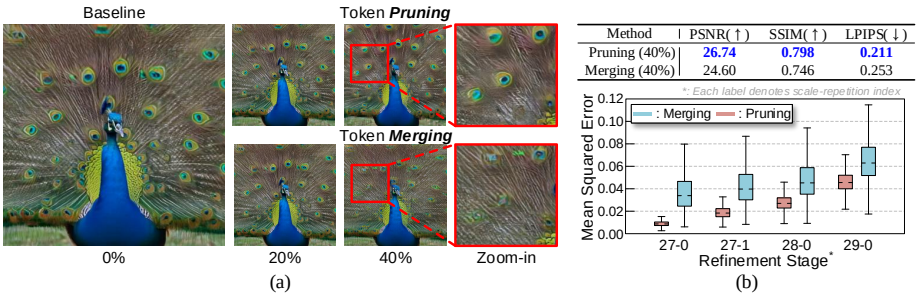


Fig. 5: Comparison of token pruning and merging. (a) Token merging at 40% obliterates high-frequency textures, while pruning preserves fine details even at high compression. (b) Merging exhibits higher average MSE and expanding variance across resolution scales, whereas pruning maintains tighter error bounds.

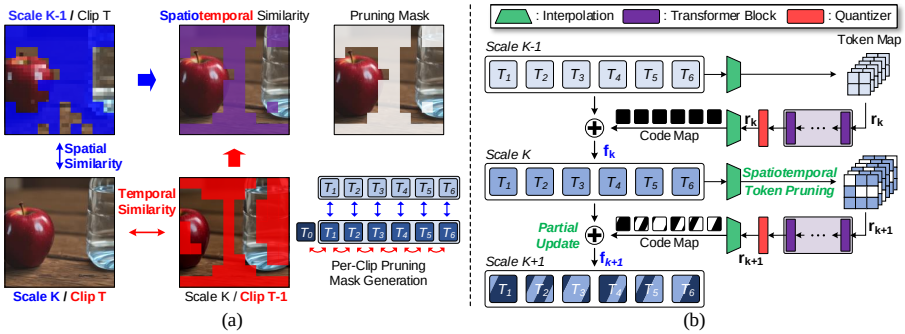


Fig. 6: Overview of Spatiotemporal Token Pruning (STTP) and Partial Update (PU). (a) Spatial similarity and temporal similarity are fused into a unified spatiotemporal score to generate a per-clip pruning mask. (b) Only high-priority tokens are passed through the transformer and quantize at each scale, while unselected regions are filled with zeros via Partial Update to preserve the integrity of the cumulative feature map.

3.3 Spatiotemporal Similarity-based Token Pruning and Partial Update

Building on the observed spectral convergence and spatiotemporal duality of video feature maps, we implement spatiotemporal similarity-based token pruning and a corresponding partial update mechanism to minimize redundant computations within the STAR transformer backbone (Fig. 6). This process ensures that only the most informative regions of the video feature map are refined while preserving the structural integrity of already converged regions.

Spatiotemporal Token Pruning (STTP). To identify tokens that exhibit persistent high-frequency energy and require continuous updates, we introduce a pruning methodology based on two distinct similarity metrics derived directly from the video feature maps. These metrics allow us to pinpoint regions of high

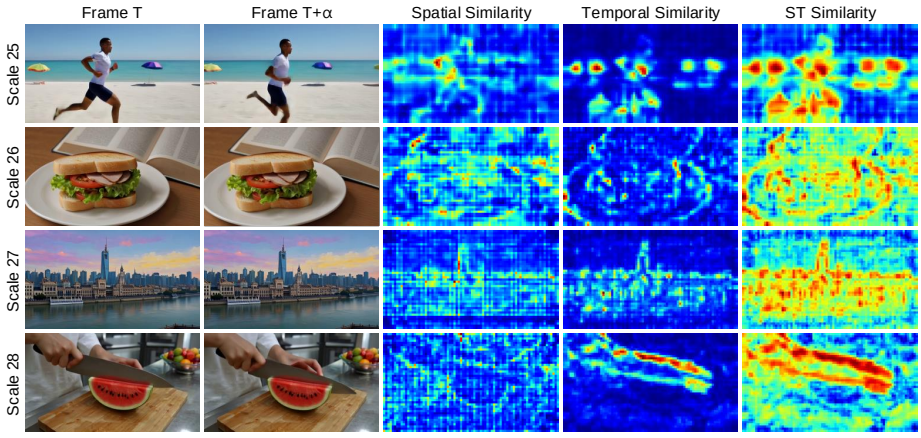


Fig. 7: Visualization of similarity maps across scales and frames. Warmer colors (red) indicate low similarity, corresponding to dynamic or high-frequency regions requiring update, while cooler colors (blue) indicate high similarity in converged, static regions.

informational gain with minimal computational overhead. Our strategy employs the inherent structural and motion-related information within the feature maps to guide the pruning process:

– **Spatial Similarity:** To detect spatial refinements across scales, we compute the cosine similarity between feature maps of the previous and current scales:

$$s_{t,k}^{\text{spatial}} = \frac{f_{t,k-1} \cdot f_{t,k}}{\|f_{t,k-1}\| \|f_{t,k}\|}, \quad t = 1, \ldots, T \tag{5}$$

As illustrated in Fig. 7, lower similarity scores are intensely localized on salient objects and detailed textures, indicating that these high-frequency regions undergo substantive updates to enhance visual fidelity. This allow us to identify non-converged tokens requiring further refinement while preserving already converged areas.

