Frequency-Synchronized Boundary Coupling for Training-Free Multi-Prompt Long Video Generation

Yipan Xu[1] **Shuyong Gao**[2*] **Jiyuan Fu**[1] **Keliang Yin**[1] **Jiayu Chen**[1]
Qianyu Guo[3] **Yang Liu**[4] **Wenqiang Zhang**[1*]

[1]Shanghai Key Lab of Intelligent Information Processing,
College of Computer Science and Artificial Intelligence, Fudan University
[2]Naval Medical Center of PLA, Second Military Medical University
[3]Shanghai Institute of Virology, Shanghai Jiao Tong University
[4]College of Electronic and Information Engineering, Tongji University

Project page: `https://yipanxu-22.github.io/FreqSync-project`

Abstract

Text-to-video generation has made remarkable progress with diffusion models and transformer-based video architectures, yet generating coherent long videos from evolving textual descriptions remains challenging. Training long-video models from scratch requires substantial computation and large-scale multi-event data, while training-free multi-prompt generation must activate new prompt semantics without breaking temporal continuity. We identify a frequency-agnostic boundary coupling problem in existing transition strategies: weak coupling causes abrupt changes, flicker, and background jumps, whereas overly strong or coarse coupling may over-preserve source semantics and produce ghosting or duplicated subjects around prompt boundaries. To address this problem, we propose FREQSYNC, a training-free framework that formulates prompt transitions as frequency-selective boundary coupling. FREQSYNC combines Frequency-Synchronized Source-Conditioned Attention (FS-SCA) with Residual-Clipped Spectral Stitching (RCSS), enabling adjacent segments to share stable low- and mid-frequency structure and motion cues while keeping high-frequency prompt-specific details target-dominant under explicit residual constraints. Extensive experiments show that FREQSYNC achieves state-of-the-art transition smoothness and prompt alignment while maintaining strong perceptual video quality, demonstrating its effectiveness for coherent multi-prompt long video generation.

1 Introduction

Text-to-video generation has advanced rapidly with diffusion models and transformer-based video architectures. Representative systems, including Imagen Video, Make-A-Video, ModelScopeT2V, Stable Video Diffusion, LaVie, AnimateDiff, Lumiere, VideoCrafter, CogVideoX, Open-Sora, HunyuanVideo, Wan, and Movie Gen, have demonstrated strong short-video generation quality from a single prompt [19, 39, 46, 4, 47, 14, 3, 11, 50, 53, 26, 44, 35]. Recent diffusion transformer (DiT) and multi-modal diffusion transformer (MM-DiT) models further improve scalability and text-video modeling capacity [34, 12, 31]. However, real-world videos often involve multiple events, scene changes, camera motions, and evolving object states, which are more naturally described by a sequence of prompts rather than a single static description [43, 28, 7].

Multi-prompt long video generation requires each segment to follow its own prompt while maintaining smooth visual trajectories across prompt boundaries. Directly using a long prompt often causes weak prompt disentanglement, whereas independently generating and concatenating clips leads to

*Corresponding authors.

