Dynamic Video Generation: Shaping Video Generation Across Time and Space

Shikang Zheng[1,2*], Jingkai Huang[2*], Jiacheng Liu[1], Guantao Chen[1], Lixuan He[3], Yuqi Lin[1], Peiliang Cai[1], Linfeng Zhang[1†]

[1]Shanghai Jiao Tong University, [2]South China University of Technology, [3]Tsinghua University

Abstract

Diffusion models have achieved impressive performance in video generation, but their iterative denoising process remains computationally expensive due to the large number of tokens processed at each timestep. Recently, progressive resolution sampling has emerged as a promising acceleration approach by reducing latent resolution in early stages. However, scaling this idea to video generation remains challenging, as the additional temporal dimension introduces diverse spatio-temporal demands across different videos, and compressing only a single dimension often leads to limited acceleration or degraded quality. Therefore, we propose **DVG**, a **D**ynamic **V**ideo **G**eneration framework that jointly allocates computation across time and space, automatically selecting content-aware acceleration strategies without manual tuning or retraining. **DVG** achieves near-lossless acceleration across models and tasks, reaching up to **7**× speedup on HunyuanVideo and HunyuanVideo-1.5, and **18**× when combined with distillation, demonstrating its potential as a key component in today's large-scale efficient video generation systems. Our code is available at: *https://github.com/DarrenZheng303/DVG*.



Figure 1: Videos generated on **HunyuanVideo-1.5** using **DVG** with distillation at **18**× speedup.

1 Introduction

Diffusion models have recently achieved remarkable performance in video generation, delivering state-of-the-art fidelity and diversity. However, their strong generation capability comes with substantial inference cost. Due to the iterative nature of diffusion sampling, each sample typically requires

*Equal contribution. †Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

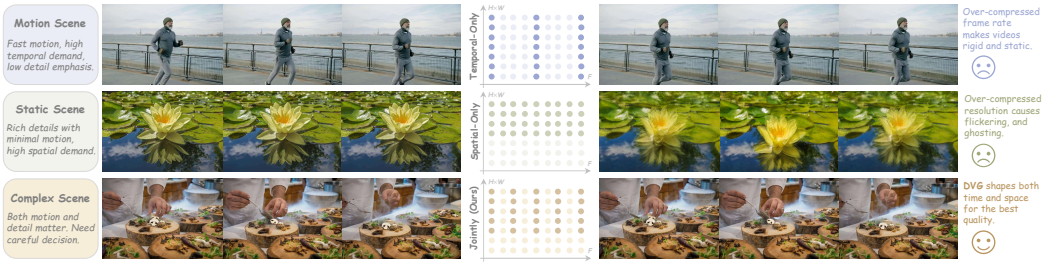


Figure 2: **Different videos require different spatio-temporal compression strategies.** Compressing only a single dimension often causes motion rigidity or visual artifacts.

dozens of forward passes through a large transformer backbone, making real-time generation and deployment on resource-constrained devices highly challenging. This computational bottleneck has motivated extensive research on efficient diffusion inference.

Existing acceleration methods can be broadly divided into two directions. One line of work reduces the number of sampling steps using advanced numerical solvers [22, 46], distillation [30], or consistency training [34]. Another line of work reduces the computation of each denoising step by optimizing model execution, such as sparse attention [8, 41], token merging [2, 4, 24]. For efficient large-scale video generation models, recent progress has been largely driven by step distillation [47, 53], which reduces sampling to only a few denoising steps. As the step count decreases, the per-step cost of processing massive spatio-temporal tokens becomes the dominant bottleneck. Recently, progressive resolution sampling [50, 36, 45] has emerged as a promising technique to accelerate image generation: it performs early denoising at lower latent resolutions to reduce the computational cost of spatial tokens [13, 36]. However, scaling this idea to video generation introduces new challenges.

Unlike images, video generation introduces an additional temporal dimension. Video prompts naturally exhibit diverse spatio-temporal demands. Some prompts describe static scenes with rich textures, where preserving spatial details is more important than using a dense frame rate. Others describe fast actions or complex camera motions, where temporal continuity is more critical than fine details. Moreover, achieving a substantial acceleration ratio by compressing only a single dimension often comes at a high cost: reducing spatial resolution causes flickering and unstable details, while reducing frame rate weakens motion dynamics, as shown in Fig. 3. Therefore, an efficient video generation system should not only use a fixed acceleration schedule on one dimension, but adapt its compression strategy according to the video content on both spatial and temporal dimensions.

This nature further motivates a different view of video acceleration: instead of manually choosing a fixed compression ratio as an isolated heuristic, we formulate it as a budgeted spatio-temporal resource allocation problem. A video model spends computation over both space and time: spatial tokens determine how much visual structure and detail can be refined, while temporal tokens determine how densely motion is represented. Under a fixed compute budget, allocating more spatial tokens to one dimension necessarily leaves fewer affordable tokens for the other. Therefore, effective video acceleration requires an optimization mechanism that automatically balances spatial and temporal computation according to the video content.

Motivated by these observations, we propose **DVG**, a **D**ynamic **V**ideo **G**eneration framework that automatically selects the acceleration strategy based on the video content and jointly compress the latent resolution and frame during denoising. It first performs a few low-cost denoising steps to obtain a coarse latent sketch, from which it estimates spatial detail and motion demands directly in latent space within **0.01s**, avoiding expensive VAE decoding. Given a target compute budget, DVG automatically selects the best spatio-temporal compression ratio without manual tuning, and finally upsamples the latent back to the original settings. Extensive experiments across multiple models and tasks show that DVG achieves **4**× speedup on Wan2.2, **7**× speedup on HunyuanVideo and HunyuanVideo-1.5 while preserving generation quality. When combined with distillation, DVG further reaches **18**× speedup, as illustrated in Fig. 1. In summary, our main contributions are:

- **Spatio-temporal View of Video Acceleration.** We revisit video diffusion acceleration from a joint space-time perspective. Since compressing only a single dimension leads to limited speedup or noticeable quality degradation, we propose DVG, a dynamic video generation framework that jointly optimizes spatial and temporal compression.

- **Content-Aware Allocation.** We formulate spatio-temporal allocation as a content-aware optimization problem. DVG automatically estimates spatial and temporal demands from latent sketches, and selects the best compression strategy under a target compute budget without manual tuning.
- **Outstanding Performance.** We evaluate DVG on various models and tasks, including the dense model HunyuanVideo, the MoE-based model Wan2.2, and the efficient video generation system HunyuanVideo-1.5. DVG consistently achieves strong acceleration while preserving quality.

