FRAME GUIDANCE: TRAINING-FREE GUIDANCE FOR FRAME-LEVEL CONTROL IN VIDEO DIFFUSION MODEL

Anonymous authors
Paper under double-blind review

Figure 1: Frame Guidance enables training-free controllable video generation using flexible frame-level inputs. It supports diverse applications, including keyframe-guided generation, stylization, and looping, using general frame-level inputs such as depth maps, sketches, and color blocks.

ABSTRACT

Advancements in diffusion models have significantly improved video quality, directing attention to fine-grained controllability. However, many existing methods depend on fine-tuning large-scale video models for specific tasks, which becomes increasingly impractical as model sizes continue to grow. In this work, we present Frame Guidance, a training-free guidance for controllable video generation based on frame-level signals, such as keyframes, style reference images, sketches, or depth maps. By applying guidance to only a few selected frames, Frame Guidance can steer the generation of the entire video, resulting in a temporally coherent controlled video. To enable training-free guidance on large-scale video models, we propose a simple latent processing method that dramatically reduces memory usage, and apply a novel latent optimization strategy designed for globally coherent video generation. Frame Guidance enables effective control across diverse tasks, including keyframe guidance, stylization, and looping, without any training, and is compatible with any models. Experimental results show that Frame Guidance can produce high-quality controlled videos for a wide range of tasks and input signals.

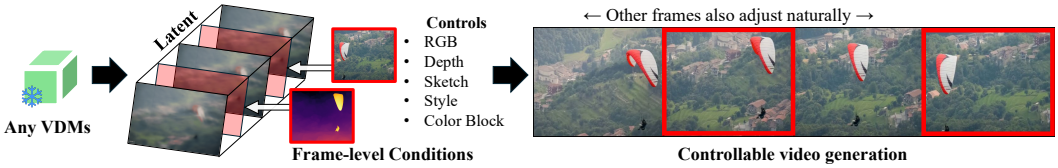


Figure 2: **Frame Guidance** steers the video generation process of a VDM by applying gradient-based guidance to selected frames, resulting in a temporally coherent controlled video. Our method is training-free, model-agnostic, and supports a wide range of frame-level conditions.

1 Introduction

The rapid advancement of diffusion models (Ho et al., 2020; Song et al., 2021; Lipman et al., 2022) has led to the development of powerful video generation models. Recent large-scale video diffusion models (VDMs) have made significant progress in high-quality text-to-video (T2V) and image-to-video (I2V) generation, which are capable of generating diverse and realistic video content (Brooks et al., 2024; Polyak et al., 2025; Yang et al., 2025; Wang et al., 2025a). With ongoing advancements, there is a growing interest in enabling more fine-grained control over the generation process.

Recent progress underscores the need for a practical approach to controllable video generation. Hence, we identify two major desiderata: (1) a *model-agnostic, training-free* framework, and (2) a *general-purpose guidance* method. Existing methods (Burgert et al., 2025; He et al., 2025; Li et al., 2025b) typically fine-tune large-scale VDMs (Yang et al., 2025; Wang et al., 2025a) for each specific control task, which is increasingly impractical due to high computational cost and the burden of retraining with every new model release. This highlights the need for training-free guidance methods that work across models. Moreover, end users prefer simple, generalizable frameworks that support diverse tasks and inputs, such as reference images, depth maps, or sketches, rather than task-specific models (Hou et al., 2024; Wang et al., 2025b) that are restricted to a fixed input type.

Existing methods fall short of satisfying both desiderata *simultaneously*: training-free approaches (Ling et al., 2025; Hou et al., 2024) are often task-specific and lack generalizability, while general-purpose methods (Li et al., 2025b; Jiang et al., 2025) require fine-tuning and need substantial training resources. Many existing methods (Wang et al., 2025b; 2024; Bai et al., 2025) are both task-specific and training-dependent, making them difficult to adapt to new models or tasks.

In this work, we propose Frame Guidance, a novel guidance method for VDMs that is model-agnostic, training-free, and supports a wide range of controllable video generation tasks using frame-level signals. As illustrated in Figure 2, Frame Guidance steers the video generation process by applying guidance to selected frames based on frame-level signals, which produce temporally coherent videos.

We present two core components for effective and flexible frame-level guidance. First, we introduce *latent slicing*, a simple latent decoding technique that enables efficient training-free guidance for large-scale VDMs. Based on temporally local patterns of video encoding, we propose to decode only the short temporal slices of the video latent for computing the guidance loss. Furthermore, we present *video latent optimization* (VLO), a novel latent update strategy designed for precise control of the video diffusion process. As the overall layout of the frames is largely determined in the first few inference steps (Wu et al., 2024a), we apply deterministic optimization at the early stages for globally coherent layout, and employ stochastic optimization until the mid-stage for refining the details.

Frame Guidance is applicable to general frame-level control tasks, as shown in Figure 1, including keyframe-guided generation, stylized video generation, and looped video generation. In particular, Frame Guidance supports general input conditions, such as depth maps, sketches, and color blocks. We demonstrate that Frame Guidance consistently produces superior results on frame-level control tasks across various VDMs (Yang et al., 2025; HaCohen et al., 2024; Wang et al., 2025a).

2 Related Work

Training-required controllable video generation Advances in T2V and I2V generation have opened up new opportunities for fine-grained user control. These include conditioning on keyframes (Zeng et al., 2024; Wang et al., 2025b), using style reference images for stylized generation (Liu et al., 2023; Wang et al., 2023a), and incorporating trajectory-based signals such as camera movement (Zheng et al., 2024; Bai et al., 2025) or motion trajectory (Wu et al., 2024b; Namekata et al., 2025) for dynamic scene generation. However, existing methods often require extensive training

and model-specific data preparation, such as fixed resolution or frame counts, making fine-tuning increasingly impractical for general users as model sizes and resource requirements continue to grow.

Training-free controllable video generation To reduce the burden of training large models, several approaches have explored training-free controllable video generation (Li et al., 2025a; Ling et al., 2025; Hou et al., 2024; Wu et al., 2023; Zhang et al., 2024; Khachatryan et al., 2023; Geyer et al., 2024). For example, CamTrol (Hou et al., 2024) enables camera control using external 3D point clouds, while MotionClone (Ling et al., 2025) performs motion cloning based on temporal attention maps extracted from a reference video, and Tune-A-Video (Wu et al., 2023) enables video editing with image diffusion models. However, these methods are tailored to specific tasks and are thereby ill-suited for more general scenarios requiring different types or even multiple input signals. In this work, we propose a training-free guidance method that generalizes to a wide range of video generation tasks using frame-level signals.

3 PRELIMINARIES

Video diffusion models (VDMs) Recent video diffusion models (Brooks et al., 2024; Yang et al., 2025; HaCohen et al., 2024; Wang et al., 2025a) learn to generate video by reversing the noising process in the latent space. The high-dimensional video x_0 is encoded into a lower-dimensional latent $z_0 = \mathcal{E}(x_0)$. The forward noising process corrupts the latent $z_t = \sqrt{\bar{\alpha}_t} z_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ and $\{\bar{\alpha}_t\}_{t\in[0,T]}$ is a pre-defined noise schedule. The reverse denoising process is learned through predicting a time-dependent velocity $v_t = \sqrt{\bar{\alpha}_t}\epsilon - \sqrt{1-\bar{\alpha}_t} z_0$, which represents the direction from a noisy sample toward the clean sample (Salimans and Ho, 2022). For each time step t, the clean sample $z_{0|t}$ can be computed from the noisy sample z_t using Tweedie's formula (Efron, 2011):

$$z_{0|t} \coloneqq \mathbb{E}[z_0|z_t] = \sqrt{\bar{\alpha}_t} z_t - \sqrt{1-\bar{\alpha}_t} \cdot v_\theta(z_t, t), \tag{1}$$

where v_θ is the predicted velocity. Latents z_0 are decoded into videos with the decoder $\hat{x}_0 = \mathcal{D}(z_0)$.

Recent large-scale VDMs (Wang et al., 2025a; Yang et al., 2025) commonly employ spatio-temporal VAEs to encode high-dimensional video data. A notable example is the CausalVAE (Yu et al., 2024; Brooks et al., 2024), which enforces *temporal causality* in the latent space by allowing only past frames to influence future ones. While this design encourages temporally coherent video generation, it also introduces temporal dependencies within the latent sequence, requiring the entire sequence to be decoded even to reconstruct a single frame.

Training-free guidance Training-free guidance (Bansal et al., 2024; Yu et al., 2023; Rout et al., 2025; Shen et al., 2024) uses pre-trained diffusion models to generate samples that satisfy a specific condition, without additional training. At each denoising step t, it estimates a clean image $x_{0|t} = \mathcal{D}(z_{0|t})$ from the current latent z_t, and computes a guidance loss $\mathcal{L}_e(\mathcal{D}(z_{0|t}), c)$ that measures alignment with the target control c. The latent z_t is then updated using the gradient $\nabla_{z_t}\mathcal{L}_e$ during inference. One such strategy is the time-travel trick (Bansal et al., 2024; Yu et al., 2023; He et al., 2024), which alternates between denoising and renoising steps to correct accumulated errors.

4 METHOD

We present Frame Guidance, a simple yet effective training-free framework for controllable video generation using frame-level signals, designed to be compatible with modern large-scale VDMs. Our approach guides the generation process of pre-trained VDMs by optimizing video latents to minimize frame-level guidance loss applied to *selected frames*. In this section, we introduce two key components that enable efficient and flexible frame-level guidance for large-scale VDMs.

4.1 LATENT SLICING

The main challenge of training-free guidance on video generation is the computational constraint. To compute the guidance loss for latent optimization, we should keep track of the gradient chain passing through the whole network (Figure 3). In Figure 4(a), we analyze the memory usage and find that it exceeds 650GB even with gradient checkpointing (Chen et al., 2016), mostly due to CausalVAE (Yu et al., 2024; Brooks et al., 2024). This overhead arises from the design of CausalVAE, which requires

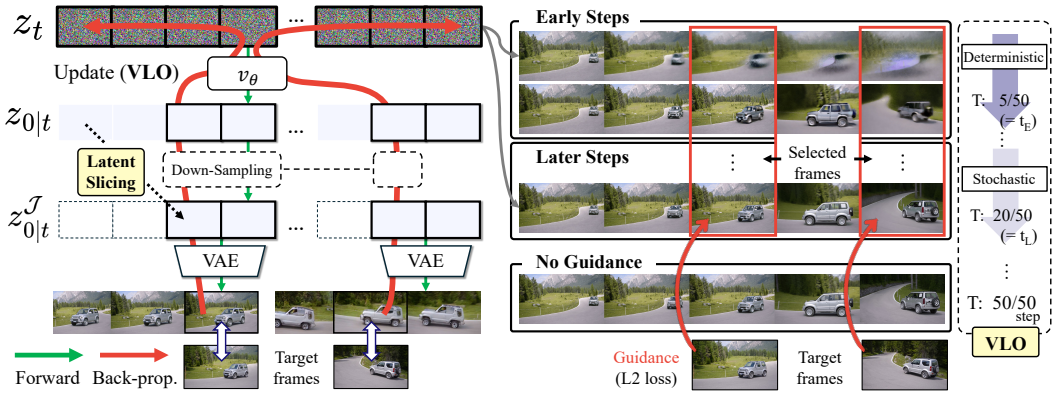


Figure 3: Frame Guidance for keyframe-guided video generation task. **(Left)** Illustration of our method with *latent slicing* and spatial down-sampling (Section 4.1), and gradient propagation with L2 loss (red arrows; Section 4.3). **(Right)** Visualization of the *video latent optimization* (VLO; Section 4.2), showing the generated video frames at each guided inference step.

decoding the *entire* latent sequence even to reconstruct a single frame. To tackle this, we first analyze the latent space of CausalVAE.

