Frame-Level Captions for Long Video Generation with Complex Multi Scenes

Anonymous Author(s)
Affiliation
Address
email

Abstract

Generating long videos that can show complex stories, like movie scenes from scripts, has great promise and offers much more than short clips. However, current methods that use autoregression with diffusion models often struggle because their step-by-step process naturally leads to a serious error accumulation (drift). Also, many existing ways to make long videos focus on single, continuous scenes, making them less useful for stories with many events and changes. This paper introduces a new approach to solve these problems. First, we propose a novel way to annotate datasets at the **frame-level**, providing detailed text guidance needed for making complex, multi-scene long videos. This detailed guidance works with a **Frame-Level Attention Mechanism** to make sure text and video match precisely. In inference, we develop **Parallel Multi-Window Denoising**, a new method that handles a long video as multiple overlapping windows. These windows are processed in parallel, and the noise prediction in overlapping areas is averaged, which allows bidirectional information interaction and introduces no error accumulation. A key feature is that each part (frame) within these windows can be guided by its own distinct text prompt. Our training uses **Diffusion Forcing** to provide the model with the ability to handle time flexibly. We tested our approach on difficult VBench 2.0 benchmarks ("Complex Plots" and "Complex Landscapes") based on the WanX2.1-T2V-1.3B model. The results show our method is better at following instructions in complex, changing scenes and creates high-quality long videos. We plan to share our dataset annotation methods and trained models with the research community.

1 Introduction

The ability to create long video sequences from text instructions opens exciting doors for rich, evolving stories, such as turning scripts into videos, producing short films, or showing complex processes. Unlike short clips, long videos provide the needed duration for multiple connected scenes, detailed character interactions, and consistent plotlines that follow complex user requests [17, 42, 32, 23, 15, 20, 22]. However, creating high-quality, consistent, and accurate long videos from text is still a major challenge for current generative models.

A primary difficulty lies in the common autoregressive (step-by-step) methods used with diffusion models to make longer videos. Their sequential way of working naturally leads to errors accumulation over time. This shows up as lower visual quality, the video drifting away from the original text's meaning, and a loss of consistency as the video gets longer, seriously weakening the quality of extended generations [14, 18, 37, 36]. Furthermore, much current research on long videos deals with single, continuous scenes or slowly changing environments. This limited focus reduces the usefulness of long video generation for dynamic stories with many events, which is a key goal for creative

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

(a). The foundational models for video generation suffer from confusion and error accumulation.

(b). Our method effectively mitigates confusion and error accumulation.

Figure 1: Illustration of issues of error accumulation and semantic confusion in the first row, and videos generated by ours are shown in the second row.

applications. Standard methods using single, general (global) video descriptions struggle with small, quick changes, leading to timing issues [26, 18, 19]. While some approaches try to make multi-shot videos by first detecting scene cuts with tools like PySceneDetect and then processing these shots, as in Presto [30] and Long Context Tuning (LCT) [12], these methods can be complicated, risk losing information, and depend heavily on good shot detection and captioning. They often still describe video at a "shot-text" level, which doesn't fully capture smooth, continuous changes.

To solve these basic problems, our work offers a new way of thinking, focused on highly detailed, frame-level text guidance and a novel non-sequential method for creating the video. Our first main contribution is an innovative **frame-level dataset annotation methodology**. We move beyond general or shot-level captions to provide very detailed text descriptions for each conceptual part (or latent segment) of a video. This rich information about meaning is essential for guiding models to understand and create the complex, changing details needed for stories with many scenes and detailed prompts. This directly addresses the limits of less detailed global or shot-level captions. This detailed annotation is designed to work closely with our **Frame-Level Attention Mechanism**, which clearly links each video segment's visual features to its specific text description, improving content accuracy and consistency over time (Section 4.2).

To properly use such detailed and dynamic text prompts, models need to be trained to handle time flexibly. We achieve this using **Diffusion Forcing** (Section 4.3), a training strategy that shows the model video segments being denoised at different rates. This prepares it to manage varied timing patterns and allows for strong, adaptable inference. Building on these training improvements, we introduce our second major innovation: **Parallel Multi-Window Denoising (PMWD)**, a new inference method designed to create very long videos that are highly consistent (Section 4.4). PMWD divides the target long video into multiple overlapping sections (windows), usually matching the model's training window size. Importantly, unlike step-by-step methods, all these windows are processed *at the same time (in parallel)* during each step of the diffusion denoising process. The data in the overlapping areas between windows is then averaged. This averaging not only ensures smooth connections but also allows information to flow in both directions, meaning later parts of the video can help refine earlier ones. A special feature of PMWD is that each conceptual frame, even within these simultaneously processed windows, can be guided by its own distinct, frame-level prompt.

We test our approach thoroughly using highly challenging benchmarks, specifically the "Complex Plots" and "Complex Landscapes" prompt categories from VBench 2.0 [45]. We use the state-of-the-art open-source WanX2.1-T2V-1.3B model [32] as our base. Our experiments show that our combined frame-level approach is much better at following prompts when creating very long videos with multiple scenes, different characters, and complex actions.

In summary, our main contributions are:

- **Scalable Frame-Level Dataset Methodology:** We introduce an efficient and scalable approach for constructing datasets with dense, frame-by-frame textual annotations. This enables highly granular video-text alignment crucial for generating complex, multi-scene narratives without relying on traditional shot detection.
- **Frame-Level Attention for Precise Guidance:** We propose a novel attention mechanism that directly couples each video segment's visual features with its unique frame-level prompt. This significantly enhances semantic fidelity, content accuracy, and temporal consistency in generated videos.