– **Temporal Similarity:** To exploit temporal continuity within the video feature map, we compute the cosine similarity between the feature maps of clip t and its predecessor $t-1$:

$$s_{t,k}^{\text{temporal}} = \frac{f_{t-1,k} \cdot f_{t,k}}{\|f_{t-1,k}\| \|f_{t,k}\|}, \quad t = 1, \ldots, T \tag{6}$$

As illustrated in Fig. 7, lower similarity scores are concentrated along motion trajectories, identifying regions that require temporal information to represent fluid motion. For the initial clip $f_{1,k}$, temporal importance is determined by comparing with the last scale feature map of the first clip $f_{0,K}$.

Joint Metric Fusion via ℓ_p-norm Dissimilarity. To robustly integrate these indicators into a unified score, we transform the cosine similarity values into a dissimilarity-based importance metric; lower similarity signifies a higher refinement priority. The fusion is defined as a ℓ_p-norm over the two dissimilarity terms:

$$\text{Score}_{ST} = [(1 - \text{s}_{\text{spatial}})^p + (1 - \text{s}_{\text{temporal}})^p]^{1/p} \tag{7}$$

where p modulates the sensitivity to high-dissimilarity regions. As demonstrated in our ablation studies (§ 4.5), the fusion mechanism is highly robust to variations in p. We adopt $p = 2$ as our default configuration to ensure a more gradual score distribution, facilitating stable transitions for content-adaptive pruning.

Implementation of Partial Update (PU). To ensure the pruning mask $M \in \{0, 1\}^{h_K \times w_K}$ is compatible with the hierarchical structure of VAR, we downsample the generated pruning mask to match the current token map dimensions (h_k, w_k) via interpolation. The complete per-scale procedure is summarized in Algorithm 1. Algorithmically, the binary mask M is upsampled via trilinear interpolation with a strict threshold ($>$0.5) to ensure no fractional values enter the residual map. STTP drops the pruned tokens before the transformer blocks. The gathered active tokens retain their original Rotary Position Embedding (RoPE) indices, keeping spatial structures aligned. The pruning process for the input token map r_k is defined as follows:

$$\hat{r}_k = r_k \odot \text{Interpolate}(M, (h_k, w_k)) \tag{8}$$

The transformer backbone subsequently processes only the selected tokens $\hat{r}_k$, significantly reducing the computational overhead. After the transformer iterations and quantization are complete, the resulting residual code map is upsampled back to the video feature map resolution (h_K, w_K). To maintain the structural integrity of the cumulative VAR process, we implement a partial update strategy. The unselected positions are explicitly zero-padded using the original pruning mask M before being integrated into the previous video feature map f_{k-1}. $\hat{M}$ is a trilinear interpolated binary mask, upsampled to fit the next scale. The partial update for the video feature map f_k is formulated as:

$$f_k = \text{Interpolate}(\hat{r}_k, (h_K, w_K)) \odot \hat{M} + f_{k-1} \tag{9}$$

Algorithm 1 FastSTAR step at scale k (pruned scales).

Require: previous cumulative map f_{k-1}, binary mask M_k

$r_k \leftarrow \textsc{Embed}(f_{k-1})$ ▷ full-shape input
$\hat{r}_k \leftarrow r_k[:, \textsc{flatten}(M_k), :]$ ▷ **STTP**: drop pruned tokens
$\hat{r}_k \leftarrow \textsc{TransformerBlocks}(\hat{r}_k)$ ▷ RoPE on original positions
$\hat{r}_k \leftarrow \textsc{Sample}(\hat{r}_k)$
$\hat{r}_k \leftarrow 0$; $\hat{r}_k[:, \textsc{flatten}(M_k), :] \leftarrow \hat{r}_k$ ▷ zero-padding
$\hat{r}_k \leftarrow \textsc{Interp}(\textsc{IndToCodes}(\hat{r}_k)$, target res.)
$\hat{M}_k \leftarrow (\textsc{TrilinearUp}(M_k) > 0.5)$ ▷ binary upsample
$\hat{r}_k \leftarrow \hat{r}_k \odot \hat{M}_k$ ▷ **PU**: zero-fill at pruned positions
$f_k \leftarrow f_{k-1} + \hat{r}_k$
return f_k

This mechanism suppresses cumulative errors and prevents uncomputed noise from polluting converged regions, ensuring that computational resources are strictly concentrated on the most dynamic and detailed portions of the video.