2 Related Work

Diffusion models [32, 10] have demonstrated remarkable capabilities in various generation tasks. While early approaches predominantly relied on U-Net architectures [28], their scalability limitations were later addressed by Diffusion Transformers [27], which have emerged as the dominant backbone for large-scale generative models. In particular, many recent state-of-the-art video generation models adopt DiT-based architectures, enabling high-resolution and long-duration video synthesis [6, 5, 51, 39]. Despite these advances, the iterative sampling process remains a fundamental limitation of diffusion models, introducing substantial computational overhead during inference. This has driven two primary research directions: reducing the number of sampling steps and lowering the per-step computational cost. Beyond efficiency, a key challenge is to preserve generative fidelity and stability, especially under aggressive acceleration.

2.1 Sampling Timestep Reduction

DDIM [33] introduced deterministic few-step sampling that preserves perceptual quality. Higher-order ODE solvers (DPM-Solver and variants) [22, 23, 46] further improve the accuracy–efficiency trade-off via multi-step and multi-stage discretizations with controlled local truncation error. Rectified Flow [21] shortens transport paths, while distillation methods [30] compress long sampling trajectories into compact generators. Consistency models [34] also enable few-step generation by learning a direct noise-to-clean mapping. Feature caching [20, 49, 48] methods reuse intermediate features across timesteps to reduce redundant computation, effectively lowering the number of actual computed timesteps. Recently, step-distillation [47, 53] has emerged as the dominant acceleration paradigm in practice, directly reducing the number of inference steps while maintaining strong generation quality. Its simplicity and effectiveness have led to widespread adoption in large-scale video generation.

2.2 Per-Step Computation Reduction

Pruning [9, 52], quantization [18, 31, 16], and token reduction [3, 15, 42, 43, 7] reduce computational cost with relatively low runtime overhead. In parallel, sparse attention methods, including fixed-window, strided, and adaptive patterns [1, 41, 8], aim to alleviate the quadratic complexity of self-attention by restricting token interactions, though their practical speedups are often limited. To achieve more substantial acceleration, recent efforts have focused on reducing spatial resolution during generation. Cascade diffusion frameworks [25, 11, 17, 29] generate low-resolution outputs and progressively refining them via learned upsampling, but often require retraining and auxiliary modules. Progressive resolution sampling methods [36, 50, 13], adapt resolution across sampling stages, offering improved flexibility but relying on carefully designed schedules and hyperparameters, which makes it hard to scale them to large video models with complex spatiotemporal dependencies.

3 Method

3.1 Preliminary

Diffusion Models. Diffusion models generate samples by simulating a forward–reverse stochastic process. In the forward process, a clean sample x_0 is progressively corrupted by Gaussian noise:

$$x_t = \sqrt{\alpha_t} x_0 + \sqrt{1-\alpha_t}\,\epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \mathbf{I}), \tag{1}$$

where α_t defines a predefined noise schedule. The reverse process learns to recover x_0 by predicting the noise component $\epsilon_\theta(x_t, t)$, leading to the update:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\, \epsilon_\theta(x_t, t) \right) + \sigma_t\, \epsilon, \tag{2}$$

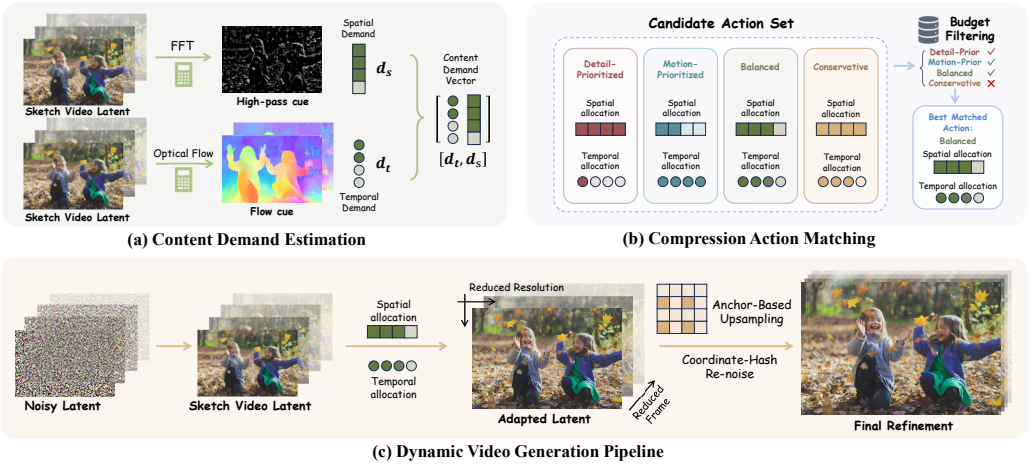


Figure 3: **DVG Framework.** DVG first predicts a coarse latent video sketch, then analyzes its spatial and temporal demands directly in latent space. Under a target compute budget, DVG selects the best compression action for denoising, then restores the latent to the original setting for refinement.

where $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, and σ_t controls the stochasticity. In video diffusion models, the latent variable x_t represents a spatiotemporal feature tensor, and each denoising step involves high-dimensional computation across both spatial and temporal dimensions, leading to substantial inference cost.

Progressive Resolution Sampling. Progressive resolution methods accelerate diffusion by performing early denoising on a lower-resolution latent grid, where fine details are less important. Given a latent feature map $\mathbf{x}_t \in \mathbb{R}^{H\times W\times C}$, the process is:

$$\hat{\mathbf{x}}_t = \mathcal{U}_s\big(f(\mathcal{D}_s(\mathbf{x}_t))\big),$$

where $\mathcal{D}_s$ and $\mathcal{U}_s$ are downsampling and upsampling operators, and $f(\cdot)$ denotes denoising on the coarse grid. By reducing the spatial tokens, these methods reduced intermediate computation.

3.2 Overview of DVG

DVG is a content-adaptive acceleration framework for video diffusion transformers that jointly adjusts spatial resolution and temporal frame-rate during denoising. The key observation is that different videos require different computational allocation: videos with rich textures benefit more from high spatial resolution, while videos with large motions require denser temporal sampling. Instead of using a fixed resolution-frame schedule for all prompts, DVG dynamically selects an efficient spatio-temporal denoising action according to an early preview of the generated video.

DVG first performs a few denoising steps at low spatial resolution and low frame-rate to obtain a noisy but informative intermediate latent. It then predicts a coarse clean latent in one step as an early video sketch. Based on this sketch, DVG estimates the spatial detail demand and temporal motion demand in latent space. Under a user-specified computational budget, DVG searches a pre-defined action list and selects the action whose spatial and temporal allocation best matches the estimated content demand. After denoising the intermediate stage with the selected action, the latent is upsampled to full spatial resolution and full frame-rate for final refinement.