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

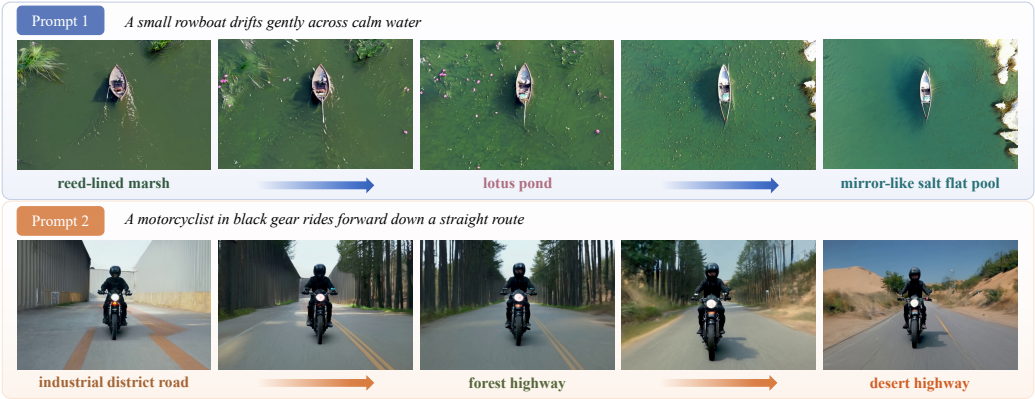


Figure 1: Teaser of FREQSYNC. Given a prompt sequence, FREQSYNC generates coherent long videos with smooth transitions, stable motion, and prompt-specific details.

abrupt transitions. Training long-horizon generators from scratch requires substantial computation and large-scale multi-event video-text data [6, 50, 26, 44, 35]. This motivates training-free adaptation of pre-trained short-video generators to coherent multi-prompt long video generation.

Existing training-free long-video methods improve temporal extension through noise rescheduling, sliding-window attention, temporal co-denoising, distributed inference, or frequency-domain blending [45, 36, 24, 42, 30, 27]. For multi-prompt generation, DiTCtrl introduces cross-segment KV-sharing and latent blending, showing that adjacent segments must be coupled for smooth prompt transitions [7]. However, our diagnostic observation suggests that coarse or frequency-agnostic coupling is insufficient. Weak coupling causes flicker, background jumps, and discontinuous motion, while overly strong coupling may over-preserve source-side layouts and semantics. As shown in Figure 2, such source retention can produce ghosting or duplicated subjects near boundaries.

We summarize this issue as a frequency-agnostic boundary coupling problem. A prompt boundary should not be handled by full-frequency source reuse or coarse overlap blending alone. Instead, source-side low- and mid-frequency components should provide layout, illumination, and motion continuity, while target-side high-frequency components should preserve newly introduced prompt details. This decomposition should guide both attention-level inheritance and prediction-level stitching.

To this end, we propose FREQSYNC, a training-free framework for multi-prompt long video generation. FREQSYNC formulates prompt transitions as frequency-selective boundary coupling and introduces two complementary modules. Frequency-Synchronized Source-Conditioned Attention (FS-SCA) performs frequency-selective source conditioning inside selected attention layers, enabling the target segment to inherit boundary-relevant structure without copying full-frequency source features. Residual-Clipped Spectral Stitching (RCSS) operates on denoised overlap predictions, performing spectral seam alignment and conservative low-frequency continuation under residual constraints. Together, FS-SCA and RCSS improve cross-segment continuity while reducing semantic contamination and stitching artifacts. Figure 1 shows representative generated results.

Our contributions are summarized as follows:

- We identify a frequency-agnostic boundary coupling problem in training-free multi-prompt long video generation. Weak coupling leads to discontinuous transitions, whereas overly strong or coarse coupling may over-preserve outdated source semantics, duplicate subjects, and introduce boundary artifacts.
- We propose FREQSYNC, a training-free frequency-synchronized framework for prompt-boundary coupling. It integrates Frequency-Synchronized Source-Conditioned Attention (FS-SCA) for frequency-selective attention-level source conditioning and Residual-Clipped Spectral Stitching (RCSS) for residual-constrained prediction-level boundary stitching.
- Extensive experiments on a 130-case multi-prompt benchmark demonstrate that FREQSYNC achieves the strongest overall performance, improving transition smoothness, temporal consistency, prompt alignment, and perceptual video quality over representative multi-prompt and long-video generation baselines.