4 Experiments

4.1 Experimental Setup

Models and Baselines. We evaluate the proposed FastSTAR using InfinityStar [22] as the backbone. To ensure a comprehensive comparison, we implemented representative training-free token reduction baselines: SparseVAR [6], FastVAR [10], and ToMe [1]. Since these methods were originally designed for text-to-image synthesis or non-autoregressive architectures, we extended them to accommodate the temporal dimension by applying their respective reduction mechanisms across the spatiotemporal pyramid. This ensures a fair and competitive evaluation against established image-based acceleration strategies within the video domain. For all experiments, we maintain the default hyperparameters of the backbone model to ensure consistency. While other video acceleration methods exist for sequential decoding (e.g., PackCache [18]) or diffusion trajectories (e.g., AsymRnR [30]), their underlying mechanisms are structurally mismatched for STAR's parallel scale refinement.

Metrics. We assess our framework across two primary dimensions: inference efficiency and generation quality.

- **Inference Efficiency:** We measure the practical acceleration of our method on a NVIDIA H100 GPU (80GB). Efficiency metrics include end-to-end inference latency across the text encoder and VAE decoder, and the achieved speedup.
- **Generation Quality:** We utilize the VBench [13] suite, evaluating 5-second videos (81 frames) at 720p resolution across all 16 dimensions using the official VBench prompt set. To ensure a fair comparison, we employ the same rewritten prompts used in the evaluation of InfinityStar. Additionally, we report Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) [44] to quantify reconstruction fidelity and perceptual similarity. These metrics are computed on 10 videos sampled per VBench dimension for each task (T2V, I2V, and V2V).

Implementations. For the spatiotemporal similarity-based pruning, we apply STTP to the high-resolution refinement stages, which account for the vast majority of the total inference latency. Specifically, we focus on the final four scales to exploit cross-scale and intra-scale redundancies during the synthesis process, alleviating the computational burden while preserving the established structural layouts. For scales with multiple iterations at the same resolution, pruning is applied to early iterations, reserving the final iteration for full-token refinement.

Table 1: Quantitative and efficiency benchmarking for 720p, 5s (81 frames) Text-to-Video generation. We compare FastSTAR with baselines across quality metrics and inference latency; **best results** are shown in bold and second-best results are underlined; † denotes results using prompt rewriting.

Methods	Config	Inference Efficiency		Quality Metrics			VBench Score (↑)		
		Latency (s)	Speedup	PSNR (↑)	SSIM (↑)	LPIPS (↓)	Quality Score	Semantic Score	Total Score
InfinityStar†	*8B, 720p, T2V*	81.7	1.00×	-	-	-	84.33	82.11	83.89
●	+ SparseVAR	56.9	1.44×	26.59	0.789	0.228	82.70	79.86	82.13
●	+ FastVAR	50.7	1.61×	25.00	0.757	0.266	81.20	81.18	81.19
●	+ ToMe	46.4	1.76×	26.47	0.790	0.227	81.98	80.10	81.60
●	**+ FastSTAR**	42.6	1.92×	**28.30**	**0.828**	**0.184**	**83.71**	79.96	82.96
◆	**+ FastSTAR**	40.6	2.01×	28.29	0.828	0.185	83.59	**81.51**	**83.18**

● denotes pruning/merging ratio of scale (26, 27, 28, 29) = (20%, 30%, 50%, 60%), ◆ denotes pruning ratio of scale (26, 27, 28, 29) = (20%, 30%, 40%, 70%)

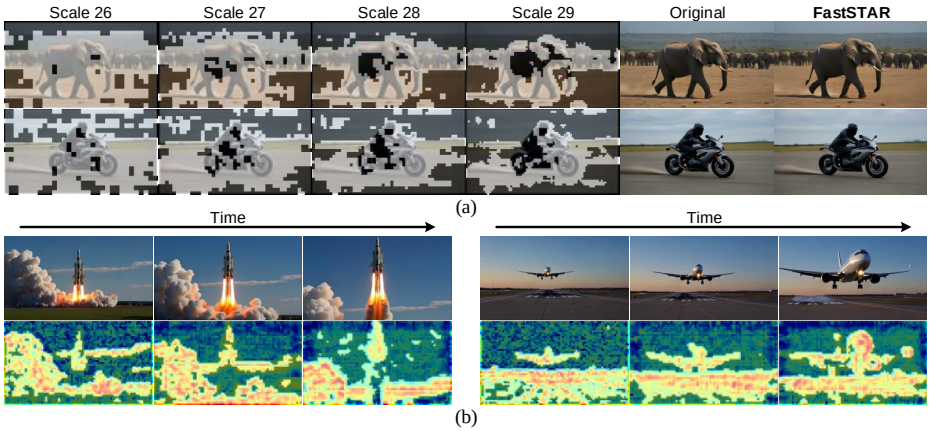


Fig. 8: Qualitative results. (a) Pruning masks across scales and resulting outputs. (b) Frame-level results at the last scale with spatiotemporal similarity heatmaps and pruning masks. Black denotes pruned regions and white denotes computed regions.

