VideoDiffCR: Content-Aware Adaptive Token Pruning for Autoregressive Video Diffusion Transformers

Ayush Shrivastava [1 2] **Connelly Barnes** [2] **Haoran You** [2] **Yifan Gong** [2] **Yan Kang** [2] **Andrew Owens** [3]
Eli Shechtman [2]

Abstract

Causal autoregressive video diffusion models enable streaming generation by producing video chunks sequentially and reusing a rolling KV cache. However, even in this efficient setting, every denoising step still processes all space–time tokens in the current chunk through every transformer block, resulting in substantial redundant computation. We present VideoDiffCR , a differentiable token-pruning framework for causal video diffusion transformers. Building on DiffCR, our method learns layer- and timestep-adaptive token retention ratios, but introduces video-specific changes required for autoregressive generation: a temporal mixing router that scores tokens using cross-frame context, an alternating initialization of layerwise retention ratios for stable training, and a denoising-step schedule that prunes more aggressively at noisier steps while preserving tokens at later refinement steps. We apply VideoDiffCR to a Self Forcing-distilled Wan2.1 1.3B video diffusion model and train it using only the DMD distillation signal on VidProM text prompts, without video supervision. On VBench, VideoDiffCR maintains performance close to the dense Self Forcing baseline while achieving up to $1.91\times$ end-to-end speedup, showing that learned token pruning can substantially improve the efficiency of causal video diffusion models.

1. Introduction

Diffusion models have become a dominant paradigm for visual generation, progressing from pixel-space denoising models to latent diffusion systems and transformer-based diffusion architectures (Ho et al., 2020; Rombach et al., 2022; Peebles & Xie, 2023). For video generation, diffusion transformers are especially attractive because they can model long-range interactions across space and time. This expressivity, however, comes at a high computational cost: the number of tokens grows with spatial resolution, temporal length, and latent-frame chunk size, and each transformer block applies attention and feed-forward computation to every token.

Recent causal video diffusion models reduce the latency of video generation by producing videos autoregressively. In particular, Self Forcing trains a causal video diffusion model with autoregressive self-rollout and uses a rolling KV cache to support efficient streaming generation (Huang et al., 2025). This changes the generation regime from full-clip denoising to chunk-by-chunk synthesis. Nevertheless, the computation within each newly generated chunk remains dense: at every denoising step, all space–time tokens in the current chunk are processed by all DiT layers. Thus, even though causal generation improves streaming latency, it does not remove redundancy within the current chunk.

In image diffusion transformers, token pruning and dynamic inference methods have shown that not all tokens, layers, or denoising timesteps require equal computation (Raposo et al., 2024; You et al., 2025). DiffCR learns token importance together with differentiable layer- and timestep-dependent compression ratios, allowing image DiTs to allocate computation adaptively across the denoising process (You et al., 2025). Directly transferring this idea to video, however, is non-trivial. Video tokens are spatiotemporal rather than purely spatial, pruning decisions affect the KV cache used by future chunks, and later denoising steps are particularly sensitive because they refine visual details and temporal consistency.

We propose ***VideoDiffCR***, a differentiable token-pruning framework for causal video diffusion transformers. Our method extends DiffCR to the autoregressive video setting with three key modifications. First, we replace the per-token linear router with a temporal mixing router that incorporates cross-frame context before predicting token importance. Second, we initialize layerwise retention ratios with an alternating $1.0, 0.8, 1.0, 0.8, \ldots$ pattern, which provides

[1]University of Michigan [2]Adobe [3]Cornell University. Correspondence to: Ayush Shrivastava <ayshrv@umich.edu>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

a stable prior for the 30-layer Wan2.1 backbone. Third, we use a timestep-dependent retention schedule around each learned layer mean, allowing earlier noisier denoising steps to prune more aggressively while later steps retain more tokens to preserve video quality.

We evaluate VideoDiffCR on a Self Forcing-distilled Wan2.1 1.3B model trained with the DMD objective from a Wan2.1 14B teacher. The pruned method is trained using VidProM text prompts only, without explicit video supervision. On VBench (Huang et al., 2024), VideoDiffCR maintains Total, Quality, and Semantic scores close to the dense Self Forcing baseline while improving throughput substantially. At 30% token reduction, it reduces per-chunk latency from 1.29s to 0.67s and achieves a $1.91\times$ end-to-end speedup.