- **Parallel Multi-Window Denoising for Coherent Long Video Generation:** We develop PMWD, a inference strategy that processes a long video as multiple overlapping windows, denoised simultaneously in parallel. Guided by distinct frame-level prompts and leveraging overlap averaging for bidirectional context, PMWD effectively avoids the error accumulation common in sequential methods. This is enabled by training strategies like Diffusion Forcing that provide temporal flexibility.
- **State-of-the-Art Performance on Complex Videos:** Through comprehensive evaluations on challenging VBench 2.0 benchmarks ("Complex Plots" and "Complex Landscapes") using the WanX2.1-T2V-1.3B model, we demonstrate our integrated approach's superior ability to follow intricate prompts in multi-element long videos, achieving high-fidelity results with minimal error accumulation.

2 Related Work

Video Generation Dataset. Large-scale video datasets have driven advancements in video generation, but many existing datasets like YouCook2 [47], VATEX [38], and ActivityNet [5] were not designed for this purpose and lack fine-grained annotations. Similarly, large-scale datasets like YTTemporal-180M [43] and HD-VILA-100M [40] suffer from low-quality captions generated through speech recognition, limiting their utility for high-quality video generation. Datasets like Panda-70M [8] offer extensive data but rely on simplistic global descriptions, which hinder the model's ability to capture fine temporal details. Newer datasets, such as Koala-36M [35] and LongTake-HD [41], provide more detailed annotations but still rely on segment-level or shot-based annotations, limiting long-duration video generation. In contrast, our method introduces a frame-level captioning approach, where each frame is independently annotated with a description that maintains contextual relevance to the preceding and succeeding frames. This ensures better alignment between visual content and text while preserving the temporal continuity and motion dynamics, ultimately improving the overall quality of long-form video generation.

Long Video Generation. Video generation has evolved from simple single-shot models to more complex long-form and multi-scene models. Early methods relied on GANs [9, 29, 31, 39], constrained by single-domain datasets. Diffusion models [3, 11, 2, 1, 10] introduced temporal layers, enabling motion modeling. DiT-based architectures [4, 25, 28, 17, 42, 32, 23, 15, 20] have achieved tremendous success in scaling diffusion transformers, significantly enhancing video quality. However, these models were limited to generating short clips. FreeNoise [27] and StreamingT2V [13] extended video sequences using auto-regressive methods and temporal attention mechanisms. Gen-L-Video [34] processes videos as sequences of overlapping short clips and employs a temporal co-denoising technique, wherein multiple predictions for each individual frame are averaged. Despite these advancements, challenges in content diversity and temporal consistency persisted. The Diffusion Forcing [6] paradigm addressed these issues by combining diffusion's high-quality generation with auto-regressive models for sequence extension.

In multi-scene video generation, models like Mask2DiT [26], LCT [12], VideoStudio [21], SKYREELS-V2 [7], MovieDreamer [44], StoryAnchors [33], and VGoT [46] focused on scene-level consistency but struggled with temporal coherence across scenes. Recent methods, including StoryDiffusion [48] and MEVG [24], employed attention mechanisms to enhance visual and dynamic consistency. Our approach uses frame-level attention for dynamic scene extension without fixed scene durations, improving flexibility and coherence in long-form videos. Combined with Diffusion Forcing [6], our method ensures smooth scene transitions, extended video lengths, and maintains both visual richness and temporal consistency.

3 Frame-Level Dataset

Previous video-text datasets such as Panda70M [8] and Koala-36M [35] provide only global-level captions, resulting in coarse supervision that cannot reflect detailed visual changes within videos. LongTake-HD [41] offers shot-level sub-captions but still depends on explicit shot boundaries, making it difficult to model continuous motion and intra-shot dynamics. In contrast, our dataset uses frame-level uniform sampling and annotation, enabling dense and temporally continuous supervision. This design captures both subtle and significant changes without being limited by artificial segmentation, supporting more precise alignment between video frames and text descriptions. Overall, our dataset

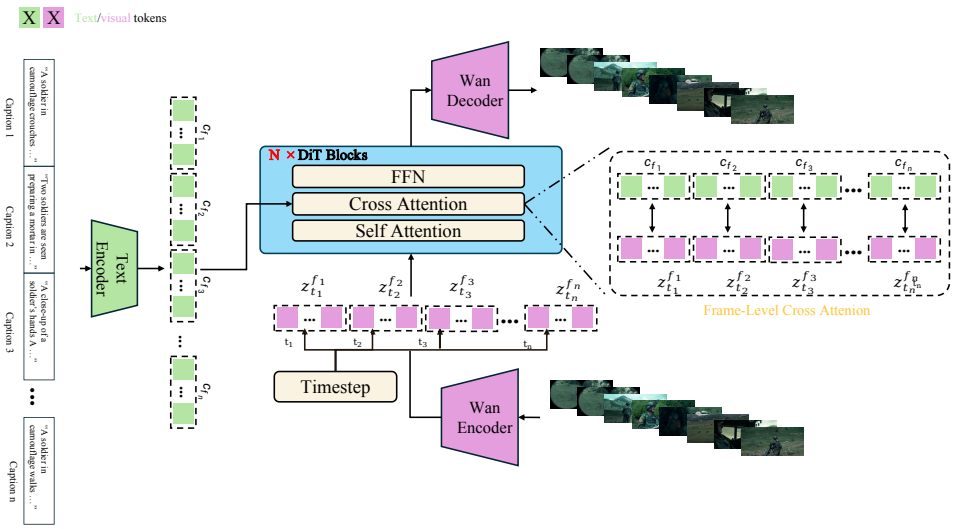

Figure 2: **Overview of the proposed frame-level training method**. Frame-Level Cross-Attention links the visual data of each video segment (latent token) directly to its own specific text description.

offers finer-grained, structurally consistent, and temporally faithful video-text supervision, facilitating improved learning of dynamic visual content.