3.3 Sketching Early Perception of Video Content

At the beginning of denoising, the latent X_0 is dominated by Gaussian noise and contains little semantic information, making early compression decisions unreliable. To obtain a lightweight preview of the video content, DVG first performs a few denoising steps under a low resolution and low frame-rate setting, producing a sketch latent X_k that already captures coarse motion and structure. Instead of running the remaining denoising process for schedule estimation, we directly extrapolate a rough clean latent from the current flow prediction. Specifically, given the model prediction $v_\theta(X_k, k)$

at step k, we estimate:

$$\tilde{X} = X_k + \alpha_k \, v_\theta(X_k, k),$$

where α_k is a timestep-dependent scaling factor controlling the extrapolation strength. This produces a coarse video latent that is sufficient for estimating motion intensity and spatial detail complexity, while avoiding expensive full-resolution denoising.

3.4 Inferring Content Demands in Latent Space

Given the predicted latent sketch $\tilde{X} \in \mathbb{R}^{C\times F\times H\times W}$, DVG estimates two complementary content demands: spatial demand and temporal demand. The spatial demand reflects the amount of fine-grained visual detail, while the temporal demand measures motion intensity and temporal variation.

Spatial demand. DVG estimates spatial complexity with Fast Fourier Transform. It applies a high-pass operator $\mathcal{H}(\cdot)$ to each latent frame and compute the normalized high-frequency energy:

$$d_s = \mathrm{Norm}\left(\frac{1}{F}\sum_{t=1}^{F}\left\|\mathcal{H}(\tilde{X}_t)\right\|_2\right), \tag{3}$$

where $\tilde{X}_t$ denotes the t-th latent frame. Larger d_s values indicate richer textures and more detailed scene structures, suggesting that more computation should be allocated to spatial resolution.

Temporal demand. To estimate motion complexity, DVG computes optical flow on the latent sketch. Let $\mathcal{F}_{t\to t+1}$ denote the optical flow between adjacent frames. The temporal demand is defined as:

$$d_t = \mathrm{Norm}\left(\frac{1}{F-1}\sum_{t=1}^{F-1}\left\|\mathcal{F}_{t\to t+1}\right\|_2\right). \tag{4}$$

Larger d_t values indicate stronger motion and more frequent temporal changes, implying that higher temporal resolution should be preserved. Then, DVG normalized the two demand into weights:

$$m_s = \frac{\exp(\alpha d_s)}{\exp(\alpha d_s)+\exp(\alpha d_t)}, \qquad m_t = 1 - m_s, \tag{5}$$

where α controls the allocation sharpness. Larger m_s favors richer spatial tokens, while larger m_t favors more temporal tokens.

3.5 Constructing the Space of Spatio-Temporal Actions

To make the acceleration ratio controllable, DVG pre-defines an action list $\mathcal{A}$. Each action $a \in \mathcal{A}$ specifies the spatial and temporal scales used across denoising stages:

$$a = \left\{(r_s^1, r_t^1), (r_s^2, r_t^2), \cdots, (r_s^L, r_t^L)\right\}, \tag{6}$$

where r_s^ℓ and r_t^ℓ denote the spatial compression ratio and temporal compression ratio at stage ℓ, at a grid of 0.05. For example, an action may use $(0.5, 0.7)$ in the intermediate stage, meaning that denoising is performed with 50% original latent resolution and 70% frame-rate.

For an action a, its computational density is estimated as:

$$\rho(a) = \frac{\sum_{\ell=1}^{L} H_\ell W_\ell F_\ell \cdot N_\ell}{HWF \cdot N}, \tag{7}$$

where $H_\ell = r_s^\ell H$, $W_\ell = r_s^\ell W$, $F_\ell = r_t^\ell F$, and N_ℓ is the number of denoising steps assigned to stage ℓ. Here, H, W, F, and N are the full-resolution height, width, frame count, and total denoising steps. The density $\rho(a)$ approximates the relative computation cost compared with full-resolution full-frame-rate denoising. Given a target budget density D and tolerance ϵ, DVG retains only actions whose computational density satisfies:

$$\mathcal{A}_B = \{a \in \mathcal{A} \mid |\rho(a) - D| \le \epsilon\}. \tag{8}$$

This budget filtering step ensures that the selected action satisfies the desired acceleration level before content-aware action matching is applied.

3.6 Matching Actions to Content Demands

After obtaining the budget-feasible action set $\mathcal{A}_B$, DVG selects the candidate action according to how well it matches the spatial and temporal demands. For each action a, we define its effective spatial and temporal allocation as:

$$g_s(a) = \frac{1}{L}\sum_{\ell=1}^{L} r_s^{\ell}, \qquad g_t(a) = \frac{1}{L}\sum_{\ell=1}^{L} r_t^{\ell}, \tag{9}$$

where L denotes the number of denoising stages. Intuitively, $g_s(a)$ measures how much spatial resolution the action preserves, while $g_t(a)$ measures how much temporal frame-rate it preserves.

The final action is then selected by solving the following demand matching objective:

$$a^{\star} = \arg\max_{a\in\mathcal{A}_B} \left[-\lambda\left(g_s(a) - m_s\right)^2 - (1-\lambda)\left(g_t(a) - m_t\right)^2\right], \tag{10}$$

where λ balances spatial and temporal matching. This formulation encourages DVG to allocate more spatial tokens to texture-rich videos and more frame to motion-intensive videos, while still satisfying the overall target computational budget.

3.7 Flexible Spatio-Temporal Latent Reshaping

When transitioning to a different resolution and frame-rate, DVG first resizes the current latent to the target spatio-temporal grid and then re-noises it at the next timestep. For each axis with source length K and target length N, we assign source position i to an anchor index:

$$a_i = \mathrm{round}\left(\frac{i(N-1)}{K-1}\right). \tag{11}$$

Each target position p is then interpolated from neighboring anchors:

$$y(p) = (1-\beta)x(i) + \beta x(i+1), \qquad \beta = \frac{p - a_i}{a_{i+1} - a_i}. \tag{12}$$

Applying this operation along temporal, height, and width axes produces the resized latent $\bar{X}$. Next, DVG generates coordinate-aligned Gaussian noise directly on the target grid. For each coordinate (t, h, w, c) and random seed s, a deterministic hash generates uniform variables, which are converted into Gaussian noise using the Box–Muller transform:

$$\epsilon_{\mathrm{coord}}(t,h,w,c) = \sqrt{-2\log u_1}\cos(2\pi u_2), \qquad u_1, u_2 = \mathrm{Hash}(t,h,w,c,s). \tag{13}$$

The latent at the next timestep is then obtained through scheduler re-noising:

$$X_{\tau}^{\mathrm{next}} = \bar{X} + \sigma(\tau)\epsilon_{\mathrm{coord}}. \tag{14}$$

Unlike prior progressive sampling methods that are usually restricted to fixed $2\times$ spatial or temporal scaling, our anchor-based resizing and coordinate-hash re-noising support arbitrary target resolutions and frame-rates. This enables DVG to flexibly reshape the latent space and construct a much richer compression action space for content-aware allocation. Finally, the refine denoising stage is always performed at full resolution and frame-rate to further capture visual details and temporal consistency.