4.2 Comparative Analysis on Text-to-Video Generation

Quantitative Comparison. Table 1 compares the inference efficiency and generation quality of various acceleration schemes applied to the InfinityStar baseline for T2V. Although the base InfinityStar model offers high-fidelity video synthesis, its original inference time of 81.7s imposes a significant computational burden for 720p generation. While existing reduction methods like SparseVAR, FastVAR, and ToMe yield speedups of 1.44×, 1.61×, and 1.76×, respectively, they suffer substantial declines in both VBench scores and Quality Metrics, including PSNR, SSIM, and LPIPS. This degradation stems from their inherent limitations in handling the hierarchical and temporal complexities of STAR. In contrast, FastSTAR maintains high generation quality across different pruning configurations. Specifically, for two configurations with similar pruning ratios—(20%, 30%, 50%, 60%) and (20%, 30%, 40%, 70%)—our method achieves acceleration up to 2.01× with a negligible VBench score reduction of only 0.85%

Table 2: Comparison with state-of-the-art token reduction methods on 720p, 5s (81 frames) Image-to-Video generation; **best results** are shown in bold and second-best results are underlined.

Methods	Config	Inference Efficiency		Quality Metrics			VBench Score (↑)		
		Latency (s)	Speedup	PSNR (↑)	SSIM (↑)	LPIPS (↓)	V-I* Subject Consistency	V-I* Bgnd** Consistency	Quality Score
InfinityStar	*8B, 720p, I2V*	81.7	1.00×	-	-	-	97.54	97.37	80.97
	◆ + SparseVAR	55.3	1.56×	23.86	0.754	0.249	96.40	97.05	78.30
	◆ + FastVAR	44.9	1.82×	22.20	0.711	0.292	94.46	95.49	74.83
	◆ + ToMe	52.0	1.57×	23.93	0.757	0.248	96.45	97.06	78.25
	◆ **+ FastSTAR**	40.6	2.01×	**25.65**	**0.809**	**0.195**	**97.24**	**97.29**	**80.01**

◆ denotes pruning ratio of scale (26, 27, 28, 29) = (20%, 30%, 40%, 70%) * : V-I denotes Video-Image ** : Bgnd denotes Background

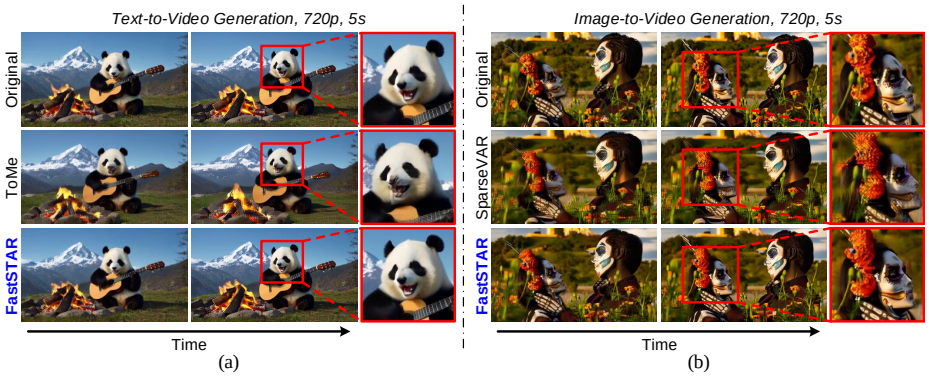


Fig. 9: Qualitative comparison of generation quality. (a) T2V. (b) I2V.

from the baseline. Furthermore, our method significantly outperforms all other baselines in quality metrics, achieving the highest PSNR of 28.29.

Qualitative Comparison. Fig. 8 provides a qualitative visualization of how our STTP accurately identifies redundant regions. As illustrated in Fig. 8(a), as the spatial scale increases, FastSTAR effectively captures areas where further updates are no longer required, spanning both static backgrounds and converged regions within dynamic objects. Furthermore, Fig. 8(b) displays generated video frames overlaid with our fused similarity score heatmaps and the resulting pruning masks. Because temporal characteristics are natively fused into the importance scores, the heatmaps clearly demonstrate that FastSTAR preserves critical motion trajectories while aggressively pruning spatiotemporally redundant tokens, ensuring fluid motion and structural integrity.

4.3 Comparative Analysis on Image-to-Video Synthesis

Quantitative Comparison. We evaluate the performance of FastSTAR against established token reduction baselines, including SparseVAR, FastVAR, and ToMe, as shown in Table 2. For a fair comparison, we align the pruning and merging ratios at $(20\%, 30\%, 40\%, 70\%)$ across the final scales of the 720p I2V task. While

Table 3: Performance across 480p STAR-based synthesis tasks (T2V, I2V, V2V).

Methods	Config			Inference Efficiency		Quality Metrics		
	Task	Resolution	Frames	Latency (s)	Speedup	PSNR (↑)	SSIM (↑)	LPIPS (↓)
InfinityStar	*Text-to-Video*	480p	81	38.3	1.00×	-	-	-
■ + FastSTAR				26.1	**1.47×**	22.96	0.697	0.262
InfinityStar	*Image-to-Video*	480p	81	38.0	1.00×	-	-	-
■ + FastSTAR				25.9	**1.47×**	23.86	0.750	0.234
InfinityStar	*Video-to-Video*	480p	81	140.2	1.00×	-	-	-
■ + FastSTAR				101.1	**1.39×**	24.93	0.767	0.219

■ denotes pruning ratio of scale (24, 25, 26, 27) = (20%, 30%, 40%, 70%)

all methods achieve speedups, FastSTAR significantly outperforms the baselines in maintaining quality metrics, reporting superior scores in PSNR of 25.65.