Analysis of CausalVAE's latent space While CausalVAE is designed to enforce temporal causality in the video latent sequence, we observe that such causality is absent in practice. To validate this, we conduct a simple experiment: replace a single frame in a real video with a black image (all pixels set to zero), and measure the difference between the latents of the original video and the modified video. As shown in Figure 4(b), the perturbation affects only a few consecutive latents rather than the entire sequence. This behavior consistently appears across various VDMs (Yang et al., 2025; Wang et al., 2025a; HaCohen et al., 2024). We refer to this property as *temporal locality*, a key observation for our efficient decoding method.

Decoding with sliced latent We introduce *latent slicing*, an essential decoding method for training-free guidance that significantly reduces the cost of gradient computation on CausalVAE. Instead of reconstructing the entire sequence, we decode only a few frames from the selected sliced latents. To be specific, when reconstructing the i-th frame x^i, we decode a small window of 3 latents, starting from the latent z^j, where the latent index j is determined by i and the temporal compression rate of its CausalVAE. Thanks to the temporal locality, it is sufficient to decode only the corresponding latents to reconstruct a single video frame. As shown in Figure 22, the reconstructed frames are nearly identical to those from full-sequence decoding. As highlighted in Figure 4(a), this latent slicing reduces memory usage by up to $15\times$ compared to using the entire latent sequence.

In parallel with latent slicing, we can further reduce the memory usage by spatially down-sampling the latents before decoding. Despite the lower resolution, the guidance loss from the down-sampled latents still provides sufficient signals to guide the generation. As shown in Figure 4(a), applying $2\times$ spatial down-sampling combined with latent slicing reduces memory usage by up to $60\times$, enabling gradient computation to be maintained on a single GPU even for large VDMs (Wang et al., 2025a).

4.2 Video Latent Optimization (VLO)

Previous training-free guidance methods for images (Bansal et al., 2024; Yu et al., 2023; Shen et al., 2024) typically *reintroduce noise* after a gradient update. However, in the video domain, we observe that this strategy often has adverse effects on guidance. The overall layout of the frames is largely determined during the early denoising steps (Wu et al., 2024a). Similarly, the influence of guidance is most significant on the overall layout in these stages. As shown in Figure 4(c) top, applying guidance to a single frame (yellow arrow) has a higher influence (dark green) on neighboring latents early on, with the effect diminishing later. This confirms that early-stage guidance is critical for temporal coherence. Yet, the noising scale at the early stage is often too large, *washing out* the guidance signal.

To address this limitation, we propose *video latent optimization* (VLO), a hybrid strategy that applies different update rules to video latents depending on the denoising stage. Specifically, at each denoising step t in the early stage, we update the latent z_t with guidance in a *deterministic* manner:

$$z_t \leftarrow z_t - \eta \nabla_{z_t} \mathcal{L}_e(x^{\mathcal{I}}_{0|t}, c_{\texttt{frames}}), \tag{2}$$

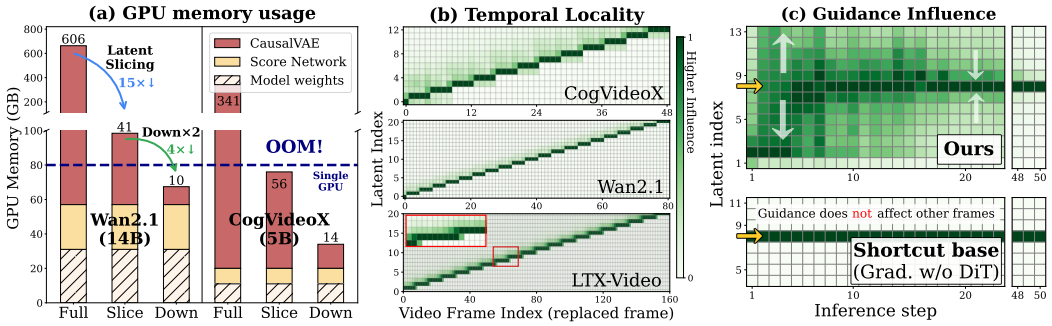


Figure 4: **(a) GPU memory for guidance** when using full latent sequence, sliced latents, and latent slicing with spatial down-sampling. **(b) Temporal locality** of CausalVAEs. Each latent (y-axis) is primarily affected by a small subset of temporally local video frames. **(c) Guidance influence** during the denoising steps. Yellow arrows indicate the location for the guidance frame.

where η is the guidance step size, $x^{\mathcal{I}}_{0|t}$ is the predicted clean frames where we apply guidance, and $\mathcal{L}_e$ is a guidance loss with frame-level controls $c_{\texttt{frames}}$. This deterministic update results in a temporally aligned global layout. In the later steps, we update the latent z_t in a stochastic manner by reintroducing noise in order to reduce accumulated errors during guidance, similar to the time-travel trick (Yu et al., 2023; Shen et al., 2024). This stage-aware procedure is illustrated in Figure 3 right. We show in Figure 23 that stochastic updates in the early steps fail to capture the desired layout, whereas our VLO successfully reflects the layout changes specified by the guidance frames.

4.3 Frame Guidance

In Algorithm 1, we provide the overall procedure of our Frame Guidance, which incorporates both the latent slicing and VLO. Given a set of frame-level controls $c_{\texttt{frames}}$ and selected frame indices $\mathcal{I} \subseteq \{i_1, \cdots\}$ to apply the guidance, we first compute their corresponding latent indices $\mathcal{J} \subseteq \{j_1, \cdots\}$ (see Figure 4(b)). For pre-defined generation phases t_E and t_L (we provide details on determining their values in Appendix C.5), we optimize the video latents in the following manner: At each denoising step $t > t_L$, we extract the sliced latents $z^{\mathcal{J}}_{0|t}$ from the latent indices $\mathcal{J}$ (Line 7) and compute the guidance loss $g_t = \nabla_{z_t}\mathcal{L}_e(x^{\mathcal{I}}_{0|t}, c_{\texttt{frames}})$ (Lines 8-9). We optimize the latent z_t using VLO (Line 11) where z_t is updated deterministically in the early denoising steps ($t > t_E$) and stochastically (Algorithm 2) during the later steps ($t_E \geq t > t_L$). After M times of latent optimization, we proceed to the next denoising step via DDIM Song et al. (2020). We provide detailed time-travel algorithm and Frame Guidance algorithm for flow matching based models, such as Wan (Wang et al., 2025a), in Appendix C.3.

Algorithm 1 Frame Guidance

Require: $\mathcal{I}$, t_E, t_L, repeat step M, step size η, guidance loss $\mathcal{L}_e$, model $v_\theta(\cdot,\cdot)$

1: $z_T \sim \mathcal{N}(0, I)$
2: $\mathcal{J} \leftarrow$ Frame-Idx-to-Latent-Idx($\mathcal{I}$)
3: **for** $t = T, ..., 1$ **do**
4: **if** $t > t_L$ **then** {Guidance step}
5: **for** $m = 1, ..., M-1$ **do**
6: $z_{0|t} \leftarrow \sqrt{\bar{\alpha}_t} z_t - \sqrt{1-\bar{\alpha}_t} \cdot v_\theta(z_t, t)$
7: $z^{\mathcal{J}}_{0|t} \leftarrow$ Latent-Slicing($z_{0|t}, \mathcal{J}$)
8: $x^{\mathcal{I}}_{0|t} \leftarrow \mathcal{D}(z^{\mathcal{J}}_{0|t})$
9: $g_t = \nabla_{z_t}\mathcal{L}_e(x^{\mathcal{I}}_{0|t}, c_{\texttt{frames}})$
10: **if** $t > t_E$ **then** {Early steps}
11: $z_t \leftarrow z_t - \eta g_t$
12: **else** {Later steps}
13: $z_t \leftarrow$ Time-Travel($z_t, z_{0|t}, g_t$)
14: **end if**
15: **end for**
16: **end if**
17: $z_{t-1} \leftarrow$ DDIM($z_t, z_{0|t}$)
18: **end for**
19: **return** z_0

Gradient propagation after slicing Without processing the full latent sequence, guidance applied to sliced latents can control the entire video, resulting in temporally coherent outputs. This coherence arises from the denoising network v_θ, which propagates the gradient of the guidance loss across the entire video latents. We show in Figure 4(c) bottom that excluding the denoising network when computing the gradient, i.e., shortcut-based update (He et al., 2024; Rout et al., 2025; Nair and Patel, 2024), restricts the gradients to the guided frame only (bottom), leading to a temporally disconnected video. On the other hand, using the denoising network propagates the gradients across all frames (top), allowing guidance on target frames to *harmonize* with other frames, as illustrated in Figure 3 (right). Therefore, guidance on a few frames where the gradient through the denoising network can

control the whole video, which enables tasks such as stylized video generation. In Appendix C.4, we further demonstrate that the temporal coherence is primarily determined by the denoising network, whereas the contribution of CausalVAE is minimal.

4.4 Loss design for various tasks

Frame Guidance is readily applicable to a wide range of frame-conditioned video generation tasks, with appropriately designed guidance loss. Here, we provide simple loss designs for representative frame-conditioned video generation tasks and general user inputs.

Keyframe-guided video generation aims to synthesize videos that transition smoothly between multiple user-specified keyframes, without enforcing strict pixel-level reconstruction. Given an initial image as the input to the I2V model, we minimize a simple *L2 loss*, $\mathcal{L}_e = \sum_{i\in\mathcal{I}} \|x^i_* - x^i_{0|t}\|_2^2$, where $x^{\mathcal{I}}_*$ denotes the target keyframes and $x^i_{0|t}$ is the predicted clean i-th frame. The similarity to each keyframe can be controlled by adjusting the guidance strength, such as the number of repeat steps M or step size η. Unlike training-based approaches (Zeng et al., 2024; Wang et al., 2025b) that are limited to fixed positions (e.g., the last frame), our method supports arbitrary keyframe placements.

Stylized video generation aims to synthesize videos in the style of a given reference image using a T2V model. We employ a differentiable style encoder Ψ to compute the *style loss* defined as $\mathcal{L}_e = -\sum_{i\in\mathcal{I}} \cos(\Psi(x_{\text{style}}), \Psi(x^i_{0|t}))$, where x_{style} is the style reference image. We use the Contrastive Style Descriptor (CSD) (Somepalli et al., 2024) for $\Psi(\cdot)$, and find that guiding only a few selected (or randomly chosen) frames is sufficient to propagate the desired style across the entire video.

Looped video generation aims to synthesize videos where the first and last frames match, producing a seamless loop using a T2V model. We define the loss as $\mathcal{L}_e = \|\,\text{sg}(x^1_{0|t}) - x^L_{0|t}\|_2^2$, where $\text{sg}(\cdot)$ denotes the *stop-gradient* operator. This design prevents over-saturation of the generated frames by forcing the last frame to be updated the most to match the first frame.

General input guidance aims to synthesize videos conditioned on general user-specified conditions beyond RGB images, for example, depth maps or sketches. We use a differentiable encoder Ψ, such as a depth estimator (Yang et al., 2024) or an edge predictor (Chan et al., 2022), to extract structural features from the estimated clean image. We minimize an encoder-aligned L2 loss defined as $\mathcal{L}_e = \sum_{i\in\mathcal{I}} \|\Psi(x^i_*) - \Psi(x^i_{0|t})\|_2^2$, where $\Psi(x^i_*)$ denotes the encoded target conditions.

5 Experiments

5.1 Keyframe-guided video generation

We evaluate Frame Guidance on *keyframe-guided* video generation tasks, which aim to synthesize videos that smoothly follow multiple user-specified keyframes. Unlike frame interpolation tasks (Feng et al., 2024; Wang et al., 2025b) that require exact frame matching, keyframe-guided generation only requires the visual similarity to the keyframes, and addresses the generation of longer videos.