4 Experiments

Model Configurations. We evaluate DVG on both text-to-video (T2V) and image-to-video (I2V) tasks across models with different architectures. For T2V, we conduct experiments on dense video model HunyuanVideo [35] at 720p with 125 frames. To further validate generality, we also evaluated the MoE-based model Wan2.2-14B [37] at 480p with 81 frames, as well as the recent efficient video generation system HunyuanVideo-1.5 [38] in the recommended setting 480p, 121-frames. For I2V generation, we evaluate DVG on HunyuanVideo-1.5 using the same 121-frame configuration. These experiments allow us to evaluate DVG across different model architectures and examine its compatibility with modern -oriented designs.

Evaluation and Metrics. T2V results are measured using VBench [12], which contains 946 prompts and provides a comprehensive multi-dimensional evaluation of video quality. We report the key metrics: Quality, Semantic, and Total Scores. I2V evaluation follows VBench-I2V [12] with 1118 prompts, where I2V Scores and Total scores are reported. In addition, we include PSNR, SSIM, and LPIPS to assess its reconstruction fidelity. *Please refer to the Appendix for detailed settings.*

Table 1: **Quantitative comparison of text-to-video generation** on HunyuanVideo.

Method	NFE	Acceleration		VBench Score (%)		
		Latency(s) ↓	Speed ↑	Quality ↑	Semantic ↑	Total ↑
HunyuanVideo [35]	50	1947.38	1.00×	82.97 (+0.0%)	71.41 (+0.0%)	80.66 (+0.0%)
SpargeAttn [44]	50	1094.01	1.78×	83.26 (+0.3%)	71.24 (-0.2%)	80.35 (-0.4%)
LIPAR [26]	50	1421.44	1.37×	81.72 (-1.5%)	69.14 (-3.2%)	79.63 (-1.3%)
ToMe [2]	50	1343.02	1.45×	80.65 (-2.8%)	67.87 (-4.9%)	78.39 (-2.8%)
VGDFR [40]	50	1209.55	1.61×	80.45 (-3.0%)	69.45 (-2.7%)	79.03 (-2.0%)
Jenga-ProRes [45]	50	1361.38	1.43×	**83.53** (+0.7%)	70.11 (-1.8%)	80.07 (-0.7%)
Bottleneck [36]	30	818.23	2.38×	81.97 (-1.2%)	69.55 (-2.6%)	79.42 (-1.5%)
TeaCache [19] ($l = 0.15$)	50	861.23	2.26×	82.09 (-1.1%)	69.17 (-3.1%)	79.51 (-1.4%)
AdaCache [14]	50	743.27	2.62×	81.78 (-1.4%)	70.12 (-1.8%)	79.58 (-1.3%)
DVG ($D = 50\%$)	50	**608.56**	**3.20**×	82.10 **(-1.0%)**	**73.83 (+3.4%)**	**80.45 (-0.3%)**
22% steps	11	427.99	4.55×	80.52 (-3.0%)	69.01 (-3.4%)	77.74 (-3.6%)
TeaCache [19] ($l = 0.3$)	50	522.09	3.73×	81.55 (-1.7%)	68.73 (-3.8%)	78.99 (-2.1%)
Jenga-ProRes [45]	10	417.18	4.66×	79.24 (-4.5%)	65.57 (-8.2%)	76.50 (-5.2%)
Fresco [50]	20	491.76	3.96×	81.58 (-1.7%)	66.93 (-6.3%)	78.65 (-2.5%)
DVG ($D = 40\%$)	25	**264.95**	**7.35**×	**81.59 (-1.7%)**	**74.33 (+4.1%)**	**80.14 (-0.6%)**

4.1 Results on Text-to-Video Generation

As summarized in Table 1, under the standard 50-step setting and density 50%, DVG delivers a **3.20**× speedup while maintaining a high total score of **80.45**, outperforming all single-dimension compression and caching methods with even better efficiency. Under more aggressive compression, DVG further reaches **7.35**× speedup with a total score of **80.14** using only 25 sampling steps, surpassing all baselines and demonstrating strong robustness with fewer-step generation. Visualization in Fig. 4 further demonstrates DVG's superior efficiency while preserving quality.

Results on Wan2.2 and HunyuanVideo-1.5 are summarized in Table 2. On the MoE-based Wan2.2, DVG achieves **3.76**× speedups while maintaining less than 1% quality drop. On HunyuanVideo-1.5, it consistently outperforms both baseline and SpargeAttention. With density 70%, it achieves the best total score of **80.51**. Under stronger acceleration, DVG still maintains a high total score of **79.77** at **6.58**× speedup. These results demonstrate that DVG generalizes well across different architectures while providing a favorable efficiency and quality.

Table 2: **Quantitative comparison of text-to-video generation** on Wan2.2 and HunyuanVideo-1.5.

Method	NFE	Acceleration		VBench Score (%)		
		Latency(s) ↓	Speed ↑	Quality ↑	Semantic ↑	Total ↑
Wan2.2-14B [37]	40	460.27	1.00×	**85.21** (+0.0%)	**73.37** (+0.0%)	**82.84** (+0.0%)
DVG ($D = 50\%$)	40	200.85	2.29×	85.02 **(-0.2%)**	72.54 **(-1.1%)**	82.53 **(-0.4%)**
DVG ($D = 50\%$)	25	**122.31**	**3.76**×	84.61 **(-0.7%)**	72.33 **(-1.9%)**	82.15 **(-0.8%)**
HunyuanVideo-1.5 [38]	50	285.59	1.00×	83.19 (+0.0%)	62.05 (+0.0%)	78.96 (+0.0%)
SpargeAttn [44]	50	186.00	1.53×	83.83 (+0.8%)	66.52 (+7.2%)	80.36 (+1.8%)
DVG ($D = 70\%$)	50	136.73	2.09×	**84.22 (+1.2%)**	**65.69 (+5.9%)**	**80.51 (+2.0%)**
DVG ($D = 50\%$)	50	91.95	3.11×	84.05 **(+1.0%)**	65.83 **(+6.1%)**	80.41 **(+1.8%)**
DVG ($D = 50\%$)	25	**43.42**	**6.58**×	83.39 **(+0.2%)**	65.26 **(+5.2%)**	79.77 **(+1.0%)**

4.2 Results on Image-to-Video Generation

DVG consistently achieves strong performance on image-to-video generation, as shown in Table 3. Under the CFG-distilled setting, it delivers a 3.79× speedup while achieving **23.03** PSNR and **0.101** LPIPS, indicating high similarity to the original video. When applying with R-MeanFlow, DVG further reaches up to **18.09**× speedup, while still achieving excellent perceptual quality (**23.92** PSNR, **0.081** LPIPS) and the highest I2V score. Compared to other step-reduction and single-dimension compression methods, DVG achieves significantly better reconstruction quality while providing the highest additional speedup, demonstrating strong compatibility with distillation, as shown in Fig. 5.