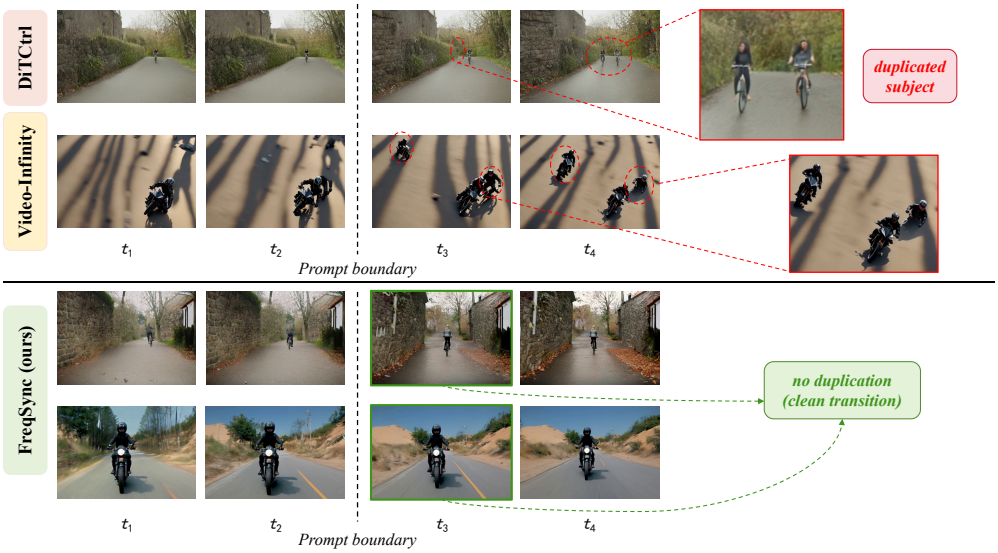


Figure 2: Diagnostic observation. Strong frequency-agnostic coupling may over-preserve source-side evidence and cause duplicated subjects. FREQSYNC transfers mainly low- and mid-frequency continuity cues while keeping target details controlled.

2 Related Work

Text-to-video diffusion models. Diffusion models have become the dominant paradigm for visual generation [18, 40, 38]. Early and representative video generators extend diffusion modeling to the temporal domain through latent video diffusion, motion modules, cascaded generation, or space-time architectures [20, 39, 19, 5, 54, 46, 14, 10, 4, 47, 3, 11, 13, 25]. More recently, diffusion transformers and MM-DiT architectures have shown stronger scalability in image and video generation [34, 12, 31, 50, 53, 26, 44, 35]. Our work builds on pre-trained video diffusion generators and improves multi-prompt temporal coherence without updating model parameters.

Long and multi-prompt video generation. Long video generation requires maintaining visual fidelity, temporal consistency, and semantic progression over extended durations. Training-based methods introduce dedicated long-horizon architectures, datasets, or autoregressive generation strategies [43, 16, 49, 51, 21]. In contrast, training-free methods reuse short-video generators through temporal co-denoising, noise rescheduling, sliding windows, queue-based generation, frequency-domain fusion, position mapping, or informative frame selection [45, 36, 24, 30, 27]. Multi-prompt generation further requires prompt-specific semantic activation and smooth transitions across segment boundaries. Gen-L-Video, VideoDirectorGPT, VideoStudio, TALC, MEVG, Video-Infinity, and DiTCtrl explore multi-text, multi-scene, or multi-event video generation under different assumptions [45, 28, 29, 2, 33, 42, 7]. FREQSYNC focuses on training-free prompt-boundary coupling for coherent multi-prompt long videos.

Attention control and frequency-domain generation. Attention control enables training-free manipulation of diffusion generation by modifying attention behavior rather than model parameters [17, 8, 9, 23]. Recent studies also analyze or accelerate attention in video diffusion transformers, showing the importance of temporal correspondence and sparse spatiotemporal attention [32, 52, 41]. In parallel, frequency-domain techniques have been used to improve initialization, temporal consistency, and long-video fidelity [48, 30]. Inspired by these works, FREQSYNC treats prompt boundaries as a frequency-sensitive coupling problem and jointly improves attention-level inheritance and prediction-level stitching.