Datasets We select 40 clips with more than 81 frames from DAVIS (Pont-Tuset et al., 2017) and 30 real-world videos from Pexels[1] dataset. Pexels features more dynamic and human-centric videos, making it more difficult for video generation. We provide more details on the dataset in Appendix B.2.

Baselines We compare Frame Guidance against frame interpolation methods, including TRF (Feng et al., 2024), SVD-Interp (Wang et al., 2025b), and CogX-Interp. TRF is a training-free approach for Stable Video Diffusion (SVD) (Blattmann et al., 2023), SVD-Interp uses a fine-tuned reversed-motion SVD, and CogX-Interp[2] fine-tunes CogX with first and last frame conditioning. We also compare with basic I2V baselines (CogX (Yang et al., 2025) and Wan (Wang et al., 2025a)). For our method, we apply Frame Guidance on CogX and Wan models using the L2 loss defined in Section 4.4 with the final frame given, and restrict the number of guidance steps so that the total runtime does not exceed 4× the base model's inference time (details in Appendix B.1). We further report results that additionally use the middle frame. We also report results of applying Frame Guidance to CogX-Interp.

[1]https://huggingface.co/datasets/jovianzm/Pexels-400k (Accessed: 2025-09-19)
[2]https://github.com/feizc/CogvideX-Interpolation (Accessed: 2025-09-19)

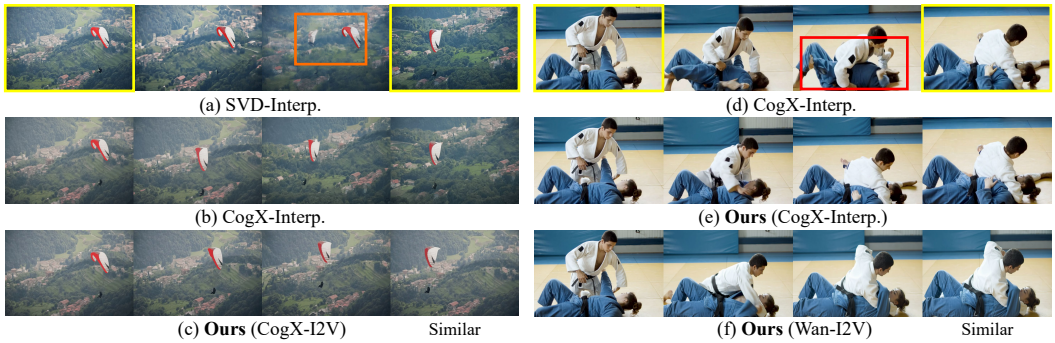
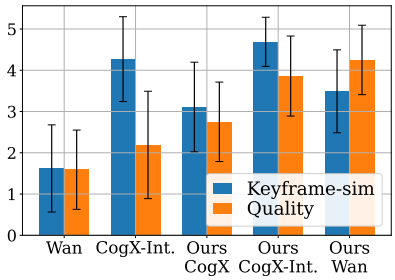


Figure 5: **Qualitative comparison on keyframe-guided video generation tasks**. Yellow box indicates the keyframe condition. Orange box in (a) shows a disconnection in SVD-Interp. Red box in (d) visualizes a failure case for the CogX-Interp baseline for dynamic human motion.



	Input frames	Train free	DAVIS FID ↓	DAVIS FVD ↓	Pexels FID ↓	Pexels FVD ↓
CogX-I2V	I	✓	60.36	890.1	74.98	1122.6
Wan-14B-I2V	I	✓	59.04	772.8	73.03	1033.3
TRF	I, F	✓	62.07	923.1	79.03	1106.2
Ours (CogX)	I, F	✓	57.62	613.4	68.54	1027.3
Ours (CogX)	I, M, F	✓	55.60	577.1	68.97	989.3
Ours (Wan-14B)	I, M, F	✓	57.68	761.1	71.63	904.8
SVD-Interp.	I, F	✗	63.89	800.3	75.37	1210.7
CogX-Interp.	I, F	✗	46.59	506.0	58.73	1081.5
Ours (CogX-Interp.)	I, M, F	✗	**37.95**	**420.3**	**47.86**	**723.26**

Figure 6: **Keyframe-guided generation results. (Left)** Human evaluation. **(Right)** Quantitative results. I, M, and F denote initial, middle, and final frames, respectively. "Train-free" indicates whether the backbone VDM is a base I2V model or fine-tuned for the frame interpolation task.

Qualitative comparison As shown in Figure 5, our approach generates videos with natural transitions, where the selected frames closely resemble the keyframes. For example, Figure 5(c) visualizes well-aligned frames, with the paraglider appearing in a consistent position. In contrast, CogX-Interp often struggles with challenging motion. Applying Frame Guidance to CogX-Interp (Figure 5(e)) or to a stronger VDM backbone (Figure 5(f)) results in notably improved output quality.

Human evaluation We conduct human evaluations to assess the quality of generated videos, focusing on (1) video quality and (2) similarity to the keyframes. As shown in Figure 6 left, applying Frame Guidance to Wan yields the highest video quality, surpassing the trained model CogX-Interp. Applying guidance to CogX-Interp produces high-quality videos with guided frames nearly identical to the keyframes. Further details are provided in Appendix B.2.

Quantitative results We measure FID (Heusel et al., 2017) and FVD (Ge et al., 2024) to assess the quality of the generated videos. As shown in Figure 6 right, Frame Guidance applied to pre-trained I2V models significantly outperforms all other training-free methods. Moreover, Frame Guidance applied to CogX-Interp outperforms all the training-required baselines. These results, combined with the human evaluation, demonstrate that our method effectively guides video generation without additional training. We discuss further details regarding the quantitative results in Appendix B.2.

5.2 Stylized Video Generation

We also validate Frame Guidance on *stylized* video generation tasks, which aim to synthesize videos in the style of a given reference image, using a T2V model.

Dataset We use a subset of the stylized video dataset introduced in StyleCrafter (Liu et al., 2023), which consists of 6 challenging style reference images, each paired with an aligned style prompt and 9 distinct content prompts. We provide further details about the dataset in Appendix B.3.

Baselines We compare our method with three baselines. CogX-T2V is a pre-trained T2V model. VideoComposer (Wang et al., 2023a) is a training-based method supporting multiple conditions, such as style image and depth maps. StyleCrafter (Liu et al., 2023) is also a training-based method that solely trains a style adapter on top of VideoCrafter (Chen et al., 2023). For our method, we apply Frame Guidance to CogX-T2V (Yang et al., 2025) model using the style loss defined in Section 4.4. We provide more details of our method in Appendix B.3.

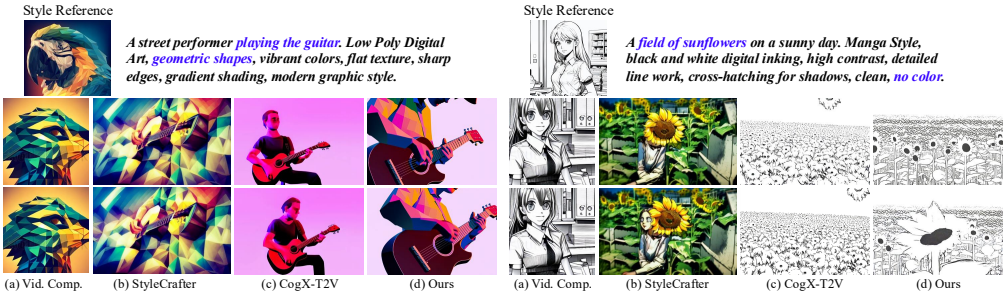
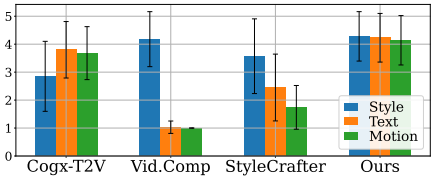


Figure 7: **Qualitative comparison on stylized video generation**. Ours generates high-quality videos that follow the reference style, whereas baselines fail to produce motion or show poor alignment.



	Train free	Text-Alignment		Style-Alignment	
		CLIP-T ↑	ViCLIP-T ↑	CLIP-S ↑	ViCLIP-S ↑
Vid.Comp.	✗	0.211	0.137	0.869	0.219
StyleCrafter	✗	0.207	0.273	**0.635**	0.157
CogX-T2V	✓	0.220	0.259	0.588	0.139
Ours	✓	**0.224**	**0.285**	0.624	**0.185**

Figure 8: **Stylized video generation results. (Left)** Human evaluation. **(Right)** Quantitative results.

Qualitative comparison Figure 7 show that our method can generate balanced stylized videos in terms of both text alignment and style conformity, with diverse motion. In contrast, VideoComposer fails to disentangle content and style in the reference images, while StyleCrafter produces videos with minimal motion that are poorly aligned to the reference style. CogX-T2V struggles to capture detailed textures or patterns, for example, geometric shapes or sunflowers.

Human evaluation We conduct human evaluation to assess the quality of stylized videos, evaluating three criteria (1) style alignment, (2) text alignment, and (3) motion dynamics. As shown in Figure 8 left, our method achieves the best results across all criteria, significantly outperforming the training-based baselines. These results show that Frame Guidance successfully guides video generation to follow the reference style without any additional training. Further details and the results on overall preference are provided in Appendix B.3.

Quantitative results We evaluate the generated videos for text alignment and style alignment using *CLIP-T*, *ViCLIP-T*, *CLIP-S*, and *ViCLIP-S* (Radford et al., 2021; Wang et al., 2023b). As shown in Figure 7 and Figure 8, our method achieves the best scores on all metrics, except for CLIP-S, where it matches the performance of StyleCrafter. While VideoComposer achieves the highest style alignment scores, this is largely due to replicating the style image without adhering to the text prompt.

5.3 Looped Video Generation

We further apply Frame Guidance on the *looped* video generation task, which aims to synthesize videos where the first and last frames match, producing a seamless loop. We use the loop loss defined in Section 4.4 to steer the last frame to match the first. Guidance is applied to the generated video *without requiring any external conditions*, using only text prompts as input. As shown in Figure 1(c) and Figure 17, Frame Guidance generates high-quality looped videos featuring dynamic motions that are well-aligned with the input text prompt.

5.4 Other Applications

Using color block drawing During keyframe-guided generation, keyframe similarity can be flexibly controlled by adjusting the guidance strength. This allows new forms of user-provided control signals that are easy to create, such as coarse sketches or color blocks. In particular, we introduce a novel application that allows users to guide video generation using edited frames, where simple visual edits via color blocks indicate changes in color or detail. As illustrated in Figure 1(d), the generated video depicts the mountain changing color and texture in three distinct ways, which is difficult to achieve using text prompts alone. For Frame Guidance, color blocks act as rough visual hints that allow natural scene transitions while preserving the contents. We provide more examples in Figure 18.

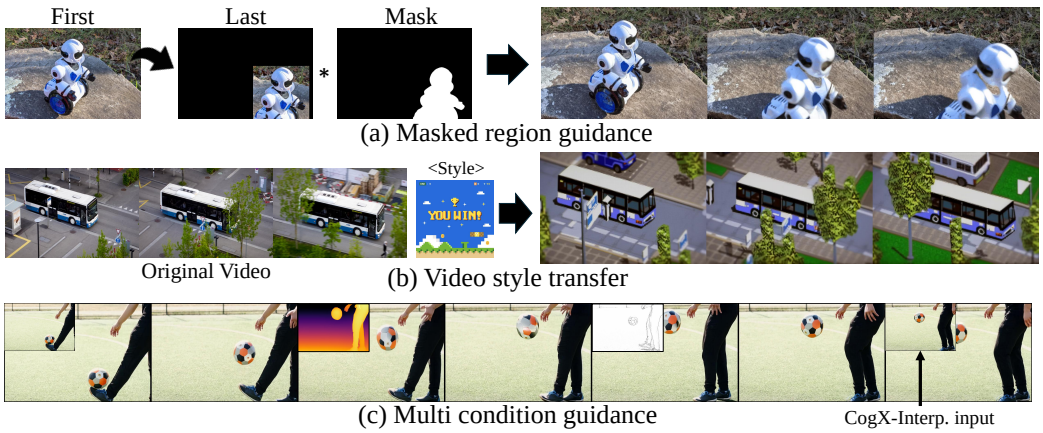


Figure 9: Examples of other applications. (a) Object movement guided by masked region. (b) Video style transfer with SDEdit (Meng et al., 2022). (c) Guidance using multiple types of inputs: depth map and sketch.