Large-scale frame-level video dataset construction. We present a frame-level video dataset comprising 700,000 high-quality clips, designed to enhance fine-grained text-video alignment and provide dense semantic supervision for diffusion-based video generation models. The dataset systematically balances visual diversity, temporal continuity, and annotation precision, and can be further improved with larger scale in the future. We collect raw videos longer than 10 minutes from multiple platforms, remove near-duplicate content using perceptual hashing, and discard the first and last 10% of frames to ensure the sampled content is dynamic and semantically meaningful. Each processed video is evenly divided into four segments, from which an 8-second continuous clip is extracted. At a frame rate of 24 fps, one frame is sampled every 8 frames, resulting in approximately 24 frames per clip. This design balances temporal context and computational efficiency, and ensures compatibility with mainstream video VAE tokenization schemes, enabling precise one-to-one frame-token supervision.

Adaptive frame-level annotation. We utilize multimodal large language models to generate frame-level captions, automatically choosing between shared or independent descriptions based on the degree of visual change. Identical captions are assigned to frames with minimal differences, while significant changes trigger independent frame-level descriptions, achieving unified and adaptive semantic alignment. To enhance consistency and richness, we design structured annotation prompts that require each frame description to cover main subjects, actions, environment, shot size, and camera angle. Outputs strictly follow JSON format with no redundant commentary, ensuring precise semantic and structural alignment at the frame level. We provide the full frame-level annotation prompt, specifying formatting and content requirements to facilitate reproducibility and further research; the template is too long to show here, see appendix. During inference, we use gemini pro 2.5 to convert a user input from short/detailed caption to a frame-level detailed caption. More details can be found in appendix.

4 Method

Our work introduces a set of interconnected improvements for DiT-based video diffusion models, aimed at generating complex, long videos. We focus on enhancing how models understand text instructions for each video part, how they handle timing and changes, and how they create long, consistent videos during inference.

4.1 Overall Framework

Current Diffusion Transformer (DiT) based models are skilled at creating high-quality short videos. They typically use a VAE (Variational Autoencoder) to compress videos into compact latent data, and a DiT then generates video from these latents. However, when generating long videos with detailed stories and dynamic action, these models face several basic problems:

- **Imprecise Content Control:** Using a single text description (caption) for the entire video often leads to unclear or mixed-up details for different parts of the video, making it hard to accurately control specific events or elements across various scenes.
- **Limited Handling of Timing:** Standard models usually denoise all parts of the video at the same rate in each step. This restricts their ability to show varied motion speeds, different pacing in scenes, or sudden changes effectively.
- **Difficulty with Long Video Coherence:** When creating long videos by generating segments one after another from models trained on short clips, errors tend to build up. This can make the video lose consistency over time.

To address these key challenges, we propose three main contributions:

1. A **Frame-Level Cross-Attention** mechanism (Section 4.2) for precise, localized text-based control over the content of each video segment during training.
2. A **Diffusion Forcing** training strategy (Section 4.3) to teach models how to handle varied timing by exposing them to video segments denoised at different rates.
3. A **Synchronized Multi-Window Denoising (PMWD)** inference method (Section 4.4), designed to generate long, coherent videos by significantly reducing the build-up of errors.

While these principles can be applied to many DiT-based video models, we demonstrate our methods by adapting and fine-tuning the WanX2.1-T2V-1.3B [32] framework, a well-known open-source model, as our base.

4.2 Frame-Level Cross Attention

To accurately control video content in line with detailed narratives, we introduce Frame-Level Cross-Attention. This method links the visual data of each video segment (latent token) directly to its own specific text description. Simultaneously, the DiT's standard self-attention mechanism continues to capture overall temporal relationships, ensuring smooth motion. This approach provides both precise local content guidance and global video coherence.

Our process starts by assigning an independent text description to each conceptual "frame" (latent unit) of a video. When the original video is converted into latent data by the VAE, each resulting latent token z_f is directly paired with the embedding of its corresponding frame-level caption, c_f. This creates a detailed, one-to-one mapping between text and video segments over time, offering exact guidance for generation. We modify the DiT's cross-attention mechanism so that each latent token z_f attends exclusively to its paired caption embedding c_f, rather than to a single caption shared by the entire video. Formally, this is:

$$\text{CrossAttention}(q_f, c_f) = \text{Softmax}\left(\frac{q_f W_q (c_f W_k)^T}{\sqrt{d}}\right)(c_f W_v), \tag{1}$$

where q_f is the query projected from z_f, and W_q, W_k, W_v are learnable matrices. This targeted attention mechanism reduces the unclear meaning that can arise from global captions, greatly improving text-to-video alignment and allowing for precise control over dynamic content within each segment.

4.3 Diffusion Forcing for Temporal Flexibility

Creating long videos with dynamic action and varied pacing requires the model to handle time flexibly. Standard diffusion models are often limited in this area because they apply the same noise level to all video segments at each step of the denoising process, restricting their ability to generate diverse visual qualities, dynamic changes, or quick scene transitions.

To give models this needed flexibility, we use a **Diffusion Forcing (DF)** strategy during training. This technique assigns an independent noise level to each video segment (latent token) in a training sequence. Specifically, we pick a reference segment, set its target noise removal stage (timestep), and then determine the noise stages for other segments in relation to it: preceding segments get "cleaner" (earlier) timesteps, and subsequent segments get "noisier" (later) timesteps. This approach maintains temporal smoothness while training the model to manage different denoising states simultaneously within one sequence.