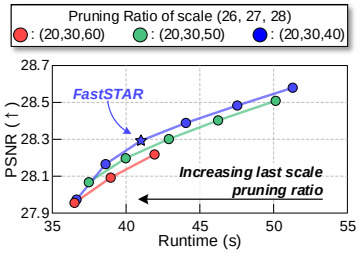
Qualitative Comparison. Fig. 9 compares the generation quality across different tasks and acceleration methods. As illustrated in the T2V task in Fig. 9(a), FastSTAR maintains nearly identical visual quality to the original baseline, preserving intricate textures and structural details that are significantly degraded in ToMe. Similarly, Fig. 9(b) demonstrates that in the I2V task, FastSTAR effectively avoids the severe artifacts and loss of semantic clarity seen in SparseVAR. By accurately identifying redundant regions through spatiotemporal awareness, FastSTAR ensures superior structural integrity and visual consistency, even under the aggressive pruning ratios required for 2.01× acceleration.

4.4 Robustness Across Various Synthesis Tasks

To demonstrate the versatility of our framework, we apply FastSTAR to various synthesis tasks at 480p resolution, including T2V, I2V, and V2V. As summarized in Table 3, our method consistently achieves acceleration across all these tasks while maintaining high reconstruction fidelity. Notably, the acceleration gain at 480p is slightly lower than at 720p due to the reduced scale count and smaller token map dimensions. These results confirm robustness across different conditioning signals and resolutions.

4.5 Ablation Studies

Different Pruning Ratios. The pruning ratio dictates the balance between computational efficiency and generation quality. As shown in Fig. 10(a), increasing the pruning ratio in the final scales results in a direct reduction in runtime, albeit with an expected trade-off in reconstruction fidelity. Through our analysis across scales 26, 27, and 28, we observe that configurations of $(20\%, 30\%, 40\%)$ maintain a stable quality-to-speed slope. Specifically, for the final scale (scale 29), we find that quality remains robust up to a 70% pruning ratio; however, exceeding this threshold leads to severe quality degradation. Consequently, we adopt the $(20\%, 30\%, 40\%, 70\%)$ configuration as our primary setting to maximize speedup while preserving structural integrity.



(a)

Norm Degree (p)		PSNR (↑)	SSIM (↑)	LPIPS (↓)
1	Addition	27.10	0.805	0.204
2	Euclidean	27.14	**0.805**	**0.203**
∞	Maximum	**27.14**	0.805	0.203

(b)

Pruning Method	PSNR (↑)	SSIM (↑)	LPIPS (↓)
Random	25.98	0.776	0.236
Spatial	26.92	0.803	0.204
Spatiotemporal	**27.14**	**0.805**	**0.203**

(c)

Fig. 10: Ablation studies of FastSTAR. (a) Effect of pruning ratios on runtime and PSNR. (b) Sensitivity to norm degree p in score fusion. (c) Comparison of pruning methods. **Best results** are shown in bold and second-best results are underlined.

Methods	Config			Inference Efficiency		Quality Metrics		
	Mode	Resolution	Frames	Latency (s)	Speedup	PSNR (↑)	SSIM (↑)	LPIPS (↓)
InfinityStar	*T2V / I2V*	720p	81	81.7	1.00×	-	-	-
(a) Partial Update (PU) Ablation for T2V								
◆ + STTP Only	*Text-to-Video*	720p	81	40.6	2.01×	25.12	0.777	0.327
(b) Content-Adaptive Pruning for I2V								
▲ + Dynamic (HC^{*})				48.3	1.69×	24.33	0.785	0.215
■ + Dynamic (LC^{**})	*Image-to-Video*	720p	81	35.6	2.30×	28.13	0.846	0.154
◆ + Static (HC^{*}+LC^{**})				40.6	2.01×	26.94	0.829	0.177

▲ denotes average pruning ratio of scale (26, 27, 28, 29) = (23.3%, 31.0%, 37.4%, 47.2%)
■ denotes average pruning ratio of scale (26, 27, 28, 29) = (34.3%, 49.9%, 61.5%, 74.8%)
◆ denotes pruning ratio of scale (26, 27, 28, 29) = (20.0%, 30.0%, 40.0%, 70.0%)
HC^{*} : High-Complexity Video, LC^{**} : Low-Complexity Video

Fig. 11: Ablation studies of FastSTAR. (a) Partial Update (PU) ablation for T2V: removing PU degrades quality due to error accumulation at pruned positions. (b) Content-adaptive pruning for I2V, where the dynamic ratio adapts to video complexity.

Norm Degree. To analyze the sensitivity of our spatiotemporal score fusion to the choice of norm degree p, we compare $p \in \{1, 2, \infty\}$ in Fig. 10(b). The results demonstrate a key robustness property: the fusion mechanism is highly insensitive to the choice of p, with all variations yielding competitive and stable generation quality. We adopt $p = 2$ as our default configuration because it acts as a balanced integrator between the maximum behavior of $p = \infty$ and the uniform weighting of $p = 1$. Furthermore, this choice yields a smoother score distribution, which facilitates an extension to a dynamic, content-adaptive pruning mechanism.