Table 3: **Quantitative comparison of image-to-video generation** on HunyuanVideo-1.5.

Method	NFE	Acceleration		VBench Score (%)		Perceptual Metrics		
		Latency(s) ↓	Speed ↑	I2V ↑	Total ↑	PSNR ↑	SSIM ↑	LPIPS ↓
HunyuanVideo-1.5 [38]	100	540.04	1.00×	94.13	88.45	-	-	-
CFG Distilled [53]	50	278.37	1.94×	93.56	87.15	∞	1.000	0.000
+ VGDFR [40]	50	176.48	3.06×	91.75	85.59	19.37	0.683	0.145
+ AdaCache [14]	50	137.76	3.92×	91.88	85.71	19.95	0.654	0.183
+ Jenga-ProRes [45]	50	199.28	2.71×	92.35	86.38	19.87	0.663	0.157
+ Bottleneck [36]	30	132.69	4.07×	91.23	86.11	18.86	0.681	0.201
+ **DVG** ($D = 70\%$)	50	142.47	3.79×	**92.75**	**86.74**	**23.03**	**0.710**	**0.101**
+ **DVG** ($D = 50\%$)	50	**99.82**	**5.41**×	92.86	86.47	22.52	0.700	0.117
R-MeanFlow [53]	12	75.23	7.18×	94.10	87.34	∞	1.000	0.000
+ VGDFR [40]	12	55.33	9.76×	91.97	85.25	19.63	0.727	0.189
+ Fresco [50]	12	52.48	10.29×	91.85	84.88	18.78	0.662	0.365
+ AdaCache [14]	12	52.02	10.38×	91.79	85.21	19.08	0.707	0.208
+ TaylorSeer [20]	12	46.76	11.55×	92.03	85.39	19.34	0.714	0.205
+ **DVG** ($D = 70\%$)	12	39.18	13.78×	**93.60**	**86.47**	**23.92**	**0.767**	**0.081**
+ **DVG** ($D = 50\%$)	12	**29.86**	**18.09**×	93.51	86.42	23.73	0.766	0.083

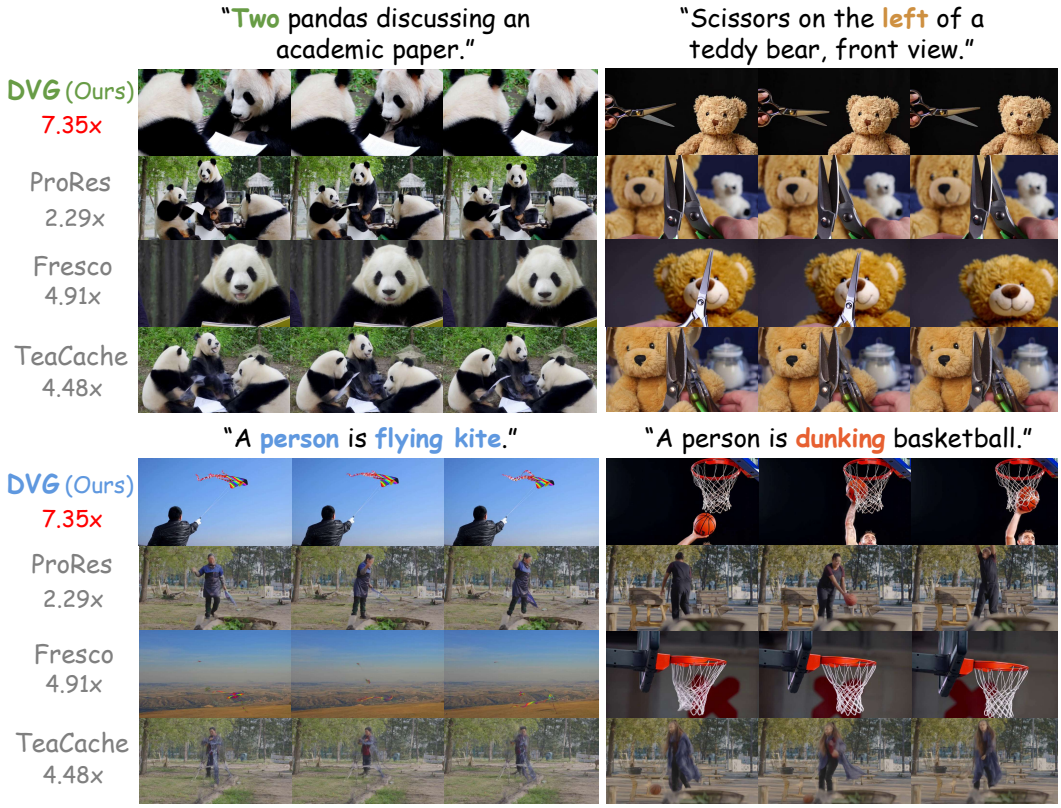


Figure 4: **Visualization of different acceleration methods on HunyuanVideo.** DVG achieves better semantic alignment and visual quality, whereas prior methods degrade under high acceleration.

5 Discussion

Ablation Study. We conduct our ablation study on HunyuanVideo-1.5. Table 4 demonstrates that the effectiveness of DVG comes primarily from its adaptive spatial-temporal acceleration design. Compressing only a single dimension leads to clear drops under the same overall density. Moreover, replacing our upsampling with linear interpolation or using random renoise consistently degrades performance, indicating that our tailored upsampling and renoise designs are essential for enabling flexible spatio-temporal compression while maintaining generation consistency. We further compare

Table 4: **Ablation Study** on HunyuanVideo-1.5.

Method	Density	Avg. Reduction Ratio		Acceleration		VBench Score (%)		
		Spatial ↓	Temporal ↓	Latency(s) ↓	Speed ↑	Dynamic ↑	Quality ↑	Total ↑
DVG (Ours)	~50%	0.29	0.46	91.95	3.11×	43.75	84.05	80.41
+ Temporal Only	~50%	0.00	0.71	93.03	3.07×	35.00	**85.13**	78.98
+ Spatial Only	~50%	0.49	0.00	89.54	3.19×	**47.25**	82.16	76.84
+ Manual Tuning (Fixed Ratio)	~50%	0.30	0.45	88.13	3.24×	41.50	83.05	78.85
+ Linear Upsampling	~50%	0.29	0.46	90.25	3.16×	43.00	79.92	79.11
+ Random Renoise	~50%	0.29	0.46	89.37	3.19×	43.50	78.37	78.92
+ Pixel-Space Eval	~50%	0.31	0.40	102.19	2.79×	43.75	84.83	**80.63**

DVG with manually tuned fixed compression ratios. Although the density budget is similar, DVG achieves significantly better performance through its content-aware allocation strategy, while manually tuned settings often produce inconsistent visual quality across different videos.