This training approach makes the model highly adaptable at inference time. By adjusting a "step-size" parameter—which controls the allowed difference in noise schedules between adjacent segments—we can smoothly shift the generation style. We can opt for fully synchronized diffusion (small step-size, for high consistency) or for more dynamic, evolving outputs (large step-size, resembling autoregressive generation). This adaptability allows the model to produce either smooth, consistent videos or to progressively unfold complex scene transitions and actions as guided by the text prompts. Furthermore, already partially denoised historical segments can serve as stable conditions for generating later segments, aiding long-range consistency without forcing all segments to share the same noise level at the same time.

4.4 Flexible Inference Modes for Long Video Generation

The temporal flexibility gained from Diffusion Forcing during training allows for various inference methods to generate videos much longer than the training segments (the "*train short, test long*" approach).

Sequential Sliding Window Approaches. Common methods for long video generation use a sequential sliding window. These include simple autoregressive techniques, where a new segment of M latents is generated based on $N - M$ previous latents from an N-latent window (often re-noising the context), and more advanced methods like FIFO-Diffusion [16], which uses a queue with diagonally progressing noise levels for better temporal consistency. However, a core problem with all such step-by-step sequential methods is the unavoidable build-up of errors, which reduces quality and long-range consistency in very long videos.

Parallel Multi-Window Denoising (PMWD). To effectively overcome this error accumulation problem, we introduce PMWD. This novel inference strategy takes full advantage of our frame-level prompt system to generate long videos more as a complete whole, rather than piece by piece. For a target long video of L latents, we view it as K overlapping windows, each the length of a training segment. Crucially, all K windows are processed *in parallel* (at the same time) during each step of the diffusion denoising process. Every latent, whether new or historical context, is guided by its own dedicated frame-level prompt. This parallel, parallel method for the entire sequence inherently avoids the cascading error build-up seen in typical autoregressive techniques. Latents located in the overlapping regions between adjacent windows are averaged after each denoising step. This averaging, along with the parallel processing, allows information to flow in both directions (bidirectionally) between an earlier and a later window. Unlike methods where only the past influences the future, PMWD allows upcoming video segments to help refine earlier ones. This is especially useful for creating natural-looking scene changes and maintaining consistency in stories with multiple scenes.

5 Experimental Results

5.1 Experimental Setup

We fully fine-tune the open-source WanX-2.1-T2V-1.3B model with Diffusion Forcing technique on resolution 81x480x832 for 100,000 iterations using our internal dataset (detailed in Section 3) of dense frame-level annotations. Training occurred on H-series GPUs with a global batch size of 64.

5.2 Evaluation Dataset

We evaluate the model's capability to generate complex videos by utilizing prompts from the VBench 2.0 benchmark, specifically focusing on the **Complex Plots** and **Complex Landscapes**.

Complex Plots assess the model's ability to construct coherent and consistent multi-scene narratives based on prompts describing multi-stage events. These prompts often involve extended descriptions

(150+ words) outlining a sequence of actions or a story with multiple acts, challenging the model to maintain plot consistency and logical flow throughout the generated video.

Complex Landscapes evaluate whether the model can faithfully translate long-form landscape descriptions (150+ words) into video, including multiple scene transitions dictated by camera movements. These prompts test the model's understanding of spatial relationships and its ability to render dynamic changes in the environment as described in the text.

5.3 Evaluation Metrics

We evaluate video quality using metrics for overall video-text alignment and also propose a new metric for the issue of semantic confusion in multi scenes generation. Let P_g be the global prompt, V the generated video, $\{P_1, \ldots, P_F\}$ the sequence of F frame-level prompts, and $\{V_1, \ldots, V_F\}$ the corresponding sequence of generated frames.

Standard VBench Evaluation. To provide a comprehensive assessment of fundamental video quality aspects, particularly for the complex scenarios presented by our chosen VBench 2.0 prompt categories (Complex Plots and Complex Landscapes), we incorporate a curated subset of established metrics from the VBench benchmark. This evaluation focuses on key indicators such as: aesthetic quality, image quality, and motion smoothness. These selected metrics offer standardized measures of the perceptual quality and spatio-temporal coherence of the generated videos.

Video-Level Video-Text Similarity. This standard metric evaluates overall coherence between P_g and V. $\Phi_V(V)$ represents overall video features (uniformly sample 8 frames as input of ViClip).

$$\mathcal{S}_{\text{global}} = \text{Sim}(\Phi_T(P_g), \Phi_V(V)) \tag{2}$$

where we use a pre-trained vision-language model (e.g., ViCLIP) for text embeddings $\Phi_T(\cdot)$ and video/latent 'frame' embeddings $\Phi_V(\cdot)$, with $\text{Sim}(\cdot, \cdot)$ denoting cosine similarity.

Confusion Degree (CD). Despite the widespread use of $\mathcal{S}_{\text{global}}$, this global metric may assign favorable scores even when content from different scenes are inappropriately combined. To pinpoint such temporal and semantic inaccuracies, we introduce the Confusion Degree (CD). A high CD score reveals a model's difficulty in maintaining a clear, sequential narrative, often resulting in a muddled or incoherent visual story. We first define two fundamental frame-level similarity metrics as follows:

$$\begin{aligned} S_{TT}(P_i, P_j) &= \text{Sim}(\Phi_T(P_i), \Phi_T(P_j)) \\ S_{TF}(P_i, V_j) &= \text{Sim}(\Phi_T(P_i), \Phi_V(V_j)) \end{aligned} \tag{3}$$

, where $S_{TT}(P_i, P_j)$ represents **frame-level text-text similarity** and $S_{TF}(P_i, V_j)$ represents **frame-level text-frame similarity**. Then $\tilde{S}_{TT}(P_i, P_j) = S_{TT}(P_i, P_j)/S_{TT}(P_i, P_i)$ and $\tilde{S}_{TF}(P_i, V_j) = S_{TF}(P_i, V_j)/S_{TF}(P_i, V_i)$ are applied as normalization function to ensure $\tilde{S}_{TT}(P_i, P_i) = 1$ and $\tilde{S}_{TF}(P_i, V_i) = 1$.