Pruning Method. To justify the necessity of our spatiotemporal approach, we compare its performance against random and purely spatial-based pruning strategies in Fig. 10(c). Random pruning fails to preserve critical information, resulting in poor quality across all dimensions. While spatial-only pruning offers an improvement, it lacks the temporal context required to maintain fluid motion and consistency. FastSTAR surpasses both baselines across the entire spectrum of metrics, establishing that integrating temporal dynamics into the pruning method is essential for high-fidelity video synthesis.

Effectiveness of Partial Update (PU). To validate the necessity of PU, we evaluate FastSTAR without it. As shown in Table 11(a), removing PU drops the PSNR to 25.12 (T2V), as cumulative errors at pruned positions introduce severe grid artifacts during residual accumulation. This confirms that PU is essential for maintaining the structural integrity of the cumulative feature map.

Content-Adaptive Pruning While our static ratios ensure reproducibility, the ST metric naturally supports content-adaptive pruning. Video complexity was analyzed using the VBench I2V samples, focusing on optical flow and Canny edge density. We find that high-complexity videos exhibit a lower 50th-to-95th percentile ratio (p_{50}/p_{95}) of ST scores, reflecting wider variance; across the final scales, low-complexity videos show a 1.78× higher median ratio. We thus define a dynamic per-scale ratio $R(s) = (B+\Delta s)(1+\alpha\, p_{50}/p_{95})$, where B, Δ, and α are the base ratio, scale increment, and pruning strength. As shown in Table 11(b), this prunes conservatively in complex videos to preserve fidelity (PSNR 24.33) while pruning up to 74.8% in simple videos (PSNR 28.13), confirming the ST metric as a robust content-adaptive indicator.

5 Conclusion

In this work, we presented FastSTAR, a training-free acceleration framework specifically designed for high-quality video generation within the Spacetime AutoRegressive modeling (STAR). Our core innovation, Spatiotemporal Token Pruning (STTP), addresses the "token explosion" at final refinement scales by integrating spatial structural convergence and temporal motion trajectories to identify high-priority tokens. Unlike conventional merging strategies that distort discrete feature distributions, our pruning-over-merging approach, combined with a Partial Update (PU) mechanism, effectively prevents error propagation while bypassing redundant computations.

Experimental results on the InfinityStar model demonstrate that FastSTAR achieves a significant end-to-end speedup of up to 2.01× on a single NVIDIA H100 GPU for 720p video synthesis. Remarkably, this acceleration is achieved with a negligible VBench score degradation of less than 1%, consistently delivering robust performance across various generation modes including T2V, I2V, and V2V. By defining a new Pareto frontier for efficient video synthesis, FastSTAR provides a practical and scalable solution for high-resolution autoregressive modeling in real-world applications.

Acknowledgements

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (IITP-2026-RS-2020-II201847, Information Technology Research Center; IITP-2026-RS-2023-00256472, Graduate School of Artificial Intelligence Semiconductor)

References

1. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your vit but faster (2023), https://arxiv.org/abs/2210.09461
2. Bolya, D., Hoffman, J.: Token merging for fast stable diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 4599–4603 (June 2023)
3. Chen, J., Ge, C., Xie, E., Wu, Y., Yao, L., Ren, X., Wang, Z., Luo, P., Lu, H., Li, Z.: Pixart-σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation (2024), https://arxiv.org/abs/2403.04692
4. Chen, J., Wu, Y., Luo, S., Xie, E., Paul, S., Luo, P., Zhao, H., Li, Z.: Pixart-δ: Fast and controllable image generation with latent consistency models (2024), https://arxiv.org/abs/2401.05252
5. Chen, J., Yu, J., Ge, C., Yao, L., Xie, E., Wu, Y., Wang, Z., Kwok, J., Luo, P., Lu, H., Li, Z.: Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis (2023), https://arxiv.org/abs/2310.00426
6. Chen, Z., Fan, J., Yu, Z., Zhuang, B., Tan, M.: Frequency-aware autoregressive modeling for efficient high-resolution image synthesis (2025), https://arxiv.org/abs/2507.20454
7. Chen, Z., Ma, X., Fang, G., Wang, X.: Collaborative decoding makes visual autoregressive modeling efficient. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 23334–23344 (June 2025)
8. Deng, H., Pan, T., Diao, H., Luo, Z., Cui, Y., Lu, H., Shan, S., Qi, Y., Wang, X.: Autoregressive video generation without vector quantization (2025), https://arxiv.org/abs/2412.14169
9. Fang, H., Tang, S., Cao, J., Zhang, E., Tang, F., Lee, T.Y.: Attend to not attended: Structure-then-detail token merging for post-training dit acceleration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 18083–18092 (June 2025)
10. Guo, H., Li, Y., Zhang, T., Wang, J., Dai, T., Xia, S.T., Benini, L.: Fastvar: Linear visual autoregressive modeling via cached token pruning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 19011–19021 (October 2025)
11. Han, J., Liu, J., Jiang, Y., Yan, B., Zhang, Y., Yuan, Z., Peng, B., Liu, X.: Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15733–15744 (June 2025)
12. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models (2020), https://arxiv.org/abs/2006.11239
13. Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., Wang, Y., Chen, X., Wang, L., Lin, D., Qiao, Y., Liu, Z.: Vbench: Comprehensive benchmark suite for video generative models (2023), https://arxiv.org/abs/2311.17982
14. Ji, L., Liu, X., Shang, J., Wang, S., Sun, Y., Wu, H., Wang, H.: Videoar: Autoregressive video generation via next-frame & scale prediction (2026), https://arxiv.org/abs/2601.05966
15. Kong, W., Tian, Q., Zhang, Z., Min, R., Dai, Z., Zhou, J., Xiong, J., Li, X., Wu, B., Zhang, J., Wu, K., Lin, Q., Yuan, J., Long, Y., Wang, A., Wang, A., Li, C., Huang, D., Yang, F., Tan, H., Wang, H., Song, J., Bai, J., Wu, J., Xue, J., Wang, J., Wang, K., Liu, M., Li, P., Li, S., Wang, W., Yu, W., Deng, X., Li, Y., Chen,