Masked region guidance While our previously described methods apply guidance to the whole area of a frame, we demonstrate that the guidance can be effectively restricted to specific regions by using L2 loss with a binary mask. In Figure 9(a), we present an example of generating a video with object motion, guided by a cropped image and its segmentation mask. By applying guidance solely to the object region, the background remains unchanged while the object shows smooth movement.

Depth map / Sketch guidance Furthermore, Frame Guidance supports general types of frame-level signals, such as depth maps and sketches, which offer more user-friendly conditioning compared to RGB images as input. Using the general input guidance defined in Section 4.4, Frame Guidance is capable of generating high-quality guided videos as shown in Figure 1(e) and (f).

Video style transfer We extend Frame Guidance to video editing tasks. Taking a video as input, we apply Frame Guidance to generate an edited video that follows a reference style. It can be achieved by applying a simple SDEdit (Meng et al., 2022) with a small noise. This results in preserving the original motion and layout while successfully transferring the reference style, as in Figure 9(b).

Multi condition guidance Frame Guidance can integrate multiple input types by combining losses. As shown in Figure 9(c), we apply guidance to intermediate frames, combining the depth map loss and sketch loss for the CogX-Interp model. The generated video demonstrates smooth motion that follows the input signals, showing the flexibility of Frame Guidance in handling complex scenarios. We provide additional examples on multi condition guidance in Figure 20.

5.5 Ablation Studies

Necessity of VLO To validate the importance of VLO in Frame Guidance, we compare it against two variants: one that uses only the time-travel trick and another that applies only the deterministic update from Equation 2 during the guidance process. Table 1 shows that using only the time-travel trick yields higher FVD scores due to difficulty in forming coherent layouts, while the deterministic update alone produces over-saturated or temporally disconnected videos. We provide an additional ablation study on VLO hyperparameter t_E that determines when to apply deterministic update in Appendix C.5.

Table 1: Ablation study on latent optimization strategy.

Method	FID ↓	FVD ↓
Time-travel	57.37	778.4
Deterministic	56.61	637.3
VLO (Ours)	**55.60**	**577.1**

Model agnostic As shown in Figure 6, our method is compatible with a variety of VDMs, including CogVideoX (Yang et al., 2025), its fine-tuned variant CogVideoX-Interpolation, and Wan-14B (Wang et al., 2025a), a flow-matching-based model. To further demonstrate its generality, we also apply our approach to two additional models: SVD (Blattmann et al., 2023), a U-Net-based (Ronneberger et al., 2015) diffusion model, and LTX-2B (HaCohen et al., 2024), which supports sequences up to 161 frames. As illustrated in Figure 21, our method consistently performs well across all these VDMs.

6 Conclusion

In this work, we present Frame Guidance, a novel training-free framework for diverse control tasks using frame-level signals. By applying guidance to selected frames, our method enables natural control throughout the video. To achieve this, we partially decode sliced latents during guidance computation and introduce a latent optimization strategy designed for video. Our approach supports a wide range of tasks without training, including special cases such as color block guidance and looped video generation. We discuss the limitations of our method in Appendix D.

Reproducibility Statement

To ensure reliable and reproducible results, we have provided the source code on supplementary materials, and detailed experiment settings in Appendix B.

References

Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, et al. Recammaster: Camera-controlled generative rendering from a single video. *arXiv preprint arXiv:2503.11647*, 2025.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Roni Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *International Conference on Learning Representations*, 2024.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

Ryan Burgert, Yuancheng Xu, Wenqi Xian, Oliver Pilarski, Pascal Clausen, Mingming He, Li Ma, Yitong Deng, Lingxiao Li, Mohsen Mousavi, et al. Go-with-the-flow: Motion-controllable video diffusion models using real-time warped noise. *arXiv preprint arXiv:2501.08331*, 2025.

Caroline Chan, Frédo Durand, and Phillip Isola. Learning to generate line drawings that convey geometry and semantics. In *Conference on Computer Vision and Pattern Recognition*, 2022.

Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, et al. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 2021.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

Haiwen Feng, Zheng Ding, Zhihao Xia, Simon Niklaus, Victoria Abrevaya, Michael J Black, and Xuaner Zhang. Explorative inbetweening of time and space. In *European Conference on Computer Vision*, 2024.

Songwei Ge, Aniruddha Mahapatra, Gaurav Parmar, Jun-Yan Zhu, and Jia-Bin Huang. On the content bias in fréchet video distance. In *Conference on Computer Vision and Pattern Recognition*, 2024.

Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. In *International Conference on Learning Representations*, 2024.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for video diffusion models. In *International Conference on Learning Representations*, 2025.

Yutong He, Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Dongjun Kim, Wei-Hsiang Liao, Yuki Mitsufuji, J Zico Kolter, Ruslan Salakhutdinov, and Stefano Ermon. Manifold preserving guided diffusion. In *International Conference on Learning Representations*, 2024.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 2020.

Chen Hou, Guoqiang Wei, Yan Zeng, and Zhibo Chen. Training-free camera control for video generation. *arXiv preprint arXiv:2406.10126*, 2024.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. *arXiv preprint arXiv:2503.07598*, 2025.

Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. In *International Conference on Computer Vision*, 2023.

Jialu Li, Shoubin Yu, Han Lin, Jaemin Cho, Jaehong Yoon, and Mohit Bansal. Training-free guidance in text-to-video generation via multimodal planning and structured noise initialization. *arXiv preprint arXiv:2504.08641*, 2025a.

Quanhao Li, Zhen Xing, Rui Wang, Hui Zhang, Qi Dai, and Zuxuan Wu. Magicmotion: Controllable video generation with dense-to-sparse trajectory guidance. *arXiv preprint arXiv:2503.16421*, 2025b.

Pengyang Ling, Jiazi Bu, Pan Zhang, Xiaoyi Dong, Yuhang Zang, Tong Wu, Huaian Chen, Jiaqi Wang, and Yi Jin. Motionclone: Training-free motion cloning for controllable video generation. In *International Conference on Learning Representations*, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Gongye Liu, Menghan Xia, Yong Zhang, Haoxin Chen, Jinbo Xing, Yibo Wang, Xintao Wang, Yujiu Yang, and Ying Shan. Stylecrafter: Enhancing stylized text-to-video generation with style adapter. *arXiv preprint arXiv:2312.00330*, 2023.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

Nithin Gopalakrishnan Nair and Vishal M Patel. Dreamguider: Improved training free diffusion-based conditional generation. *arXiv preprint arXiv:2406.02549*, 2024.

Koichi Namekata, Sherwin Bahmani, Ziyi Wu, Yash Kant, Igor Gilitschenski, and David B. Lindell. SG-i2v: Self-guided trajectory control in image-to-video generation. In *International Conference on Learning Representations*, 2025.

OpenAI. Gpt-4o system card. *Technical report*, 2024.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, David Yan, Dhruv Choudhary, Dingkang Wang, Geet Sethi, Guan Pang, Haoyu Ma, Ishan Misra, Ji Hou, Jialiang Wang, Kiran Jagadeesh, Kunpeng Li, Luxin Zhang, Mannat Singh, Mary Williamson, Matt Le, Matthew Yu, Mitesh Kumar Singh, Peizhao Zhang, Peter Vajda, Quentin Duval, Rohit Girdhar, Roshan Sumbaly, Sai Saketh Rambhatla, Sam Tsai, Samaneh Azadi, Samyak Datta, Sanyuan Chen, Sean Bell, Sharadh Ramaswamy, Shelly Sheynin, Siddharth Bhattacharya, Simran Motwani, Tao Xu, Tianhe Li, Tingbo Hou, Wei-Ning Hsu, Xi Yin, Xiaoliang Dai, Yaniv Taigman, Yaqiao Luo, Yen-Cheng Liu, Yi-Chiao Wu, Yue Zhao, Yuval Kirstain, Zecheng He, Zijian He, Albert Pumarola, Ali Thabet, Artsiom Sanakoyeu, Arun Mallya, Baishan Guo, Boris Araya, Breena Kerr, Carleigh Wood, Ce Liu, Cen Peng, Dimitry Vengertsev, Edgar Schonfeld, Elliot Blanchard, Felix Juefei-Xu, Fraylie Nord, Jeff Liang, John Hoffman, Jonas Kohler, Kaolin Fire, Karthik Sivakumar, Lawrence Chen, Licheng Yu, Luya Gao, Markos Georgopoulos, Rashel Moritz, Sara K. Sampson, Shikai Li, Simone Parmeggiani, Steve Fine, Tara Fowler, Vladan Petrovic, and Yuming Du. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2025.

Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015.

Litu Rout, Yujia Chen, Nataniel Ruiz, Abhishek Kumar, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. RB-modulation: Training-free stylization using reference-based modulation. In *International Conference on Learning Representations*, 2025.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

Yifei Shen, Xinyang Jiang, Yifan Yang, Yezhen Wang, Dongqi Han, and Dongsheng Li. Understanding and improving training-free loss-based diffusion guidance. In *Advances in Neural Information Processing Systems*, 2024.

Gowthami Somepalli, Anubhav Gupta, Kamal Gupta, Shramay Palta, Micah Goldblum, Jonas Geiping, Abhinav Shrivastava, and Tom Goldstein. Measuring style similarity in diffusion models. *arXiv preprint arXiv:2404.01292*, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022.

Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025a.

Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion controllability. *Advances in Neural Information Processing Systems*, 2023a.

Xiaojuan Wang, Boyang Zhou, Brian Curless, Ira Kemelmacher-Shlizerman, Aleksander Holynski, and Steve Seitz. Generative inbetweening: Adapting image-to-video models for keyframe interpolation. In *International Conference on Learning Representations*, 2025b.

Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*, 2023b.

Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH*, 2024.

Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *International Conference on Computer Vision*, 2023.

Tianxing Wu, Chenyang Si, Yuming Jiang, Ziqi Huang, and Ziwei Liu. Freeinit: Bridging initialization gap in video diffusion models. In *European Conference on Computer Vision*, 2024a.

Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, David Junhao Zhang, Mike Zheng Shou, Yan Li, Tingting Gao, and Di Zhang. Draganything: Motion control for anything using entity representation. In *European Conference on Computer Vision*, 2024b.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 2024.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan.Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, 2025.

Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. In *International Conference on Computer Vision*, 2023.

Lijun Yu, Jose Lezama, Nitesh Bharadwaj Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A Ross, and Lu Jiang. Language model beats diffusion - tokenizer is key to visual generation. In *International Conference on Learning Representations*, 2024.

Yan Zeng, Guoqiang Wei, Jiani Zheng, Jiaxin Zou, Yang Wei, Yuchen Zhang, and Hang Li. Make pixels dance: High-dynamic video generation. In *Conference on Computer Vision and Pattern Recognition*, 2024.

Yabo Zhang, Yuxiang Wei, Dongsheng Jiang, XIAOPENG ZHANG, Wangmeng Zuo, and Qi Tian. Controlvideo: Training-free controllable text-to-video generation. In *International Conference on Learning Representations*, 2024.