Figure 5: **Visualization of DVG on HunyuanVideo-1.5.** DVG successfully maintains high-fidelity and motion consistency with the original video, even under distillation, reaching up to 18× speedup.

Latent or Pixel Evaluation? We compare two strategies for estimating videos' spatial and temporal demand: evaluating directly in latent space versus decoding to pixel space. Across 1118 prompts, the two strategies achieve a Top-1 agreement of 85% and a Top-2 agreement of 99%. This suggests that discrepancies arise mainly because of the rich action set, rather than fundamentally different demand estimations. As shown in Table 4, pixel-space evaluation achieves slightly higher VBench scores, but our latent-space evaluation requires only **0.01s**, because it avoids an additional VAE decoding, thereby significantly reducing end-to-end latency. This advantage becomes increasingly important when combined with other acceleration techniques. For example, in a highly optimized setting (e.g., R-MeanFlow + DVG), generation takes only **29s**; introducing an extra VAE decode would increase the total latency to **40s**, resulting in a **38%** extra overhead.

6 Conclusion

In this paper, we revisit efficient video generation from a joint spatial-temporal perspective, showing that accelerating video generation requires balancing both dimensions rather than compressing only one. Based on this observation, we present **DVG**, a dynamic video generation framework that adaptively allocates computation across space and time under a target compute budget. By estimating content demands directly in latent space and matches the best compression action, DVG achieves strong acceleration while maintaining generation quality across large scale dense models, MoE models, and suits well in modern efficient video generation systems.

Acknowledgment

This paper was supported by CCF-Tencent Rhino-Bird Funds.

References

[1] I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The long-document transformer, 2020. URL https://arxiv.org/abs/2004.05150.

[2] D. Bolya and J. Hoffman. Token merging for fast stable diffusion, 2023. URL https://arxiv.org/abs/2303.17604.

[3] D. Bolya and J. Hoffman. Token merging for fast stable diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4599–4603, 2023.

[4] D. Bolya, C.-Y. Fu, X. Dai, P. Zhang, C. Feichtenhofer, and J. Hoffman. Token merging: Your vit but faster, 2023. URL https://arxiv.org/abs/2210.09461.

[5] J. Chen, C. Ge, E. Xie, Y. Wu, L. Yao, X. Ren, Z. Wang, P. Luo, H. Lu, and Z. Li. Pixart-σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation, 2024.

[6] J. Chen, J. Yu, C. Ge, L. Yao, E. Xie, Y. Wu, Z. Wang, J. Kwok, P. Luo, H. Lu, and Z. Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *International Conference on Learning Representations*, 2024.

[7] X. Cheng, Z. Chen, and Z. Jia. Cat pruning: Cluster-aware token pruning for text-to-image diffusion models, 2025. URL https://arxiv.org/abs/2502.00433.

[8] R. Child, S. Gray, A. Radford, and I. Sutskever. Generating long sequences with sparse transformers, 2019. URL https://arxiv.org/abs/1904.10509.

[9] G. Fang, X. Ma, and X. Wang. Structural pruning for diffusion models. *arXiv preprint arXiv:2305.10924*, 2023.

[10] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[11] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans. Cascaded diffusion models for high fidelity image generation, 2021. URL https://arxiv.org/abs/2106.15282.

[12] Z. Huang, Y. He, J. Yu, F. Zhang, C. Si, Y. Jiang, Y. Zhang, T. Wu, Q. Jin, N. Chanpaisit, Y. Wang, X. Chen, L. Wang, D. Lin, Y. Qiao, and Z. Liu. VBench: Comprehensive Benchmark Suite for Video Generative Models, Nov. 2023. URL http://arxiv.org/abs/2311.17982. arXiv:2311.17982 [cs].

[13] W. Jeong, K. Lee, H. Seo, and S. Y. Chun. Training-free mixed-resolution latent upsampling for spatially accelerated diffusion transformers, 2026. URL https://arxiv.org/abs/2507.08422.

[14] K. Kahatapitiya, H. Liu, S. He, D. Liu, M. Jia, C. Zhang, M. S. Ryoo, and T. Xie. Adaptive caching for faster video generation with diffusion transformers, 2024. URL https://arxiv.org/abs/2411.02397.

[15] M. Kim, S. Gao, Y.-C. Hsu, Y. Shen, and H. Jin. Token fusion: Bridging the gap between token pruning and token merging. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1383–1392, 2024.

[16] S. Kim, H. Lee, W. Cho, M. Park, and W. W. Ro. Ditto: Accelerating diffusion model via temporal value similarity. In *Proceedings of the 2025 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2025.

[17] H. Li, Y. Yang, M. Chang, H. Feng, Z. Xu, Q. Li, and Y. Chen. Srdiff: Single image super-resolution with diffusion probabilistic models, 2021. URL https://arxiv.org/abs/2104.14951.

[18] X. Li, Y. Liu, L. Lian, H. Yang, Z. Dong, D. Kang, S. Zhang, and K. Keutzer. Q-diffusion: Quantizing diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17489–17499, 2023. doi: 10.1109/ICCV51070.2023.01608.

[19] F. Liu, S. Zhang, X. Wang, Y. Wei, H. Qiu, Y. Zhao, Y. Zhang, Q. Ye, and F. Wan. Timestep embedding tells: It's time to cache for video diffusion model, 2025. URL https://arxiv.org/abs/2411.19108.

[20] J. Liu, C. Zou, Y. Lyu, J. Chen, and L. Zhang. From reusing to forecasting: Accelerating diffusion models with taylorseers, 2025. URL https://arxiv.org/abs/2503.06923.

[21] X. Liu, C. Gong, et al. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023.

[22] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

[23] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022.

[24] W. Lu, S. Zheng, Y. Xia, and S. Wang. Toma: Token merge with attention for diffusion models, 2025. URL https://arxiv.org/abs/2509.10918.

[25] Z. Luo, D. Chen, Y. Zhang, Y. Huang, L. Wang, Y. Shen, D. Zhao, J. Zhou, and T. Tan. Videofusion: Decomposed diffusion models for high-quality video generation, 2023. URL https://arxiv.org/abs/2303.08320.

[26] D. Menn, Y. Yang, B. Wang, X. Wei, M. Munir, F. Liang, R. Marculescu, C. Xu, and D. Marculescu. Video compression meets video generation: Latent inter-frame pruning with attention recovery, 2026. URL https://arxiv.org/abs/2603.05811.

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

[28] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[29] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi. Image super-resolution via iterative refinement, 2021. URL https://arxiv.org/abs/2104.07636.