The confusion degree of a text P_i in the generated video V is defined as:

$$\text{CD}(P_i) = \sum_{j \in \{1,\ldots,F\}} \max(0, \tilde{S}_{TF}(P_i, V_j) - \tilde{S}_{TT}(P_i, P_j)) \tag{4}$$

where $\tilde{S}_{TF}(P_i, V_j) - \tilde{S}_{TT}(P_i, P_j)$ indicates that the content of P_i is more aligned with frame V_j than its inherent semantic relationship with P_j would suggest, thereby signaling confusion. Then the confusion of a video V is defined as

$$\text{CD} = \frac{1}{F} \sum_{i=1}^{F} \text{CD}_i \tag{5}$$

, representing the average confusion degree across all frames. Lower CD values indicate superior narrative consistency and reduced semantic confusion throughout the video.

5.4 Comparison and Discussion

Analysis of Video Generation under Complex Prompts. Tab. 1 provides a comparative analysis of models trained and inferenced using either global video-level or granular frame-level prompts. When

Method	Video Length	Prompts Type	Confusion Degree↓	Video-level Text-Video Consistency↑	Frame-level Text-Video Consistency↑	Motion Smoothness↑	Aesthetic Quality↑	Image Quality↑
DF + Video-level Prompt	5s	Complex Plot	0.2952 ± 0.0461	0.2100 ± 0.0410	0.1635 ± 0.0282	98.43	**59.20**	**67.92**
DF + Video-level Prompt	30s	Complex Plot	0.2962 ± 0.0487	0.2053 ± 0.0368	0.1518 ± 0.0258	**98.63**	52.02	58.07
DF + Frame-level Prompt	30s	Complex Plot	$\mathbf{0.1385 \pm 0.0498}$	$\mathbf{0.2196 \pm 0.0309}$	$\mathbf{0.2054 \pm 0.0231}$	98.53	55.04	61.56
DF + Video-level Prompt	5s	Complex Landscape	0.2745 ± 0.0412	0.2101 ± 0.0341	0.1831 ± 0.0227	98.70	**61.32**	**59.61**
DF + Video-level Prompt	30s	Complex Landscape	0.2806 ± 0.0474	0.2066 ± 0.0351	0.1723 ± 0.0230	98.58	52.63	51.02
DF + Frame-level Prompt	30s	Complex Landscape	$\mathbf{0.1528} \pm 0.0479$	$\mathbf{0.2195 \pm 0.0326}$	$\mathbf{0.2139 \pm 0.0167}$	**98.99**	56.31	55.98

Table 1: Comparing video-level versus frame-level prompting for complex narrative videos. While global Video-Level Text-Video Consistency can yield misleadingly high scores despite internal scene blending or semantic confusion, metrics like Confusion Degree and frame-level consistency more effectively expose these flaws, highlighting the superior prompt adherence of frame-level strategies. Further analysis in Section 5.4.

Method	Prompt Level	Confusion Degree ↓	Video-level ↑ Text-Video Consistency	Frame-level ↑ Text-Video Consistency	Motion ↑ Smoothness	Aesthetic ↑ Quality	Image ↑ Quality
First-In-First-Out (FIFO)	Video	0.2962 ± 0.0487	0.2053 ± 0.0368	0.1518 ± 0.0258	98.63	52.02	58.07
First-In-First-Out (FIFO)	Frame	0.2416 ± 0.0514	0.2100 ± 0.0370	0.1660 ± 0.0266	**98.81**	51.32	59.25
Sequential Sliding Window	Frame	0.1773 ± 0.0550	0.2134 ± 0.0312	0.1842 ± 0.0227	98.80	52.04	60.11
Parallel Multi-Window Denosing	Frame	$\mathbf{0.1385 \pm 0.0498}$	$\mathbf{0.2196 \pm 0.0309}$	$\mathbf{0.2054 \pm 0.0231}$	98.53	**55.04**	**61.56**

Table 2: Inference method comparison for 30s complex plot videos. Parallel Multi-Window Denoising (PMWD) achieves lower error accumulation (improved aesthetic/image quality) and better prompt adherence (reduced Confusion Degree, higher text-video consistency) versus causal methods (FIFO, Sliding Window). Detailed analysis in Section 5.4.

generating short videos (e.g., 5 seconds) conditioned on a single **video-level prompt**, the model operates closer to an ideal scenario without temporal error accumulation. However, such prompts often lead to a high Confusion Degree (CD), as the model struggles to render extensive semantic information within a condensed timeframe, resulting in blended or muddled content.

Conversely, employing **frame-level prompts** demonstrates a marked improvement in prompt adherence, evidenced by lower CD scores alongside high frame-level consistency metrics. This enhanced ability to follow detailed, segmented instructions makes the frame-level prompting strategy more reliable and effective for generating coherent multi-scene long videos. Furthermore, metrics such as aesthetic and image quality serve as indirect indicators of error accumulation; significant degradation in these scores over time typically reflects compounding errors. This accumulation is an inherent consequence of the causal nature of sequential generation processes, a fundamental issue that even precise frame-level semantic guidance cannot resolve on its own when operating within such autoregressive frameworks.