Y., Cui, Y., Peng, Y., Yu, Z., He, Z., Xu, Z., Zhou, Z., Xu, Z., Tao, Y., Lu, Q., Liu, S., Zhou, D., Wang, H., Yang, Y., Wang, D., Liu, Y., Jiang, J., Zhong, C.: Hunyuanvideo: A systematic framework for large video generative models (2025), https://arxiv.org/abs/2412.03603
16. Li, J., Ma, Y., Zhang, X., Wei, Q., Liu, S., Zhang, L.: Skipvar: Accelerating visual autoregressive modeling via adaptive frequency-aware skipping (2025), https://arxiv.org/abs/2506.08908
17. Li, K., Chen, Z., Yang, C.Y., Hwang, J.N.: Memory-efficient visual autoregressive modeling with scale-aware kv cache compression (2025), https://arxiv.org/abs/2505.19602
18. Li, K., Shah, M., Shang, Y.: Packcache: A training-free acceleration method for unified autoregressive video generation via compact kv-cache (2026), https://arxiv.org/abs/2601.04359
19. Li, S., Wang, K., Khan, S., Khan, F.S., Yang, J., Wang, Y.: Stagevar: Stage-aware acceleration for visual autoregressive models (2025), https://arxiv.org/abs/2512.16483
20. Li, X., Ma, C., Yang, X., Yang, M.H.: Vidtome: Video token merging for zero-shot video editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7486–7495 (June 2024)
21. Lin, S., Yang, X.: Animatediff-lightning: Cross-model diffusion distillation (2024), https://arxiv.org/abs/2403.12706
22. Liu, J., Han, J., Yan, B., Wu, H., Zhu, F., Wang, X., Jiang, Y., Peng, B., Yuan, Z.: Infinitystar: Unified spacetime autoregressive modeling for visual generation (2025), https://arxiv.org/abs/2511.04675
23. Lu, C., Zhou, Y., Bao, F., Chen, J., LI, C., Zhu, J.: Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) Advances in Neural Information Processing Systems. vol. 35, pp. 5775–5787. Curran Associates, Inc. (2022), https://proceedings.neurips.cc/paper_files/paper/2022/file/260a14acce2a89dad36adc8eefe7c59e-Paper-Conference.pdf
24. Lu, W., Zheng, S., Xia, Y., Wang, S.: Toma: Token merge with attention for diffusion models (2025), https://arxiv.org/abs/2509.10918
25. Lv, Z., Si, C., Song, J., Yang, Z., Qiao, Y., Liu, Z., Wong, K.Y.K.: Fastercache: Training-free video diffusion model acceleration with high quality (2025), https://arxiv.org/abs/2410.19355
26. Ren, S., Chen, C., Wang, Z., Song, L., Zhu, X., Yuille, A., Yang, Y., Lu, J.: Autoregressive video generation beyond next frames prediction (2025), https://arxiv.org/abs/2509.24081
27. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models (2022), https://arxiv.org/abs/2112.10752
28. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models (2022), https://arxiv.org/abs/2010.02502
29. Su, S., Liu, J., Gao, L., Song, J.: F^3-pruning: A training-free and generalized pruning strategy towards faster and finer text-to-video synthesis. Proceedings of the AAAI Conference on Artificial Intelligence **38**(5), 4961–4969 (Mar 2024). https://doi.org/10.1609/aaai.v38i5.28300, https://ojs.aaai.org/index.php/AAAI/article/view/28300
30. Sun, W., Tu, R.C., Liao, J., Jin, Z., Tao, D.: Asymrnr: Video diffusion transformers acceleration with asymmetric reduction and restoration. In: International Conference on Machine Learning. pp. 57694–57711. PMLR (2025)