Our contributions are:

- We introduce VideoDiffCR , a differentiable token-pruning framework for causal, KV-cached video diffusion transformers.
- We design a temporal mixing router that predicts token importance using cross-frame context, making routing decisions better suited to spatiotemporal tokens.
- We adapt layer- and timestep-wise differentiable retention ratios to few-step autoregressive video diffusion using a stable alternating initialization and a denoising-step-aware pruning schedule.
- We show that VideoDiffCR preserves VBench performance close to a dense Self Forcing baseline while achieving up to $1.91\times$ end-to-end speedup.

2. Related Work

Video diffusion and causal generation. Diffusion models have achieved strong results in image and video synthesis by learning iterative denoising processes (Ho et al., 2020; Rombach et al., 2022). Diffusion transformers further replace convolutional U-Net backbones with transformer blocks over latent patches, improving scalability but increasing token-dependent computation (Peebles & Xie, 2023). Most video diffusion systems denoise an entire clip jointly, which enables bidirectional temporal interactions but prevents true streaming generation. Causal and autoregressive video diffusion models address this limitation by generating videos sequentially. Self Forcing bridges the train-test gap in autoregressive video diffusion by training with self-generated context and a holistic distribution-matching objective, while using a rolling KV cache for efficient long video generation (Huang et al., 2025). Our work builds on this causal generation setting and targets the remaining dense computation inside each generated chunk.

Token pruning and dynamic computation. Dynamic inference methods reduce computation by allocating model capacity adaptively across inputs, layers, or tokens. Mixture-of-Depths routes only a subset of tokens through transformer layers, allowing token-level conditional computation (Raposo et al., 2024). In vision transformers, soft token pruning methods such as SPViT reduce token counts to improve inference efficiency (Kong et al., 2022). For diffusion transformers, recent methods such as FlexiDiT and D^{2}iT explore adaptive computation and patch/token reduction during generation (Anagnostidis et al., 2025; Jia et al., 2025). These methods show that diffusion models contain substantial redundancy, but many are designed for image generation or rely on schedules that do not directly account for causal video generation.

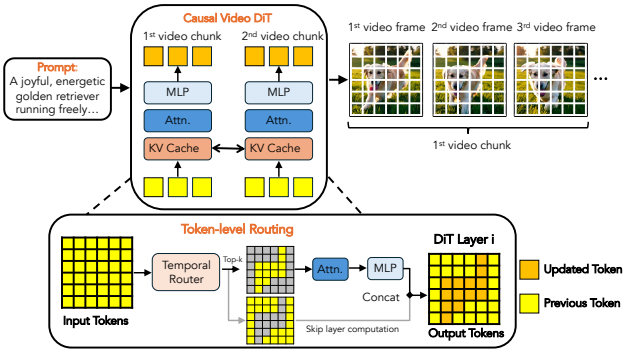

Figure 1. **Overview of VideoDiffCR.** A temporal router predicts token importance within each latent video chunk, after which only the top-ranked tokens are processed by the DiT block while skipped tokens bypass computation. Full token features are reused before updating the causal KV cache for subsequent chunks.

Differentiable compression ratios in diffusion transformers. DiffCR introduces learnable token pruning for image DiTs by attaching routers to transformer layers and learning differentiable compression ratios across layers and denoising timesteps (You et al., 2025). This is closely related to our work, but image-DiT pruning does not directly handle the temporal structure and KV-cache dependencies of autoregressive video models. VideoDiffCR extends DiffCR to causal video diffusion by making the router temporally aware, stabilizing ratio learning with an alternating initialization, and adapting retention ratios to the four-step denoising schedule of the Self Forcing student. These changes allow token pruning to be applied to spatiotemporal video tokens while preserving quality across autoregressive chunks.