Comparative Analysis of Long Video Inference Strategies. We further analyze the efficacy of different inference strategies for extending video generation beyond training lengths, comparing our proposed Parallel Multi-Window Denoising (PMWD) with established sequential methods like FIFO-Diffusion and naive sliding windows.

Sequential approaches, by their nature, tackle long video generation segment by segment. Naive sliding window techniques autoregressively generate a new chunk of latents conditioned on a limited history of prior latents (often re-noised to manage error). FIFO-Diffusion [16] offers a more sophisticated sequential mechanism, processing a queue of latents with diagonally increasing noise levels to output one clean latent per step, thereby aiming for better temporal consistency through extended context. While these methods incorporate mechanisms to manage error, such as FIFO's broader context or the re-noising of historical data in naive sliding windows, they fundamentally struggle with the *inevitable accumulation of errors* over very long sequences. This compounding error degrades long-range coherence and overall video quality.

Our proposed **Parallel Multi-Window Denoising (PMWD)** is architecturally designed to overcome this critical limitation. Instead of sequential generation, PMWD processes the entire target long video (composed of multiple overlapping windows) *simultaneously* at each denoising step, with each latent guided by its specific frame-level prompt. This parallel, holistic approach fundamentally disrupts the chain of error propagation seen in sequential methods. The averaging of latents in overlapping regions is a key aspect of PMWD. This, combined with parallel processing, not only fuses information effectively but also transforms the strictly causal dependency of sequential models into a *bidirectional contextual influence*, where information from temporally subsequent windows can refine earlier ones. This capability is particularly advantageous for rendering naturalistic scene transitions and ensuring global narrative consistency.

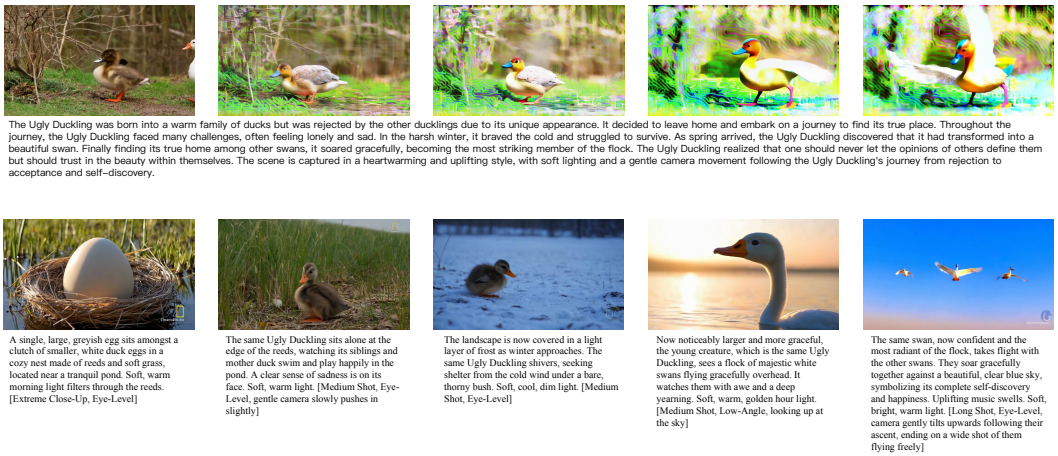

Figure 3: Complex Plot Generation. This figure illustrates the impact of different prompting strategies on visual storytelling performance in a complex narrative task based on The Ugly Duckling. The first row shows results generated using DF with a single global prompt, while the second row presents results from our proposed method that combines DF with multiple tailored prompts (multi-prompting). Our method demonstrates significantly improved coherence, reduced error accumulation, and less narrative confusion across the sequence. The images generated with multi-prompting maintain better stylistic and semantic consistency, showcasing its superiority over the global prompt approach in handling complex plot developments.

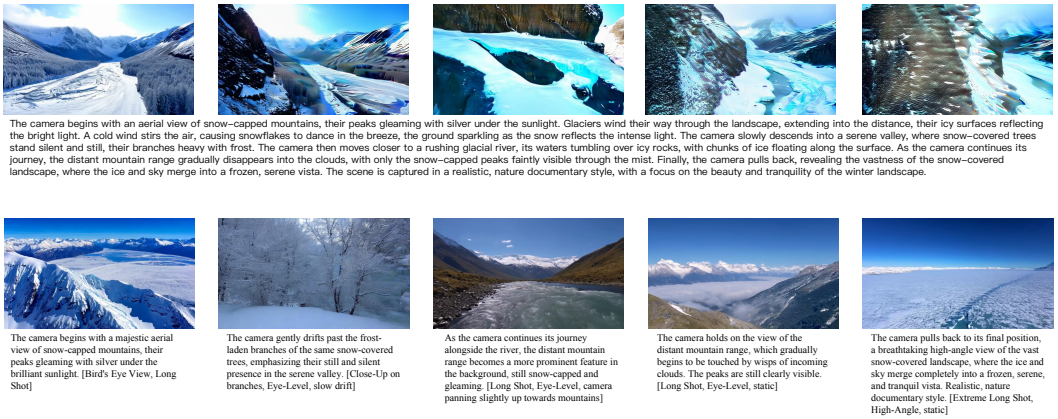

Figure 4: Complex Landscapes Generation. This figure compares two prompting strategies for generating complex scenes. The top row uses DF + global prompt, while the bottom row shows results from our method: DF + multi-prompt. Our approach significantly reduces content drift and error accumulation across frames. By using multiple prompts tailored to each scene segment, it achieves higher accuracy and coherence, capturing the complexity and progression of the winter landscape more effectively than the global prompt method.