Guangcong Zheng, Teng Li, Rui Jiang, Yehao Lu, Tao Wu, and Xi Li. Cami2v: Camera-controlled image-to-video diffusion model. *arXiv preprint arXiv:2410.15957*, 2024.

Appendix

Organization The Appendix is organized as follows: In Section A, we provide the additional backgrounds of our work. We describe the details of the experiments and our framework in Section B, and further discussion in Section C. Lastly, in Section D, we discuss the limitations of our work.

A Backgrounds

A.1 Training-free diffusion guidance

Recent works Song et al. (2021); Dhariwal and Nichol (2021); Yu et al. (2023); Chung et al. (2023); Bansal et al. (2024); He et al. (2024); Shen et al. (2024) have explored conditional generation by injecting external conditions into pre-trained diffusion models. Among them, training-free guidance methods (Yu et al., 2023; Chung et al., 2023; Bansal et al., 2024; He et al., 2024; Shen et al., 2024) achieve controllable generation without additional training by optimizing the noisy latent during the reverse process. This optimization is guided by a loss function that measures the alignment between intermediate latents and the target condition at each denoising step. FreeDom (Yu et al., 2023) and UniversalGuidance (Bansal et al., 2024) leverage off-the-shelf models to compute the various guidance losses, achieving a wide range of controllable image generation tasks. Later works (He et al., 2024; Nair and Patel, 2024; Rout et al., 2025) bypass the denoising module for computing the guidance loss, enabling more efficient training-free diffusion guidance.

A.2 Flow matching

Flow matching (Lipman et al., 2022) belongs to the family of flow-based generative models, which are known for faster sampling compared to diffusion models (Ho et al., 2020). Let $t \in [0, 1]$ be the time, $x \in \mathbb{R}^d$ be a data, and q be a unknown target distribution. The goal of flow matching Lipman et al. (2022) is to estimate a time-dependent transformation $z_t : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$ (referred to as *flow*) that maps a prior distribution p_0 (e.g., Gaussian) to a distribution $p_1 \approx q$. Instead of directly estimating the flow, Lipman et al. (2022) proposes to regress a *generating vector field* $v_t(\cdot, t) : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$ that induces the flow z_t via the following ordinary differential equation (ODE):

$$\frac{dz_t(x)}{dt} = v_t(z_t(x)) \quad \text{and} \quad z_0(x) = x. \tag{3}$$

It is common practice to design this flow ϕ_t along an optimal transport (OT) trajectory that connects a prior sample to a target sample with a straight interpolation: $z_t := (1 - t)x_0 + tx_1$, where $x_0 \sim p_0$ and $x_1 \sim q$. In this case, the target v_t is computed as a constant: $v_t(x, t) = x_1 - x_0$ for all $t \in [0, 1]$. With a neural network v_θ that estimates v_t, we can generate a data x_1 by numerically solving the ODE in Equation 3 (e.g., Euler method). Similar to Tweedie's formula Efron (2011), we can approximate a cleaned sample at each time t by

$$z_{1|t} := z_t + \frac{1}{1 - t} v_\theta(z_t, t). \tag{4}$$

Throughout this paper, we interchangeably reverse the direction of time by parameterizing it as $s(t) = T(1 - t),\ t \in [0, 1]$ to align with the convention of the diffusion models where the generative process proceeds from T to 0.

B Experimental details

B.1 Implementation details

All our experiments are conducted on a single H100 GPU. Hyperparameters related to guidance, such as step size η and repetition M, are adjustable depending on the task and model characteristics. For example, in keyframe-guided video generation using diffusion-based CogVideoX (Yang et al., 2025), we define the layout stage within the first 5 steps, set $M = 10$, and use a step size of $\eta = 3.0$. For the time-travel trick, M is linearly decreased over 15 steps. At each step, gradients are L2-normalized before being scaled by η for the update. All comparisons in our paper were conducted using the same random seed.

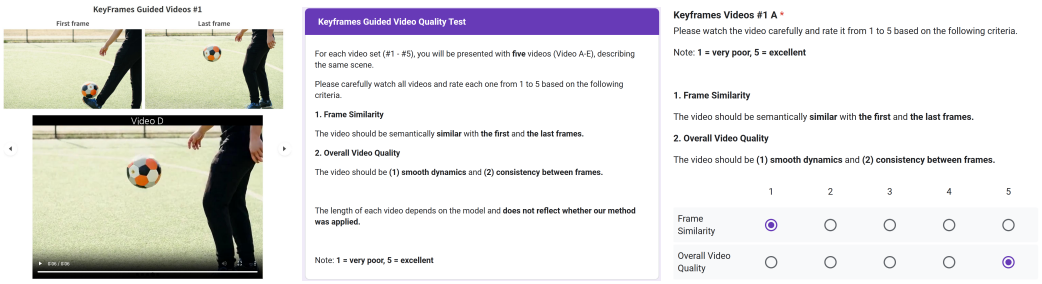


Figure 10: A screenshot of questionnaires from our human evaluation on keyframe-guided generation.

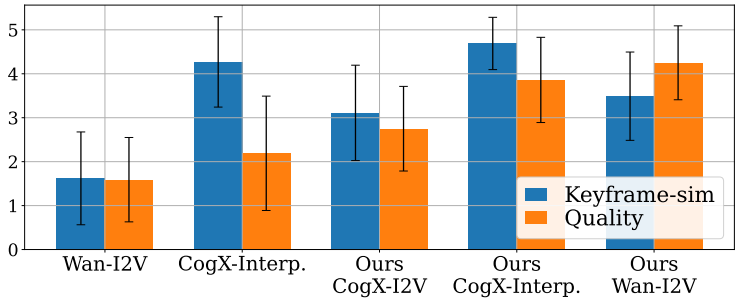


Figure 11: Human evaluation results on keyframe-guided generation including Wan-I2V.

Since Wan-14B (Wang et al., 2025a) employs flow matching as its generative modeling, its inference is fully deterministic, and the layout is mostly established within 2 steps. Therefore, we set the layout stage to the first 2 inference steps, and apply the same M, η, and time-travel configuration. Moreover, since our implementation introduces more stochasticity (see Appendix C.3), we slightly reduce the number of time-travel steps. To maintain practicality, we empirically limit the number of guidance steps such that the overall runtime does not exceed $4\times$ the base model's inference time.

To reduce GPU memory usage, we apply gradient checkpointing (Chen et al., 2016) to the denoising network using the Diffusers (von Platen et al., 2022) library. For the CausalVAE, gradient checkpointing is applied only in CogVideoX (Yang et al., 2025), as Wan-14B (Wang et al., 2025a) implementation does not currently support it. We do not apply spatial downsampling in CogVideoX, since it runs on a single GPU without it. In contrast, we apply $2\times$ spatial downsampling in experiments with Wan-14B.

B.2 Keyframe-guided video generation

Dataset For evaluation, we use videos from the DAVIS (Pont-Tuset et al., 2017) dataset and Pexels. From DAVIS, we select 40 videos with at least 81 frames, matching the maximum frame length supported by Wan-14B (Wang et al., 2025a). The resolution of each video is resized and center-cropped according to the requirements of each pre-trained model. To ensure fair comparisons across models, the same initial and final frames are used. Based on this setup, the reference set for each model is configured with slightly different FPS settings. For example, for an 81-frame video, CogVideoX (Yang et al., 2025) supports only 49 frames, so we temporally downsample the video accordingly. The Pexels dataset contains more real-world videos with challenging motions and frequent camera view changes. We randomly select a subset of 30 videos, which features more dynamic and human-centric content compared to DAVIS.

For pre-trained models that accept text prompts as input, except for Stable Video Diffusion (Blattmann et al., 2023)(SVD)-based methods (Feng et al., 2024; Wang et al., 2025b), we used prompts derived from the original videos. Specifically, we concatenated three frames from each original video and generated a caption using GPT-4o (OpenAI, 2024). The same prompt was applied consistently across all baseline models.

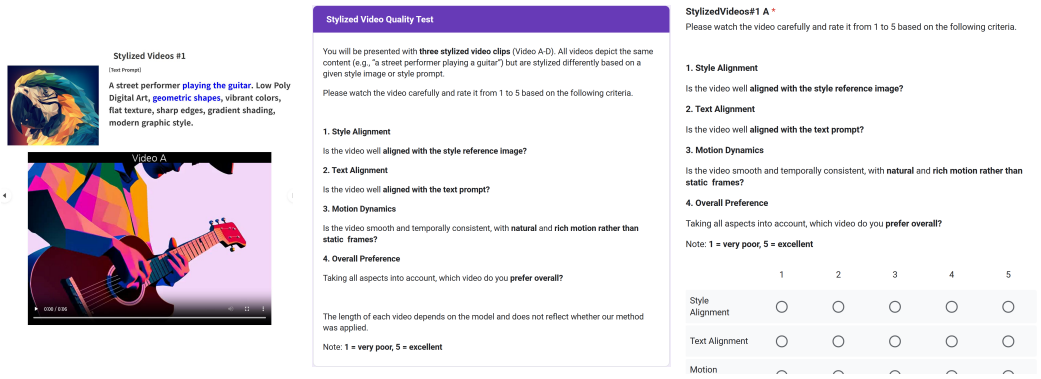


Figure 12: A screenshot of questionnaires from our human evaluation on stylized video generation.

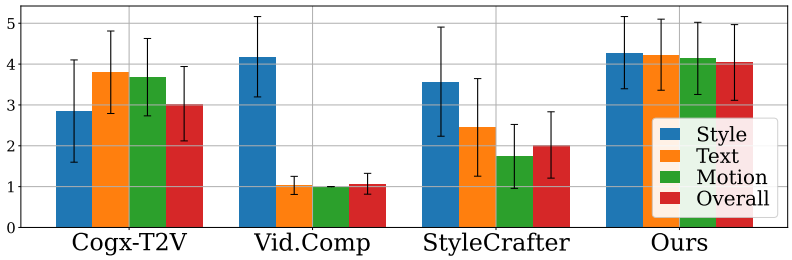


Figure 13: Human evaluation results on stylized video generation including overall preference.

Human evaluation We conduct human evaluation for keyframe-guided video generation task to evaluate two main aspects: (1) video quality and (2) similarity to the keyframes. Both metrics are rated on an absolute scale from 1 to 5. As shown in Figure 10, participants evaluated all videos generated from the same keyframes side by side. We collected responses from 20 participants, evaluating 5 types of videos across 5 different methods. The full human evaluation results, including Wan-I2V, are provided in Figure 11.

Evaluation metric For evaluation metric, we employ FID (Heusel et al., 2017) and content-debiased FVD (Ge et al., 2024) between generated videos and real videos. Both metrics quantify the distributional distance between generated videos and real videos from the dataset. FID is computed by extracting all frames from the video and treating them as individual images. FVD is measured against reference videos adjusted to match each model's resolution and FPS. Therefore, cross-model comparisons are not strictly valid.

As shown in Figure 6 right, our method with Wan slightly outperforms Wan I2V in these quantitative metrics. However, human evaluations in Figure 6 left suggest a more noticeable improvement, which may not be fully captured by such metrics. Notably, the overall FID and FVD scores are relatively high, as our setting involves longer and more dynamic videos compared to related tasks such as video interpolation, making the dataset more challenging.

We provide more qualitative examples in Figure 14.

B.3 Stylized Video Generation

Based on our analysis of layout formation in Section 4.2, we apply VLO with a different schedule for stylized video generation compared to keyframe-guided video generation. Specifically, we start applying the deterministic latent update (Equation 2) at step 3 before entering the detail stage (step 5), and then switch to time travel during steps 15 - 20. This design helps shape the geometric patterns and structure of the style reference image during the layout stage. After that, we proceed the inference without guidance. We set the guidance step size $\eta = 3$ and the number of repetition $M = 5$. We compute the style guidance loss on 4 evenly spaced frames from the entire video.