3. Approach

Background: DiffCR. DiffCR (You et al., 2025) accelerates image DiTs by attaching a token router to each transformer block and learning differentiable token retention ratios across layers and denoising timesteps. Given a retention ratio ρ, only the top-k tokens are processed by the attention and MLP layers, while the remaining tokens bypass the block. VideoDiffCR adopts this differentiable routing for-

mulation, but modifies it for causal video generation where tokens carry temporal structure and pruning decisions affect the KV cache used by future chunks.

Causal video backbone. We build on the Self Forcing Wan2.1-T2V-1.3B student, which generates $C = 3$ latent frames per chunk with $T = 4$ denoising steps and reuses a causal KV cache over previous chunks. Unlike dense Self Forcing, VideoDiffCR prunes tokens inside each DiT block before attention and MLP computation.

Temporal mixing router. A per-token router is insufficient for video because token importance depends on motion and cross-frame context. We therefore introduce a temporal mixing router that first aggregates information across the latent frames in the current chunk and then predicts a scalar importance score for each token. The top-k tokens are processed by the block, while the remaining tokens bypass computation following the DiffCR/MoD routing rule.

Layer- and timestep-adaptive ratios. For each layer ℓ, we learn a retention ratio $\rho^{(\ell)}$ and regularize the mean ratio toward a target budget:

$$\mathcal{L}_{\text{ratio}} = \left(\frac{1}{L} \sum_{\ell=1}^{L} \rho^{(\ell)} - \rho_{\text{target}} \right)^2 .$$

Instead of zero initialization, which is unstable for causal video generation, we initialize layer ratios as $1.0, 0.8, 1.0, 0.8, \ldots$ across the 30 DiT blocks. To account for the four-step denoising schedule, we use timestep-specific ratios $\rho_t^{(\ell)} = \rho^{(\ell)} + \delta_t$ with $\delta_{1:4} = (-0.3, -0.1, +0.1, +0.3)$, allowing stronger pruning at noisier steps and higher token retention at later refinement steps. After each block, full token features are restored before updating the KV cache, preserving context for future chunks.

4. Experiments

We evaluate VideoDiffCR on the Self Forcing Wan2.1-T2V-1.3B student distilled from a Wan2.1 14B teacher. The model generates 5-second videos at 832×480 resolution and 16 FPS using $C = 3$ latent frames per chunk and $T = 4$ denoising steps. We train with the DMD objective on VidProM prompts only, without video supervision, using AdamW with learning rate 10^{-5} for 20K iterations. Ratio bins are placed at 10% intervals, with weighted interpolation during training and nearest-bin rounding at inference. We report VBench Total, Quality, and Semantic scores, as well as end-to-end latency, throughput, and speedup on a single H100 GPU. We compare against dense Self Forcing and token-reduction baselines adapted to the same backbone: FlexiDiT, D^{2}iT, and SPViT.

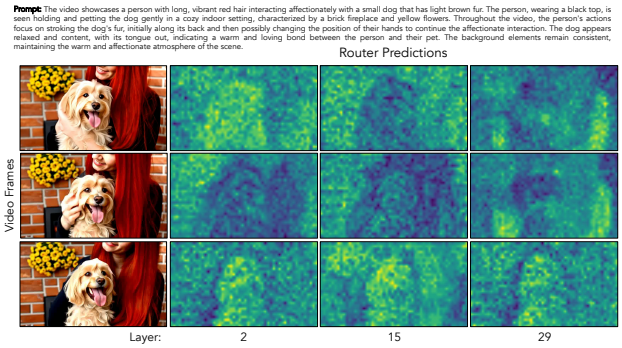

Figure 2. **Router predictions for different generated frames and different layers.**