6 Conclusion

Generating long, narratively complex videos with high fidelity remains challenging, primarily due to issues with coarse semantic guidance and the error accumulation inherent in common sequential generation techniques. We propose a comprehensive solution combining fine-grained frame-level annotations, novel training strategies, and a Parallel Multi-Window Denoising (PMWD) inference method. Our experiments on demanding VBench 2.0 benchmarks demonstrate that this integrated system significantly improves prompt adherence for complex, multi-scene narratives in ultra-long videos, achieving high-quality results with minimal error accumulation.

References

[1] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, et al. Lumiere: A space-time diffusion model for video generation. In *SIGGRAPH*, pages 1–11, 2024.

[2] 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.

[3] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, pages 22563–22575, 2023.

[4] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1:8, 2024.

[5] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970, 2015.

[6] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

[7] Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Juncheng Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengchen Ma, et al. Skyreels-v2: Infinite-length film generative model. *arXiv preprint arXiv:2504.13074*, 2025.

[8] Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, et al. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *CVPR*, pages 13320–13331, 2024.

[9] Mengyu Chu, You Xie, Jonas Mayer, Laura Leal-Taixé, and Nils Thuerey. Learning temporal coherence via self-supervision for gan-based video generation. *TOG*, 39(4):75–1, 2020.

[10] Rohit Girdhar, Mannat Singh, Andrew Brown, Quentin Duval, Samaneh Azadi, Sai Saketh Rambhatla, Akbar Shah, Xi Yin, Devi Parikh, and Ishan Misra. Factorizing text-to-video generation by explicit image conditioning. In *ECCV*, pages 205–224. Springer, 2024.

[11] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023.

[12] Yuwei Guo, Ceyuan Yang, Ziyan Yang, Zhibei Ma, Zhijie Lin, Zhenheng Yang, Dahua Lin, and Lu Jiang. Long context tuning for video generation. *arXiv preprint arXiv:2503.10589*, 2025.

[13] Roberto Henschel, Levon Khachatryan, Daniil Hayrapetyan, Hayk Poghosyan, Vahram Tadevosyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Streamingt2v: Consistent, dynamic, and extendable long video generation from text. *arXiv preprint arXiv:2403.14773*, 2024.

[14] Emiel Hoogeboom, Alexey A Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models. In *International Conference on Learning Representations*, 2023.

[15] Haoyang Huang, Guoqing Ma, Nan Duan, Xing Chen, Changyi Wan, Ranchen Ming, Tianyu Wang, Bo Wang, Zhiying Lu, Aojie Li, et al. Step-video-ti2v technical report: A state-of-the-art text-driven image-to-video generation model. *arXiv preprint arXiv:2503.11251*, 2025.

[16] Jihwan Kim, Junoh Kang, Jinyoung Choi, and Bohyung Han. Fifo-diffusion: Generating infinite videos from text without training, 2024. URL `https://arxiv.org/abs/2405.11473`.

[17] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

[18] Chengxuan Li, Di Huang, Zeyu Lu, Yang Xiao, Qingqi Pei, and Lei Bai. A survey on long video generation: Challenges, methods, and prospects, 2024. URL https://arxiv.org/abs/2403.16407.

[19] Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model, 2025. URL https://arxiv.org/abs/2503.00200.

[20] Bin Lin, Yunyang Ge, Xinhua Cheng, Zongjian Li, Bin Zhu, Shaodong Wang, Xianyi He, Yang Ye, Shenghai Yuan, Liuhan Chen, et al. Open-sora plan: Open-source large video generation model. *arXiv preprint arXiv:2412.00131*, 2024.

[21] Fuchen Long, Zhaofan Qiu, Ting Yao, and Tao Mei. Videostudio: Generating consistent-content and multi-scene videos. In *ECCV*, pages 468–485. Springer, 2024.

[22] Guoqing Ma, Haoyang Huang, Kun Yan, Liangyu Chen, Nan Duan, Shengming Yin, Changyi Wan, Ranchen Ming, Xiaoniu Song, Xing Chen, Yu Zhou, Deshan Sun, Deyu Zhou, Jian Zhou, et al. Step-video-t2v technical report: The practice, challenges, and future of video foundation model, 2025. URL https://arxiv.org/abs/2502.10248.

[23] Guoqing Ma, Haoyang Huang, Kun Yan, Liangyu Chen, Nan Duan, Shengming Yin, Changyi Wan, Ranchen Ming, Xiaoniu Song, Xing Chen, et al. Step-video-t2v technical report: The practice, challenges, and future of video foundation model. *arXiv preprint arXiv:2502.10248*, 2025.

[24] Gyeongrok Oh, Jaehwan Jeong, Sieun Kim, Wonmin Byeon, Jinkyu Kim, Sungwoong Kim, and Sangpil Kim. Mevg: Multi-event video generation with text-to-video models. In *ECCV*, pages 401–418. Springer, 2024.

[25] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4195–4205, 2023.

[26] Tianhao Qi, Jianlong Yuan, Wanquan Feng, Shancheng Fang, Jiawei Liu, SiYu Zhou, Qian He, Hongtao Xie, and Yongdong Zhang. Mask2dit: Dual mask-based diffusion transformer for multi-scene long video generation. *arXiv preprint arXiv:2503.19881*, 2025.

[27] Haonan Qiu, Menghan Xia, Yong Zhang, Yingqing He, Xintao Wang, Ying Shan, and Ziwei Liu. Freenoise: Tuning-free longer video diffusion via noise rescheduling. *ICLR*, 2023.

[28] Team Seawead, Ceyuan Yang, Zhijie Lin, Yang Zhao, Shanchuan Lin, Zhibei Ma, Haoyuan Guo, Hao Chen, Lu Qi, Sen Wang, et al. Seaweed-7b: Cost-effective training of video generation foundation model. *arXiv preprint arXiv:2504.08685*, 2025.