Table 2: Text prompts Liu et al. (2023) used for stylized video generation.

Content prompt	Content prompt
A street performer playing the guitar.	A wolf walking stealthily through the forest.
A chef preparing meals in kitchen.	A hot air balloon floating in the sky.
A student walking to school with backpack.	A wooden sailboat docked in a harbor.
A bear catching fish in a river.	A field of sunflowers on a sunny day.
A knight riding a horse through a field.	

Table 3: Style references and style prompts (Liu et al., 2023) used for stylized video generation.

Style image	Style prompt	Style image	Style prompt
	Manga Style, black and white digital inking, high contrast, detailed line work, cross-hatching for shadows, clean, no color.		Ink and watercolor on paper, urban sketching style, detailed line work, washed colors, realistic shading, and a vintage feel.
	Low Poly Digital Art, geometric shapes, vibrant colors, flat texture, sharp edges, gradient shading, modern graphic style.		Manga-inspired digital art, dynamic composition, exaggerated proportions, sharp lines, cel-shading, high-contrast colors with a focus on sepia tones and blues.
	Wartercolor Paining, fluid brushstrokes, transparent washes, color blending, visible paper texture, impressionistic style.		Pixel art illustration, digital medium, detailed sprite work, vibrant color palette, smooth shading, and a nostalgic, retro video game aesthetic.

Dataset We use a subset of the test dataset introduced in StyleCrafter (Liu et al., 2023), which consists of 9 content prompts and 6 style reference images with corresponding style descriptions. In Table 2 and Table 3, we detail our test dataset. The content prompts describe an entire video content using a simple sentence, while the style prompts describe the styles of the video. The style prompts are generated by GPT-4o (OpenAI, 2024). We concatenate each content prompt with each style prompt, resulting in a total of 54 full prompts for stylized video generation.

Human evaluation In Figure 12, we provide screenshots of the questionnaires and labeling instructions. 20 participants are asked to evaluate four metrics: (1) style alignment, (2) text alignment, (3) motion dynamics, and (4) overall video preference of five stylized videos generated by four models. All metrics were rated on an absolute scale from 1 to 5. The complete evaluation results, including overall preference, are provided in Figure 13.

Evaluation metric We employ CLIP-Text and ViCLIP-Text to access the text alignment of the generated videos. We also compute CLIP-Style and ViCLIP-Style to access the style conformity of the generated videos. Specifically,CLIP-Text and CLIP-Style are computed by using the CLIP Radford et al. (2021) text and image encoders, respectively:

$$\frac{1}{L}\sum_{l=1}^{L}\frac{f_I(x_l)\cdot f_T(p)}{\|f_I(x_l)\|_2\|f_T(p)\|_2} \quad \text{and} \quad \frac{1}{L}\sum_{l=1}^{L}\frac{f_I(x_l)\cdot f_I(x_{\text{style}})}{\|f_I(x_l)\|_2\|f_I(x_{\text{style}})\|_2}, \tag{5}$$

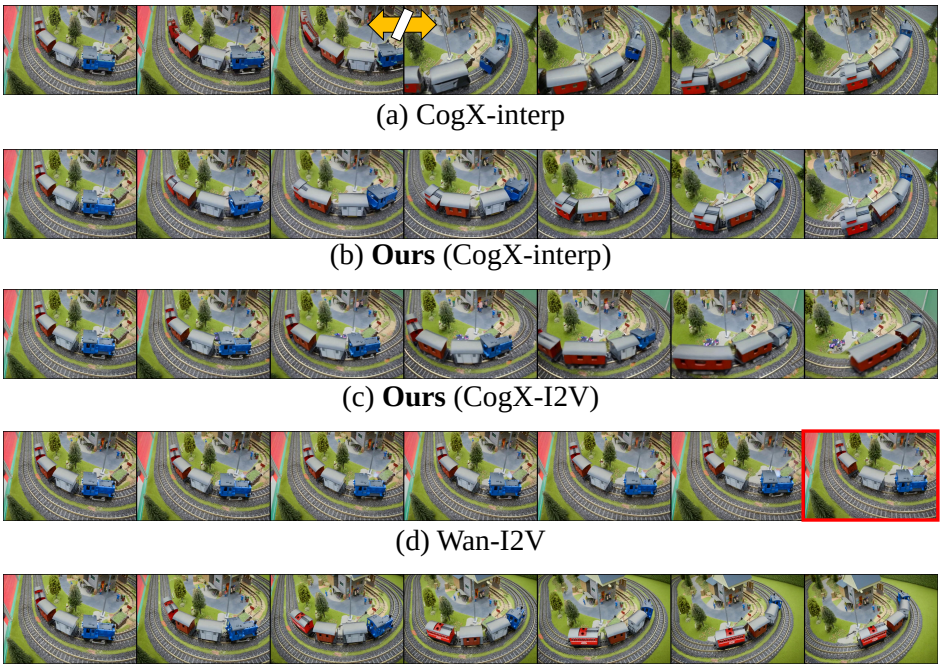
where x_l is the l-th frame, p is the text prompt, x_{style} is the style reference image, and $f_I(\cdot)$ and $f_T(\cdot)$ are the CLIP (Radford et al., 2021) image and text encoders, respectively.

Similarly, ViCLIP-Text and ViCLIP-Style are both computed by using Video CLIP model (Wang et al., 2023b):

$$\frac{f_V(x)\cdot f_T(p)}{\|f_V(x)\|_2\|f_T(p)\|_2} \quad \text{and} \quad \frac{f_V(x)\cdot f_T(p_{\text{style}})}{\|f_V(x)\|_2\|f_T(p_{\text{style}})\|_2}, \tag{6}$$

where x is the video, p and p_{style} are the full and style prompts, and $f_V(\cdot)$ and $f_T(\cdot)$ are the ViCLIP video and text encoders, respectively.

We provide more qualitative examples in Figure 15 and Figure 16.



(a) CogX-interp

(b) **Ours** (CogX-interp)

(c) **Ours** (CogX-I2V)

(d) Wan-I2V

(e) **Ours** (Wan)

Figure 14: **Qualitative comparison** of keyframe-guided video generation. Orange arrows indicate temporally disconnected frames, and red boxes highlight poor keyframe similarity. Our method generates temporally coherent videos while maintaining semantic similarity to the keyframes.

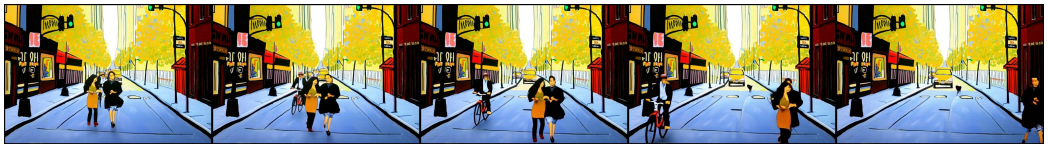
B.4 Loop video generation

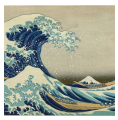
We use the similar guidance schedule with keyframe-guided video generation task, but reduce the early guidance strength to avoid producing over-saturated examples. We provide more qualitative examples in Figure 17.

B.5 Additional generated examples

We provide more examples on Frame Guidance with color block image in Figure 18, multi condition (style and loop loss) in Figure 20. We show examples generated by other models, SVD (Blattmann et al., 2023) and LTX-2B (HaCohen et al., 2024), are shown in Figure 21.

"A New York City street scene with a man and a woman walking down the street, a dog running after them, and a bicyclist passing by **in oil painting style**."





"A New York City street scene with a man and a woman walking down the street, a dog running after them, and a bicyclist passing by **in Ukiyo-e style**."

"A bustling Paris café in the morning, waiters serving coffee, people chatting at tables, and a dog lying under a chair in **Impasto oil painting style with vibrant colors.**"

Figure 15: Stylized video generated by Frame Guidance using style loss. These videos are generated by CogVideoX-T2V.

Ø
(Base T2V)

"A New York City street scene with a man and a woman walking down the street, a dog running after them, and a bicyclist passing by, **in watercolor painting style**."

(a) Base T2V

"A New York City street scene with a man and a woman walking down the street, a dog running after them, and a bicyclist passing by, **in watercolor painting style**."

(b) Frame Guidance using style loss

Figure 16: Stylized video generated by Frame Guidance using style loss with the same random seed. While their content remains similar, the style is primarily altered. These videos are generated by CogVideoX-T2V.

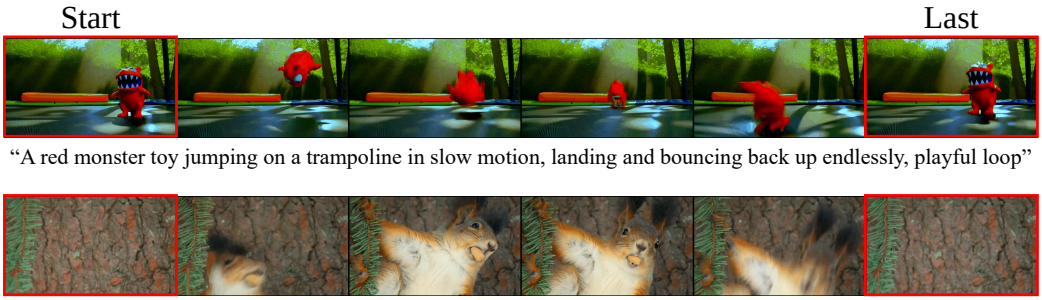
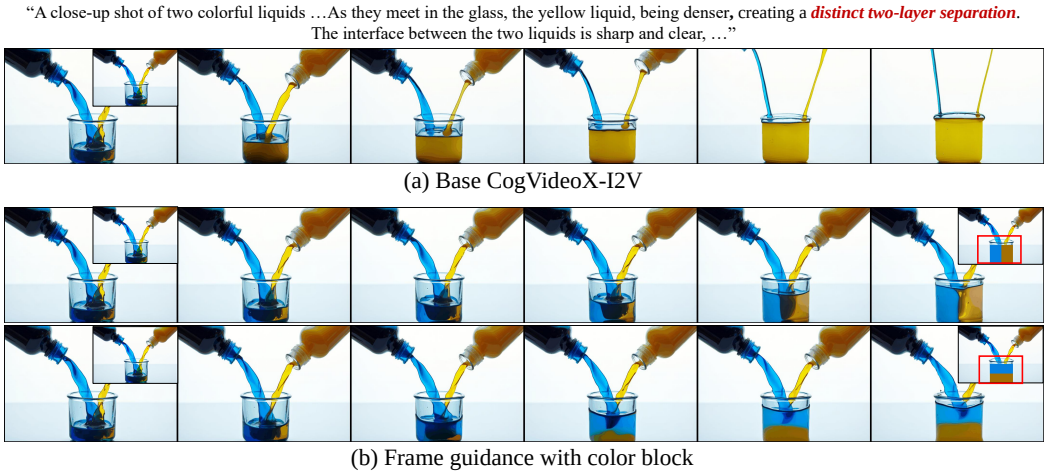


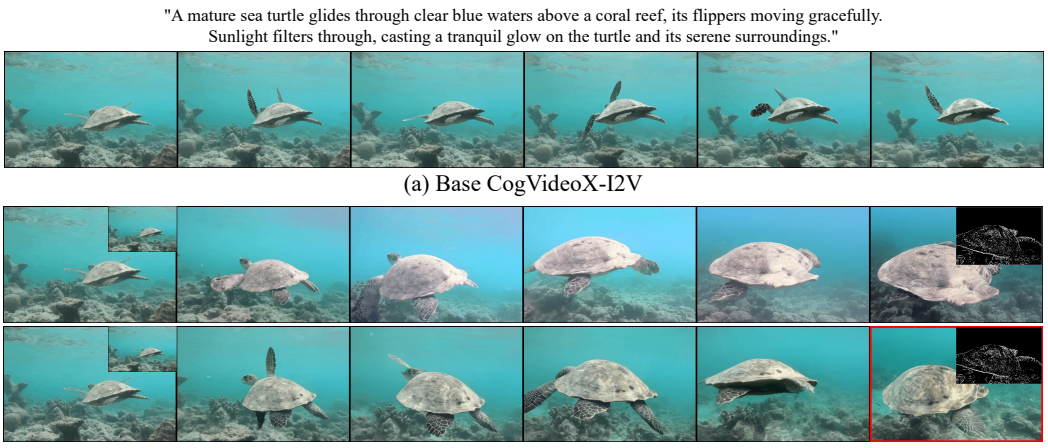
Figure 17: Loop video generated by Frame Guidance using loop loss. These videos are generated by Wan-14B T2V.