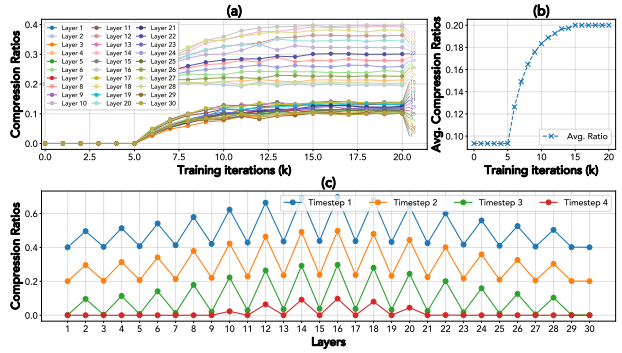

Figure 3. **Layerwise compression ratios in VideoDiffCR.**

5. Results

Main results. Table 1 compares VideoDiffCR with dense Self Forcing and token-reduction baselines. VideoDiffCR preserves VBench performance close to the dense model across all compression levels while improving throughput. At 10% and 20% token reduction, our method remains within 0.10 and 0.30 Total Score of the dense baseline, respectively. At 30% reduction, VideoDiffCR achieves a $1.91\times$ end-to-end speedup, reducing latency from 1.29s to 0.67s while maintaining a Total Score of 81.32. Compared with FlexiDiT, D^{2}iT, and SPViT, VideoDiffCR consistently preserves stronger semantic alignment at comparable speedups.

Ablations. Table 2 validates the main design choices. Learning ratios from scratch performs poorly, showing that direct transfer of DiffCR initialization is unstable for causal video generation. The alternating initialization improves stability, while adding timestep-wise ratios gives the largest gain, increasing Total Score from 63.56 to 78.94. Finally, the temporal router improves the full model to 82.21 Total Score, confirming that token importance in video benefits from cross-frame context.

Qualitative analysis. Figure 2 shows that different layers learn different pruning patterns based on the semantics of the generated content, suggesting that the router learns

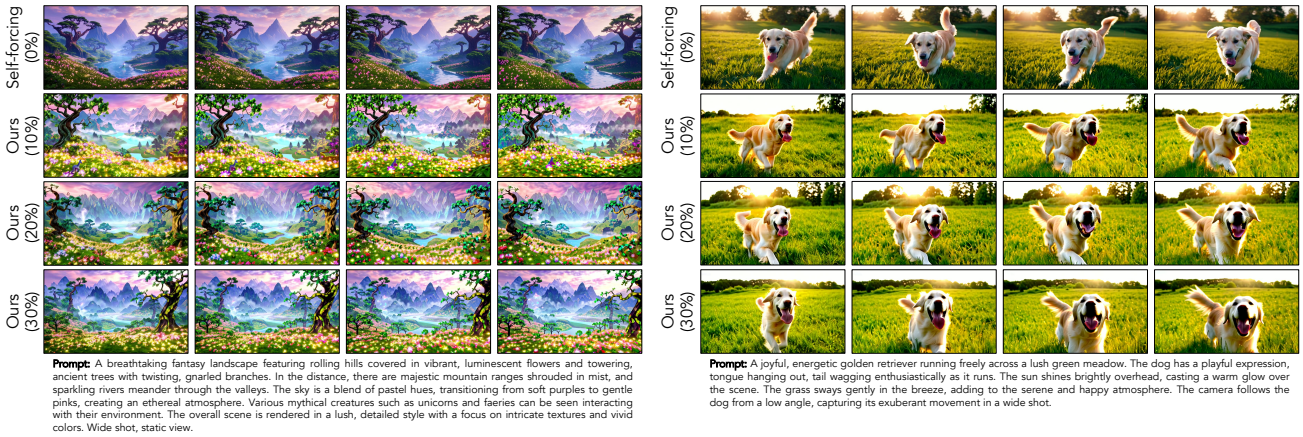

Figure 4. **Qualitative results of VideoDiffCR for video generation with different compression ratios (10%, 20%, 30%).**

Table 1. Comparison with other methods on VBench.