[30] T. Salimans and J. Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

[31] Y. Shang, Z. Yuan, B. Xie, B. Wu, and Y. Yan. Post-training quantization on diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1972–1981, 2023.

[32] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[33] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[34] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever. Consistency models. In *International Conference on Machine Learning*, pages 32211–32252. PMLR, 2023.

[35] X. Sun et al. Hunyuanvideo: A systematic framework for large video generation models, 2024. URL https://arxiv.org/abs/2412.03603.

[36] Y. Tian, X. Xia, Y. Ren, S. Lin, X. Wang, X. Xiao, Y. Tong, L. Yang, and B. Cui. Training-free diffusion acceleration with bottleneck sampling, 2025. URL https://arxiv.org/abs/2503.18940.

[37] T. Wan, A. Wang, B. Ai, B. Wen, C. Mao, C.-W. Xie, D. Chen, F. Yu, H. Zhao, J. Yang, J. Zeng, J. Wang, J. Zhang, J. Zhou, J. Wang, J. Chen, K. Zhu, K. Zhao, K. Yan, L. Huang, M. Feng, N. Zhang, P. Li, P. Wu, R. Chu, R. Feng, S. Zhang, S. Sun, T. Fang, T. Wang, T. Gui, T. Weng, T. Shen, W. Lin, W. Wang, W. Wang, W. Zhou, W. Wang, W. Shen, W. Yu, X. Shi, X. Huang, X. Xu, Y. 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, Z. Jiang, Z. Han, Z.-F. Wu, and Z. Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[38] B. Wu, C. Zou, C. Li, D. Huang, F. Yang, H. Tan, J. Peng, J. Wu, J. Xiong, J. Jiang, Linus, Patrol, P. Zhang, P. Chen, P. Zhao, Q. Tian, S. Liu, W. Kong, W. Wang, X. He, X. Li, X. Deng, X. Zhe, Y. Li, Y. Long, Y. Peng, Y. Wu, Y. Liu, Z. Wang, Z. Dai, B. Peng, C. Li, G. Gong, G. Xiao, J. Tian, J. Lin, J. Liu, J. Zhang, J. Lian, K. Pan, L. Wang, L. Niu, M. Chen, M. Chen, M. Zheng, M. Yang, Q. Hu, Q. Yang, Q. Xiao, R. Wu, R. Xu, R. Yuan, S. Sang, S. Huang, S. Gong, S. Huang, W. Guo, X. Yuan, X. Chen, X. Hu, W. Sun, X. Wu, X. Ren, X. Yuan, X. Mi, Y. Zhang, Y. Sun, Y. Lu, Y. Li, Y. Huang, Y. Tang, Y. Li, Y. Deng, Y. Zhou, Z. Hu, Z. Liu, Z. Yang, Z. Yang, Z. Lu, Z. Zhou, and Z. Zhong. Hunyuanvideo 1.5 technical report, 2025. URL https://arxiv.org/abs/2511.18870.

[39] Z. Yang, J. Teng, W. Zheng, M. Ding, S. Huang, J. Xu, Y. Yang, W. Hong, X. Zhang, G. Feng, D. Yin, X. Gu, Yuxuan.Zhang, W. Wang, Y. Cheng, B. Xu, Y. Dong, and J. Tang. Cogvideox: Text-to-video diffusion models with an expert transformer. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=LQzN6TRFg9.

[40] Z. Yuan, R. Xie, Y. Shang, H. Zhang, S. Wang, S. Yan, G. Dai, and Y. Wang. Vgdfr: Diffusion-based video generation with dynamic latent frame rate, 2025. URL https://arxiv.org/abs/2504.12259.

[41] M. Zaheer, G. Guruganesh, A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, and A. Ahmed. Big bird: Transformers for longer sequences, 2021. URL https://arxiv.org/abs/2007.14062.

[42] E. Zhang, B. Xiao, J. Tang, Q. Ma, C. Zou, X. Ning, X. Hu, and L. Zhang. Token pruning for caching better: 9 times acceleration on stable diffusion for free, 2024. URL https://arxiv.org/abs/2501.00375.

[43] E. Zhang, J. Tang, X. Ning, and L. Zhang. Training-free and hardware-friendly acceleration for diffusion models via similarity-based token pruning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

[44] J. Zhang, C. Xiang, H. Huang, J. Wei, H. Xi, J. Zhu, and J. Chen. Spargeattention: Accurate and training-free sparse attention accelerating any model inference, 2025. URL https://arxiv.org/abs/2502.18137.

[45] Y. Zhang, J. Xing, B. Xia, S. Liu, B. Peng, X. Tao, P. Wan, E. Lo, and J. Jia. Training-free efficient video generation via dynamic token carving, 2025. URL https://arxiv.org/abs/2505.16864.

[46] K. Zheng, C. Lu, J. Chen, and J. Zhu. DPM-solver-v3: Improved diffusion ODE solver with empirical model statistics. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=9fWKExmKa0.

[47] K. Zheng, Y. Wang, Q. Ma, H. Chen, J. Zhang, Y. Balaji, J. Chen, M.-Y. Liu, J. Zhu, and Q. Zhang. Large scale diffusion distillation via score-regularized continuous-time consistency, 2026. URL https://arxiv.org/abs/2510.08431.

[48] S. Zheng, G. Chen, Q. Zhou, Y. Lin, L. He, C. Zou, P. Cai, J. Liu, and L. Zhang. Let features decide their own solvers: Hybrid feature caching for diffusion transformers, 2025. URL https://arxiv.org/abs/2510.04188.

[49] S. Zheng, L. Feng, X. Wang, Q. Zhou, P. Cai, C. Zou, J. Liu, Y. Lin, J. Chen, Y. Ma, and L. Zhang. Forecast then calibrate: Feature caching as ode for efficient diffusion transformers, 2025. URL https://arxiv.org/abs/2508.16211.

[50] S. Zheng, G. Chen, L. He, J. Liu, Y. Lin, C. Zou, and L. Zhang. From sketch to fresco: Efficient diffusion transformer with progressive resolution, 2026. URL https://arxiv.org/abs/2601.07462.

[51] Z. Zheng, X. Peng, T. Yang, C. Shen, S. Li, H. Liu, Y. Zhou, T. Li, and Y. You. Open-sora: Democratizing efficient video production for all, March 2024. URL https://github.com/hpcaitech/Open-Sora.

[52] H. Zhu, D. Tang, J. Liu, M. Lu, J. Zheng, J. Peng, D. Li, Y. Wang, F. Jiang, L. Tian, S. Tiwari, A. Sirasao, J.-H. Yong, B. Wang, and E. Barsoum. Dip-go: A diffusion pruner via few-step gradient optimization, 2024.