(a) Base CogVideoX-I2V

(b) Frame guidance with color block

Figure 18: Frame Guidance with a color block image allows the generation of a video with a complex scene. These videos are generated by CogVideoX-I2V.



(a) Base CogVideoX-I2V

(b) Frame guidance with edge map (Sobel filter)

Figure 19: Frame Guidance with edge map. Canny edges are intractable, so we replaced it with Sobel filter. While this approach works to some extent, it struggles to capture fine details and fails under large scene changes, which we discuss in our limitation Section D. These videos are generated by CogVideoX-2V.

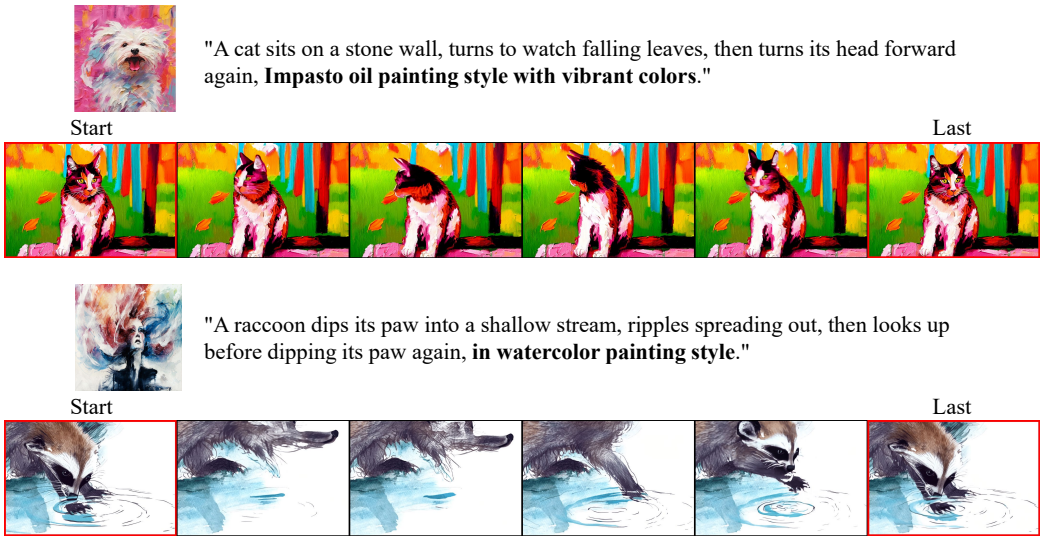


Figure 20: Frame Guidance with style and loop loss. Simply summing the two losses enables effective composition of both guidance signals. These videos are generated by CogVideoX-T2V.

(a) LTX-2B

(a) Stable Video Diffusion

Figure 21: Frame Guidance is model-agnostic. It is compatible with both SVD (Blattmann et al., 2023) and LTX-2B (HaCohen et al., 2024). For SVD, since it does not use a temporally compressed VAE, we skip latent slicing. Some saturation observed in the LTX-2B results occasionally occurs due to the model itself.

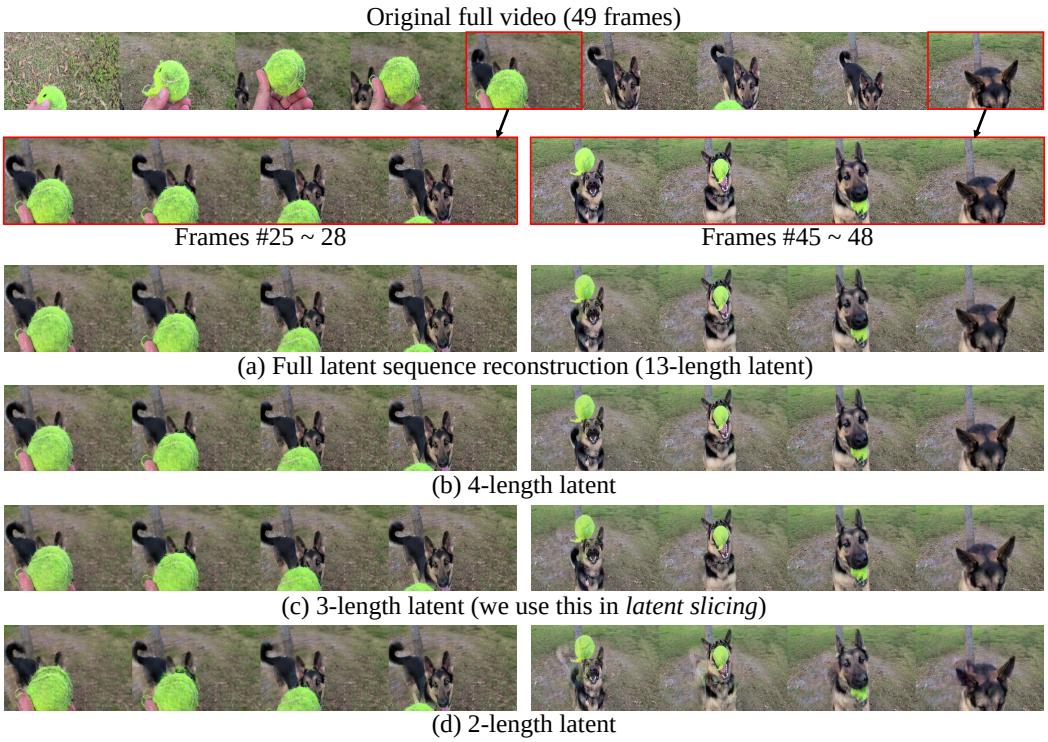


Figure 22: **Video reconstruction with temporally sliced latent.** **(a)** Decoding the full latent sequence successfully reconstructs the original video. **(b)–(c)** Using 4 or 3-length latent around the target latent (frame) is sufficient for accurate reconstruction. **(d)** With only 2-length latent, there is slight degradation, therefore, we adpot 3-length latent for the main experiments.

C More discussions

C.1 Video reconstruction with sliced latent

As shown in Figure 22, we can reconstruct nearly identical frames even with temporally sliced latents. When the 49-frame real video at the top is encoded by CogVideoX's CausalVAE, each frame is mapped to a latent $z_t \in \mathbb{R}^{c \times f \times h \times w}$ with a temporal latent length of $f = 13$. The four reconstructed frames on the right side of panels (a–d) correspond to the last four frames of the original video.

- In (a), we fully decode the entire 13-length latent z_t to obtain the 49-frame reconstructed video and visualize the last four frames.
- In (b), we decode only the last four temporal slices (i.e., z_t[:, -4:]), which we refer to as 4-length latent. From this partial latent, the model produces 13-frame reconstructed video and we visualize the last four frames.

These qualitative results indicates that even for fast-motion videos, a 3-length latent around the target frame is sufficient for accurate reconstruction (Figure 22(c)), while a 2-length latent shows minor degradation but remains close to the full-latent result (Figure 22(d)).

C.2 Time-travel trick in layout stage

As discussed in Section 4.2, directly applying the time-travel trick (Shen et al., 2024; Yu et al., 2023; Bansal et al., 2024) to video diffusion models struggles due to excessive stochasticity. The time-travel trick in Algorithm 2 includes a single-step forward process, but in practice, the added noise is extremely large, and the coefficient multiplied with the latent is very small, as shown in Table 4. In fact, in the very first inference step, the coefficient $\sqrt{\beta_t}$ becomes 0, resulting in no guidance effect at all.

Therefore, since the effect of guidance is absent during the early stages when the layout is largely established, the model fails to produce a layout that aligns with the given condition. Even when guidance is applied later, as discussed in Figure 4(d), only the guided frame is updated, and it cannot correct the overall layout. Our proposed VLO addresses this issue by applying a deterministic latent optimization in the early stage.

Table 4: Forward process coefficients in early inference steps.

Step (·/50)	$\sqrt{\beta_t}$	$\sqrt{1-\beta_t}$
1	0.00	1.00
2	0.48	0.88
3	0.64	0.77

Additionally, we provide a visual comparison in Figure 23. The time-travel trick offers almost no guidance effect in the early steps, which are crucial for layout formation. Since it fails to establish a proper layout early on, later steps cannot correct this deficiency. As shown in Figure 23(a), it generates a static camera view, similar to ordinary I2V generation. In contrast, our VLO provides sufficiently effective guidance through deterministic updates in the early steps, enabling the model to establish a proper left-to-right moving layout, which in turn allows later guidance to take meaningful effect, as illustrated in Figure 23(b).

Algorithm 2 Time Travel (diffusion model)

Require: $z_t, z_{0|t}, t, g_t$
$\epsilon \leftarrow \mathcal{N}(0, I)$
$z_{t-1} \leftarrow \mathrm{DDIM}(z_t, z_{0|t})$
$z_{t-1} \leftarrow z_{t-1} - \eta \cdot g_t$
$\beta_t \leftarrow \alpha_t / \alpha_{t-1}$
$z_t \leftarrow \sqrt{\beta_t} z_t + \sqrt{1-\beta_t}\epsilon$ {Renoising}
return z_t

Algorithm 3 Time Travel-F (flow matching)

Require: $z_t, z_{0|t}, t, g_t$
$\epsilon \leftarrow \mathcal{N}(0, I)$
$z_t \leftarrow \sigma_t \epsilon + (1-\sigma_t) z_{0|t}$ {Renoising}
$z_t \leftarrow z_t - \eta \cdot g_t$
return z_t

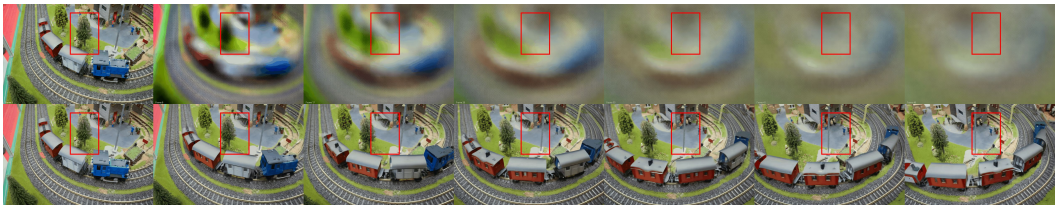
Algorithm 4 Frame Guidance (Diffusion, full)