Token Reduction Rate	Method	Total Score ↑	Quality Score ↑	Semantic Score ↑	Latency (s) ↓	Throughput (fps) ↑	Speedup ↑
0%	Self-Forcing	82.31	83.07	79.28	1.29	9.23	1.0×
10%	VideoDiffCR (Ours)	82.21	82.64	79.11	1.17	10.15	1.1×
20%	FlexiDiT (Anagnostidis et al., 2025)	81.80	83.22	76.10	1.17	10.15	1.1×
	D^2iT (Jia et al., 2025)	81.84	83.42	75.51	1.08	11.08	1.2×
	SPViT (Kong et al., 2022)	81.23	82.95	74.36	1.08	11.08	1.2×
	VideoDiffCR (Ours)	**82.01**	**82.21**	**79.01**	**0.92**	**12.92**	1.4×
30%	FlexiDiT (Anagnostidis et al., 2025)	80.25	82.02	73.19	0.92	12.92	1.4×
	D^2iT (Jia et al., 2025)	81.08	82.85	74.02	0.99	12.00	1.3×
	SPViT (Kong et al., 2022)	79.20	81.21	71.18	0.86	13.85	1.5×
	VideoDiffCR (Ours)	**81.32**	**82.64**	**78.12**	**0.67**	**17.69**	1.91×

Table 2. Ablation of VideoDiffCR components on VBench. We progressively add the proposed ratio schedule and temporal router, showing that each design choice improves generation quality.

Variant	Total Score ↑	Quality Score ↑	Semantic Score ↑
Learning ratios from scratch	56.31	67.98	66.64
Constant alternating schedule	61.04	70.32	67.94
+ DiffCR schedule	63.56	73.81	69.42
+ Timewise ratios	78.94	80.27	77.18
+ Temporal Router	**82.21**	**82.64**	**79.11**

meaningful token selection rather than pruning uniformly. Figure 3 further shows that the learned retention ratios vary across both layers and denoising steps, with more pruning concentrated in less sensitive stages and higher retention in earlier and later layer and denoising step. Finally, Figure 4 compares generated videos under different pruning levels. Moderate pruning largely preserves scene layout, object appearance, and motion relative to dense Self Forcing, while more aggressive pruning can occasionally introduce small visual changes or cross-chunk identity drift.

6. Conclusion and Future Work

We presented VideoDiffCR, a differentiable token-pruning framework for causal video diffusion transformers. By combining temporal-aware routing with layer- and timestep-adaptive retention ratios, VideoDiffCR extends DiffCR to autoregressive, KV-cached video generation. On a Self Forcing-distilled Wan2.1 1.3B model, our method maintains VBench performance close to the dense baseline while achieving up to a 1.91× end-to-end speedup. Under aggressive pruning, we observe occasional identity drift across generated chunks. Since autoregressive video generation relies on previously generated chunks through the KV cache, pruning can remove or weaken token information that is useful for preserving object appearance over time. This can lead to small changes in object identity, texture, or structure when generation moves from one chunk to the next. Future work could address this by adding cross-chunk consistency objectives or cache-aware routing constraints that encourage identity and appearance preservation across video rollouts.

References

Anagnostidis, S., Bachmann, G., Kim, Y., Kohler, J., Georgopoulos, M., Sanakoyeu, A., Du, Y., Pumarola, A., Thabet, A., and Schönfeld, E. Flexidit: Your diffusion transformer can easily generate high-quality samples with less compute. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 28316–28326, 2025.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

Huang, X., Li, Z., He, G., Zhou, M., and Shechtman, E. Self forcing: Bridging the train–test gap in autoregressive video diffusion. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

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., and Liu, Z. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Jia, W., Huang, M., Chen, N., Zhang, L., and Mao, Z. Dˆ 2it: Dynamic diffusion transformer for accurate image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12860–12870, 2025.

Kong, Z., Dong, P., Ma, X., Meng, X., Niu, W., Sun, M., Shen, X., Yuan, G., Ren, B., Tang, H., et al. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In *European conference on computer vision*, pp. 620–640. Springer, 2022.

Peebles, W. and Xie, S. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Raposo, D., Ritter, S., Richards, B., Lillicrap, T., Humphreys, P. C., and Santoro, A. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*, 2024.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

You, H., Barnes, C., Zhou, Y., Kang, Y., Du, Z., Zhou, W., Zhang, L., Nitzan, Y., Liu, X., Lin, Z., Shechtman, E., Amirghodsi, S., and Lin, Y. Layer- and timestep-adaptive differentiable token compression ratios for efficient diffusion transformers. *arXiv preprint arXiv:2412.16822*, 2025.