[29] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *CVPR*, pages 3626–3636, 2022.

[30] Gabriel Tseng, Ruben Cartuyvels, Ivan Zvonkov, Mirali Purohit, David Rolnick, and Hannah Kerner. Lightweight, pre-trained transformers for remote sensing timeseries. *arXiv preprint arXiv:2304.14065*, 2023.

[31] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *CVPR*, pages 1526–1535, 2018.

[32] 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*, 2025.

[33] Bo Wang, Haoyang Huang, Zhiyin Lu, Fengyuan Liu, Guoqing Ma, Jianlong Yuan, Yuan Zhang, and Nan Duan. Storyanchors: Generating consistent multi-scene story frames for long-form narratives, 2025. URL https://arxiv.org/abs/2505.08350.

[34] Fu-Yun Wang, Wenshuo Chen, Guanglu Song, Han-Jia Ye, Yu Liu, and Hongsheng Li. Gen-l-video: Multi-text to long video generation via temporal co-denoising. *arXiv preprint arXiv:2305.18264*, 2023.

[35] Qiuheng Wang, Yukai Shi, Jiarong Ou, Rui Chen, Ke Lin, Jiahao Wang, Boyuan Jiang, Haotian Yang, Mingwu Zheng, Xin Tao, et al. Koala-36m: A large-scale video dataset improving consistency between fine-grained conditions and video content. *arXiv preprint arXiv:2410.08260*, 2024.

[36] Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Shiyu Huang, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. Lvbench: An extreme long video understanding benchmark, 2024.

[37] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent, 2024. URL `https://arxiv.org/abs/2403.10517`.

[38] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *ICCV*, pages 4581–4591, 2019.

[39] Yaohui Wang, Piotr Bilinski, Francois Bremond, and Antitza Dantcheva. Imaginator: Conditional spatio-temporal gan for video generation. In *WACV*, pages 1160–1169, 2020.

[40] Hongwei Xue, Tiankai Hang, Yanhong Zeng, Yuchong Sun, Bei Liu, Huan Yang, Jianlong Fu, and Baining Guo. Advancing high-resolution video-language representation with large-scale video transcriptions. In *CVPR*, pages 5036–5045, 2022.

[41] Xin Yan, Yuxuan Cai, Qiuyue Wang, Yuan Zhou, Wenhao Huang, and Huan Yang. Long video diffusion generation with segmented cross-attention and content-rich video data curation. *arXiv preprint arXiv:2412.01316*, 2024.

[42] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

[43] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. Merlot: Multimodal neural script knowledge models. *Neurips*, 34:23634–23651, 2021.

[44] Canyu Zhao, Mingyu Liu, Wen Wang, Jianlong Yuan, Hao Chen, Bo Zhang, and Chunhua Shen. Moviedreamer: Hierarchical generation for coherent long visual sequence. *arXiv preprint arXiv:2407.16655*, 2024.

[45] Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, Yu Qiao, and Ziwei Liu. Vbench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness, 2025. URL `https://arxiv.org/abs/2503.21755`.

[46] Mingzhe Zheng, Yongqi Xu, Haojian Huang, Xuran Ma, Yexin Liu, Wenjie Shu, Yatian Pang, Feilong Tang, Qifeng Chen, Harry Yang, et al. Videogen-of-thought: A collaborative framework for multi-shot video generation. *arXiv preprint arXiv:2412.02259*, 2024.

[47] Luowei Zhou, Chenliang Xu, and Jason Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, volume 32, 2018.

[48] Yupeng Zhou, Daquan Zhou, Ming-Ming Cheng, Jiashi Feng, and Qibin Hou. Storydiffusion: Consistent self-attention for long-range image and video generation. *Neurips*, 37:110315–110340, 2024.

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 made in the abstract and introduction are aligned with the theoretical and experimental results presented in the paper. The introduction provides a concise summary of what the reader can expect from the paper.

   Guidelines:

   - The answer NA 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 NA 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: Primarily covered in Section 6.

   Guidelines:

   - The answer NA 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: [NA]

Justification: This is not a purely theoretical paper.

Guidelines:

- The answer NA 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: We explained our framework in Section 4 and experiment settings in Section 5.

Guidelines:

- The answer NA 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: We will release code with instructions to reproduce the results.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper specifies training and test details in Section 5

Guidelines:

- The answer NA 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: [Yes]

Justification: The paper reports means and variances of the experiments. We made sure that for all the experiments conducted throughout the paper, we averaged across multi runs to make sure that the results are reliable.

Guidelines:

- The answer NA 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: We explained our settings in Section 5.

   Guidelines:

   - The answer NA 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 paper adheres to the ethical guidelines set forth by NeurIPS. We ensured that the research is conducted responsibly, with considerations for potential biases, fairness, and the broader impact on society.

   Guidelines:

   - The answer NA 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: [Yes]

    Justification: The paper discusses both potential positive societal impacts and negative societal impacts of the work performed in Section 6.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The work doesn't pose any risks. We either generated our own data or used a justified reliable benchmark.

    Guidelines:

    - The answer NA 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: We credited the author for the code package and benchmark dataset that have been used in the paper.

    Guidelines:

    - The answer NA 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: We will release code with detailed documentation and an appropriate license.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: We do not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: During inference, we employ LLMs to transform video-level prompts into frame-level prompts. During training, we similarly use the LLM to assist in constructing frame-level annotations, including generating shared or independent captions based on visual changes. The detailed LLM usage in both stages is described in the Section 3.

    Guidelines:

    - The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.