Require: $\mathcal{I}$, t_E, t_L, repeat step M, step size η, guidance loss $\mathcal{L}_e$, model $v_\theta(\cdot,\cdot)$
$z_T \sim \mathcal{N}(0, I)$
$\mathcal{J} \leftarrow$ Frame-Idx-to-Latent-Idx($\mathcal{I}$)
for $t = T, ..., 1$ **do**
  if $t > t_L$ **then** {Guidance step}
    for $m = 1, ..., M-1$ **do**
      $z_{0|t} \leftarrow \sqrt{\bar{\alpha}_t} z_t - \sqrt{1-\bar{\alpha}_t} \cdot v_\theta(z_t, t)$
      $z^{\mathcal{J}}_{0|t} \leftarrow$ Latent-Slicing$(z_{0|t}, \mathcal{J})$
      $x^{\mathcal{I}}_{0|t} \leftarrow \mathcal{D}(z^{\mathcal{J}}_{0|t})$
      $g_t = \nabla_{z_t} \mathcal{L}_e(x^{\mathcal{I}}_{0|t}, c_{\texttt{frames}})$
      if $t > t_E$ **then** {Early steps}
        $z_t \leftarrow z_t - \eta g_t$
      else {Later steps}
        $\epsilon \leftarrow \mathcal{N}(0, I)$
        $z_{t-1} \leftarrow \mathrm{DDIM}(z_t, z_{0|t})$
        $z_{t-1} \leftarrow z_{t-1} - \eta \cdot g_t$
        $\beta_t \leftarrow \alpha_t / \alpha_{t-1}$
        $z_t \leftarrow \sqrt{\beta_t} z_t + \sqrt{1-\beta_t}\epsilon$
      end if
    end for
  end if
  $z_{t-1} \leftarrow \mathrm{DDIM}(z_t, z_{0|t})$
end for
return z_0

Algorithm 5 Frame Guidance (flow matching)

Require: $\mathcal{I}$, t_E, t_L, repeat step M, step size η, guidance loss $\mathcal{L}_e$, model $v_\theta(\cdot,\cdot)$
$z_T \sim \mathcal{N}(0, I)$
$\mathcal{J} \leftarrow$ Frame-Idx-to-Latent-Idx($\mathcal{I}$)
for $t = T, ..., 1$ **do**
  if $t > t_L$ **then** {Guidance step}
    for $m = 1, ..., M-1$ **do**
      $z_{0|t} \leftarrow z_t - \sigma_t \cdot v_\theta(z_t, t)$
      $z^{\mathcal{J}}_{0|t} \leftarrow$ Latent-Slicing$(z_{0|t}, \mathcal{J})$
      $x^{\mathcal{I}}_{0|t} \leftarrow \mathcal{D}(z^{\mathcal{J}}_{0|t})$
      $g_t = \nabla_{z_t} \mathcal{L}_e(x^{\mathcal{I}}_{0|t}, c_{\texttt{frames}})$
      if $t > t_E$ **then** {Early steps}
        $z_t \leftarrow z_t - \eta \cdot g_t$
      else {Later steps}
        $z_t \leftarrow$ Time-Travel-F$(z_{0|t}, g_t)$
      end if
    end for
  end if
  $z_{t-1} \leftarrow z_t + (\sigma_{t-1} - \sigma_t) \cdot v_\theta(z_t, t)$
end for
return z_0

(a) Time-travel in early steps



(b) Deterministic update in early steps (VLO)

Figure 23: **Importance of VLO in early steps.** (a) The time-travel method fails to produce a proper layout. (b) VLO successfully generates the video, capturing the view transition from left to right. Each top row shows early inference steps, and the bottom row shows the final generated results. Red boxes are drawn at the same fixed location across all frames.

(a) Shortcut method with video diffusion (b) Visualization of generated video

Figure 24: Shortcut-based approaches (He et al., 2024; Rout et al., 2025; Nair and Patel, 2024) lead to temporal disconnects in video generation.

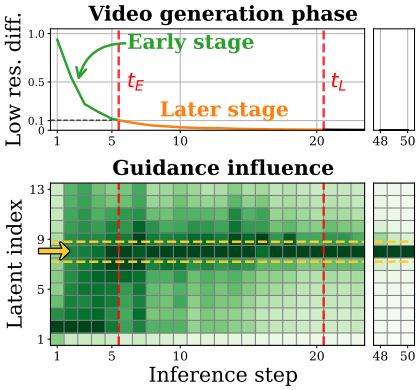
C.3 Video Latent Optimization (VLO) for Flow Matching

As noted in Section A.2, we follow the time convention of diffusion models by reversing the flow matching time axis, aligning $t = 0$ with clean data and $t = T$ with pure noise.

In Algorithm 5, we extend our Frame Guidance to video generation models, which employ the flow matching (Lipman et al., 2022) for their generative modeling (e.g., Wan (Wang et al., 2025a) and LTX (HaCohen et al., 2024)). Similar to the diffusion case in Equation 2, we apply the latent slicing (Lines 7) and optimize the current latent z_t through the guidance loss g_t (Lines 9-11). Specifically, we predict the clean sample $z_{0|t}$ by based on the tweedie-like formula in Equation 4.

Time-travel for flow matching However, directly applying the time-travel trick to flow matching is non-trivial, as a single forward step (Line 5 in Algorithm 2) is not explicitly defined in the context of flow matching. While renoising in time travel is effective for mitigating accumulated sampling errors, it cannot be directly utilized here. Our deterministic optimization excludes renoising entirely and can be applied as is, but performing it fully during inference, as in diffusion, can result in over-saturated samples or temporally disconnected videos.

To address this, we adopt a simple alternative: instead of stepping from t to $t-1$, we move directly from t to 0 (i.e., the estimated clean latent), apply guidance there, and then simulate a forward step from 0 back to t. Although a single forward step is not defined in flow matching, it is still possible to apply the forward process for time t from clean data. While this process introduces higher stochasticity than a single diffusion step, applying it in the later stages of VLO, after the layout has already been established, does not significantly disrupt the structure. This makes it a viable option. Empirically, this approach enables the application of VLO to flow matching-based models as well.



t_E step (·/50)	FVD (↓)	FID (↓)
0 (Time-travel)	725.30	59.03
2	730.38	59.18
4	569.02	57.74
6	**514.47**	58.03
8	572.55	**56.18**
10	617.42	59.12
12	662.37	58.17
14	642.01	58.36
16 (≈ Deter.)	662.63	58.80

Figure 25: **(Left)** Video generation phases and the corresponding guidance influence maps. **(Right)** Ablation study on t_E.

C.4 Importance of gradient propagation via denoising network

In training-free guidance for image generation, a "shortcut" (Rout et al., 2025; He et al., 2024; Nair and Patel, 2024) method has been proposed that utilizes a proximal gradient approach to *bypass* back-propagation through the denoising network. This strategy significantly reduces memory usage and enables efficient sampling for gradient-based optimization. While effective for static images, directly applying this method to video generation poses challenges due to the temporal characteristic of video data.

Specifically, when guidance is applied to only a few frames, the resulting video often becomes temporally inconsistent. As illustrated in Figure 24, the latents corresponding to the guided frames are updated to resemble the target frames, and adjacent frames may also partially align. However, earlier frames remain disconnected, and the guided frames themselves may exhibit unnatural artifacts. This is because temporal priors, crucial for maintaining coherence across frames, are primarily encoded in the denoising network. Consequently, for video generation tasks where temporal consistency is critical, gradient propagation through the denoising network is essential.

C.5 The choice of the timestep range for stages (t_E, t_L)

As discussed in Section 4.2, VLO employs a hybrid strategy that applies different update rules depending on the generation stage. We define the early stage, where deterministic updates are applied, as complete once the low-frequency structure of the video stabilizes. Concretely, this is when the difference from the final layout falls below 20% of the difference from the initial step, as shown in Figure 25 left top. To quantify this, we measure the L2 distance in the low-frequency region across inference steps, which confirms that video layouts are largely determined within the first few steps. Based on this stabilization criterion, we set t_E automatically rather than tuning it manually according to downstream video quality.

We further conduct an ablation study on t_E using the keyframe-guided generation task across 20 DAVIS videos (Figure 25 right). The results show that the best performance occurs at $t_E = 6$, which closely matches our stabilization-based criterion. Notably, performance remains robust over a range of nearby values, indicating that the method is relatively insensitive to the precise choice of t_E. Furthermore, as shown in Figure 25 left, the gradient propagation map reveals that gradients become increasingly localized around the guided frame. This trend mirrors the behavior of the video generation process itself.

Regarding t_L, which specifies how long guidance is applied, it correlates most directly with inference time. This reveals a trade-off between the strength of guidance and the additional NFEs. In practice, we set t_L such that the overall runtime does not exceed $4\times$ that of the base model's inference time.

C.6 Is temporal locality limited on rapid motion video?

We conduct the same experiment in Figure 4(b) on a rapid motion video. Specifically, we replace a single frame with a black image and measured the difference between the latents of the original video and the modified video. To simulate rapid motion, we *sparsely* sample the frames from a video at a rate of 16 times the original frame rate, which results in large differences between adjacent frames.

As shown in Figure 26, we still observe the same pattern as in Figure 4(b), with activations remaining localized around the modified frame. Notably, this behavior persists even when we extremely increase the motion speed by up to 16×, indicating that the same localized pattern consistently holds. This result confirms that temporal locality is largely independent of motion speed, as it reflects how latent frames are mapped to video frames during latent decoding. Temporal locality stems from the design of CausalVAE, not from the video content itself.

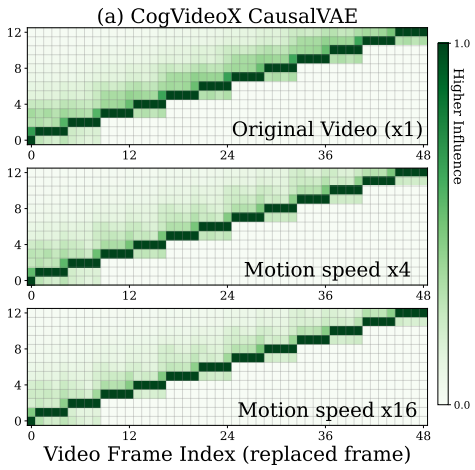
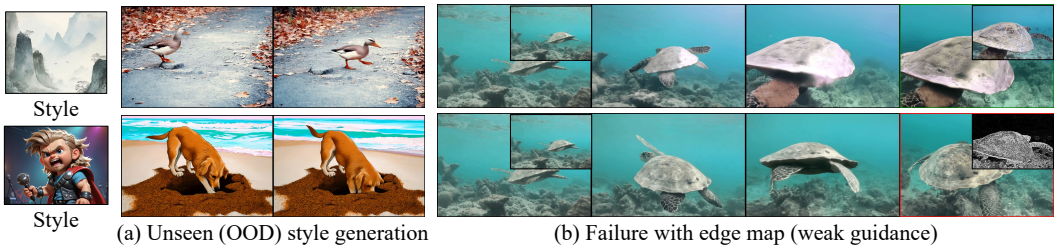


Figure 26: Temporal locality persists even under rapid motion.



(a) Unseen (OOD) style generation (b) Failure with edge map (weak guidance)

Figure 27: Failure cases of Frame Guidance.

D Limitations

Although Frame Guidance is training-free and supports various applications, it has some limitations:

(1) The computational cost of guidance sampling is higher than that of training-based methods. Since it requires back-propagation and multiple predictions, the inference speed is approximately up two to four times slower than that of the base model, depending on the task. This issue is particularly significant in video generation, which is computationally intensive. We leave addressing this inefficiency to future work.

(2) While our method is model-agnostic, it is heavily dependent on the performance of the base model. Since our approach samples videos that align with given conditions within the generation distribution of the base model, it struggles to generate videos that are either too dynamic or contain fine-detailed objects the model has not encountered during training. For example, as shown in Figure 27(a), it often fails to generate unseen (OOD) styles, such as 3D animation character.

(3) As discussed in FreeDom (Yu et al., 2023), it is inherently difficult for training-free guidance to control fine-grained structural features. For example, as shown in Figure 27(b), when we apply Frame Guidance using edge maps obtained from Sobel filtering, the guidance often becomes weak or unstable, even when combined with a large number of iterations, though it works well with the RGB keyframe. In such cases, training-required methods (Jiang et al., 2025; Li et al., 2025b) offer a more reliable alternative.

The use of Large Language Models (LLMs)

In this paper, we used large language models (LLMs) to assist with writing refinement, such as checking for grammatical errors.