31. Tian, K., Jiang, Y., Yuan, Z., Peng, B., Wang, L.: Visual autoregressive modeling: Scalable image generation via next-scale prediction. In: Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., Zhang, C. (eds.) Advances in Neural Information Processing Systems. vol. 37, pp. 84839–84865. Curran Associates, Inc. (2024). https://doi.org/10.52202/079017-2694, https://proceedings.neurips.cc/paper_files/paper/2024/file/9a24e284b187f662681440ba15c416fb-Paper-Conference.pdf
32. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., Zeng, J., Wang, J., Zhang, J., Zhou, J., Wang, J., Chen, J., Zhu, K., Zhao, K., Yan, K., Huang, L., Feng, M., Zhang, N., Li, P., Wu, P., Chu, R., Feng, R., Zhang, S., Sun, S., Fang, T., Wang, T., Gui, T., Weng, T., Shen, T., Lin, W., Wang, W., Wang, W., Zhou, W., Wang, W., Shen, W., Yu, W., Shi, X., Huang, X., Xu, X., Kou, Y., Lv, Y., Li, Y., Liu, Y., Wang, Y., Zhang, Y., Huang, Y., Li, Y., Wu, Y., Liu, Y., Pan, Y., Zheng, Y., Hong, Y., Shi, Y., Feng, Y., Jiang, Z., Han, Z., Wu, Z.F., Liu, Z.: Wan: Open and advanced large-scale video generative models (2025), https://arxiv.org/abs/2503.20314
33. Wang, X., Zhang, S., Zhang, H., Liu, Y., Zhang, Y., Gao, C., Sang, N.: Videolcm: Video latent consistency model (2023), https://arxiv.org/abs/2312.09109
34. Wang, X., Zhang, X., Luo, Z., Sun, Q., Cui, Y., Wang, J., Zhang, F., Wang, Y., Li, Z., Yu, Q., Zhao, Y., Ao, Y., Min, X., Li, T., Wu, B., Zhao, B., Zhang, B., Wang, L., Liu, G., He, Z., Yang, X., Liu, J., Lin, Y., Huang, T., Wang, Z.: Emu3: Next-token prediction is all you need (2024), https://arxiv.org/abs/2409.18869
35. Yan, W., Zhang, Y., Abbeel, P., Srinivas, A.: Videogpt: Video generation using vq-vae and transformers (2021), https://arxiv.org/abs/2104.10157
36. Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., Yin, D., Zhang, Y., Wang, W., Cheng, Y., Xu, B., Gu, X., Dong, Y., Tang, J.: Cogvideox: Text-to-video diffusion models with an expert transformer (2025), https://arxiv.org/abs/2408.06072
37. Ye, Y., Guo, J., Wu, H., He, T., Pearce, T., Rashid, T., Hofmann, K., Bian, J.: Fast autoregressive video generation with diagonal decoding (2025), https://arxiv.org/abs/2503.14070
38. Yu, L., Cheng, Y., Sohn, K., Lezama, J., Zhang, H., Chang, H., Hauptmann, A.G., Yang, M.H., Hao, Y., Essa, I., et al.: Magvit: Masked generative video transformer. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10459–10469. IEEE Computer Society (2023)
39. Yu, L., Lezama, J., Gundavarapu, N.B., Versari, L., Sohn, K., Minnen, D., Cheng, Y., Birodkar, V., Gupta, A., Gu, X., Hauptmann, A.G., Gong, B., Yang, M.H., Essa, I., Ross, D.A., Jiang, L.: Language model beats diffusion – tokenizer is key to visual generation (2024), https://arxiv.org/abs/2310.05737
40. Yuan, H., Chen, W., Cen, J., Yu, H., Liang, J., Chang, S., Lin, Z., Feng, T., Liu, P., Xing, J., Luo, H., Tang, J., Wang, F., Yang, Y.: Lumos-1: On autoregressive video generation from a unified model perspective (2025), https://arxiv.org/abs/2507.08801
41. Zhai, Y., Lin, K., Yang, Z., Li, L., Wang, J., Lin, C.C., Doermann, D., Yuan, J., Wang, L.: Motion consistency model: Accelerating video diffusion with disentangled motion-appearance distillation. In: Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., Zhang, C. (eds.) Advances in Neural Information Processing Systems. vol. 37, pp. 111000–111021. Curran Associates, Inc. (2024). https://doi.org/10.52202/079017-3524, https://proceedings.neurips.cc/

paper_files/paper/2024/file/c859b99b5d717c9035e79d43dfd69435-Paper-Conference.pdf
42. Zhang, E., Tang, J., Ning, X., Zhang, L.: Training-free and hardware-friendly acceleration for diffusion models via similarity-based token pruning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 9878–9886 (2025)
43. Zhang, E., Xiao, B., Tang, J., Ma, Q., Zou, C., Ning, X., Hu, X., Zhang, L.: Token pruning for caching better: 9 times acceleration on stable diffusion for free (2024), https://arxiv.org/abs/2501.00375
44. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric (2018), https://arxiv.org/abs/1801.03924
45. Zhao, Y., Xiong, Y., Krähenbühl, P.: Image and video tokenization with binary spherical quantization (2024), https://arxiv.org/abs/2406.07548
46. Zhou, X., Liang, D., Chen, K., Feng, T., Chen, X., Lin, H., Ding, Y., Tan, F., Zhao, H., Bai, X.: Less is enough: Training-free video diffusion acceleration via runtime-adaptive caching (2025), https://arxiv.org/abs/2507.02860
47. Zou, C., Liu, X., Liu, T., Huang, S., Zhang, L.: Accelerating diffusion transformers with token-wise feature caching (2025), https://arxiv.org/abs/2410.05317