[53] C. Zou, C. Li, Y. Li, P. Li, J. Wu, X. He, S. Liu, Z. Zhong, K. Huang, and L. Zhang. Disca: Accelerating video diffusion transformers with distillation-compatible learnable feature caching, 2026. URL https://arxiv.org/abs/2602.05449.

A Appendix

A.1 Detailed Experiment Settings

A.1.1 Model Configurations.

HunyuanVideo T2V. For the HunyuanVideo text-to-video backbone, we evaluate our method on 720×1280 resolution, 125 frames, with 50 denoising steps and a fixed random seed of 42. The sampler follows the default configuration, using an embedded CFG scale of 6.0, a flow shift of 7.0, and reverse-flow sampling.

HunyuanVideo-1.5 T2V. For the HunyuanVideo-1.5 text-to-video backbone, videos are generated from the VBench text prompt at the recommended 480p resolution with a $16{:}9$ aspect ratio and 121 frames. We use 50 denoising steps, a fixed random seed of 42, `bf16` precision, and a flow shift of 7.0.

HunyuanVideo-1.5 I2V. For the HunyuanVideo-1.5 image-to-video backbone, we use the CFG-distilled `480p_i2v_distilled` checkpoint. Videos are generated from the reference image and text prompt at the recommended 480p resolution with a $16{:}9$ aspect ratio and 121 frames. We use 50 denoising steps, a fixed random seed of 42, `bf16` precision, and a flow shift of 7.0.

Wan2.2 T2V. For the Wan2.2 text-to-video backbone, we use the MoE-based `Wan2.2-T2V-A14B` checkpoint. We evaluate on VBench using the default 480p configuration, with a resolution of 832×480, 81 frames, and 40 denoising steps. We use a fixed random seed of 42 and follow the repository default mixed-precision setup, where both the T5 encoder and diffusion model use `bf16` weights. Sampling follows the default Wan2.2 T2V configuration with the UniPC sampler, a sample shift of 12.0, and a two-stage classifier-free guidance scale of $(3.0, 4.0)$, with the expert boundary set to 0.875. For large-scale evaluation, inference is distributed with FSDP and DeepSpeed Ulysses using 2 GPUs per worker.

A.1.2 DVG Inference Settings.

For our method, we use a Three-stage dynamic inference schedule in our experiments. The first stage starts from a fixed coarse configuration, where the spatial resolution is simply set to 0.5 of the target resolution and the temporal length is set to 13 latent frames, corresponding to approximately 42% of the full 31-latent-frame video. The final stage restores the full target resolution and frame count. The intermediate stages are determined automatically according to our action matching algorithm.

A.1.3 Benchmarks.

VBench. For text-to-video evaluation, we use the official VBench [12] protocol with the full prompt set from `VBench_full_info.json`, which contains 946 prompt entries. We evaluate all VBench dimensions used in our setup, including subject consistency, imaging quality, background consistency, motion smoothness, overall consistency, human action, multiple objects, spatial relationship, object class, color, aesthetic quality, appearance style, temporal flickering, scene, temporal style, and dynamic degree. We report the normalized VBench scores, including the aggregated Quality Score, Semantic Score, and Total Score.

VBench-I2V. For image-to-video evaluation, we use the official VBench-I2V protocol with the full prompt set from `vbench2_i2v_full_info.json`, which contains 1118 image-prompt pairs. We evaluate all VBench-I2V dimensions used in our setup, including i2v subject, i2v background, camera motion, subject consistency, background consistency, aesthetic quality, imaging quality, temporal flickering, motion smoothness, and dynamic degree. We report the normalized VBench-I2V scores together with the aggregated Quality Score, I2V Score, and Total Score.

For the aggregated scores, the Quality Score measures intrinsic visual-temporal quality, while the Semantic Score in VBench measures text-video alignment. In VBench-I2V, the I2V Score measures image-conditioned generation faithfulness, including the consistency of the generated video with the input image and the camera-motion instruction. The Total Score is the weighted aggregate of the corresponding normalized scores. Among individual dimensions, imaging quality captures

frame-level artifacts such as blur, noise, and over-exposure, while dynamic degree measures whether the generated video contains sufficient motion.

A.2 Timing Breakdown.

We further report the timing breakdown of our method on HunyuanVideo-1.5 I2V at the 480p setting with 121 frames, using 50 denoising steps and a density budget of 0.5, measured on an NVIDIA RTX PRO 6000 GPU. The total generation time is 99.04s, where the preparation before entering the denoising loop takes 0.36s. The early sketch stage takes 4.18s. The action decision module, including clean latent prediction, demand calculation, and best-action scoring, takes only 0.01s in total, while latent upsampling takes 0.003s. The spatial-temporally compressed stages take 31.20s in total. The refinement stage, which runs at the full target resolution and frame rate, takes 53.77s. The final VAE decoding step takes 9.51s. This breakdown shows that the additional overhead introduced by action selection and upsampling is negligible compared with the main denoising computation.

A.3 Limitations.

Although DVG demonstrates strong acceleration across multiple video generation models and tasks, it still has several limitations. First, our current implementation relies on a discrete set of spatio-temporal compression actions. While this design makes the acceleration ratio controllable and avoids additional training, a richer or continuous learnable action space may further improve the quality-efficiency trade-off for more diverse video contents. Second, DVG estimates spatial and temporal demands from lightweight latent-space cues, which are efficient and generally effective, but may not fully capture all perceptual factors in highly complex scenes, such as subtle object interactions or fine-grained camera motion. Finally, our experiments focus on representative open-source video diffusion models and standard benchmarks; extending the evaluation to more architectures, longer videos, and additional real-world deployment settings would further validate the generality of the framework. We leave these directions for future work.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims match the contributions and experimental scope presented in the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a separate discussion of limitations in appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not present new theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides the experimental setup, baselines, evaluation metrics, and implementation details needed to reproduce the main results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code is in supplementary material and will be released on Github.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the models, datasets, baselines, metrics, and implementation details used in the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper reports quantitative experimental results and ablations, but does not currently include error bars, confidence intervals, or statistical significance tests for the main results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper reports the main compute resources used in our experiments, including RTX PRO 6000 GPUs and the associated vCPU resources.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The paper does not include a separate broader impact discussion, as the work focuses on a general inference acceleration method.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk models or datasets that require additional safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper credits the existing datasets, models, codebases, and evaluation tools used in the experiments, and describes their licenses or terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The released code is documented with usage instructions and implementation details.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: The paper includes a small-scale user study involving voluntary participants who completed anonymous electronic questionnaires for perceptual evaluation of generated videos. The study did not collect sensitive personal information and posed minimal risk to participants. According to our institutional guidelines, this type of anonymous perceptual evaluation does not require formal IRB approval.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [No]

    Justification: Participants were informed of the scope of collected information and its intended use, but the paper does not report IRB approval or an equivalent approval.

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method does not involve LLMs as an important, original, or non-standard component.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.