Difference from DiTCtrl and frequency-domain long-video methods. FREQSYNC is distinct from both DiTCtrl [7] and existing frequency-domain long-video methods. DiTCtrl [7] improves multi-prompt transitions through cross-segment K/V sharing and latent blending, but such source reuse is largely frequency-agnostic and can over-preserve outdated layouts or semantics near prompt

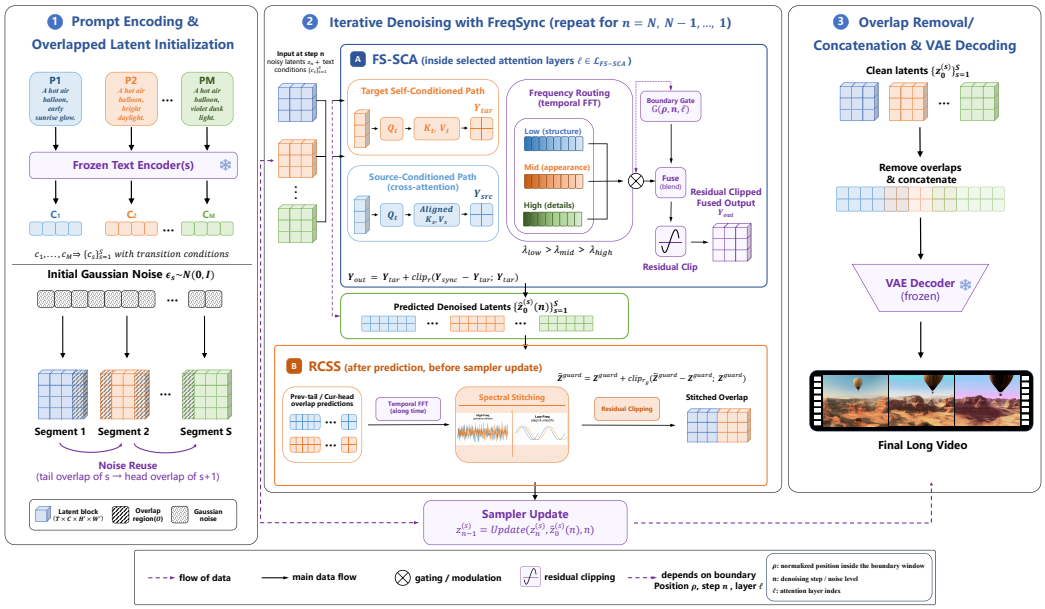


Figure 3: Overview of FREQSYNC. Given a prompt sequence, FREQSYNC first encodes frozen text conditions and initializes overlapped latent segments with noise reuse. During iterative denoising, FS-SCA performs frequency-synchronized source-conditioned attention inside selected attention layers, while RCSS applies residual-clipped spectral stitching to denoised overlap predictions before the sampler update. After denoising, overlapped regions are removed and the clean latent segments are concatenated for VAE decoding, producing the final long video.

boundaries. In contrast, FREQSYNC keeps the target segment as the semantic anchor and introduces source information only through boundary-aligned frequency-selective routing: low- and mid-frequency components carry structure and motion continuity, while high-frequency details remain target-dominant. Frequency-domain long-video methods such as FreeLong [30] mainly enhance long-range temporal consistency through spectral feature blending, but they do not explicitly address the multi-prompt boundary trade-off between source continuity and target-prompt activation. FREQSYNC resolves this trade-off jointly at the attention level through FS-SCA and at the prediction level through RCSS, with residual clipping to prevent excessive boundary corrections.

3 Method

We study training-free multi-prompt long video generation with a pre-trained video diffusion generator. Given an ordered prompt sequence, FREQSYNC synthesizes a coherent long video through overlapped latent segments, as shown in Figure 3. It first encodes prompt conditions and initializes neighboring segments with overlap noise reuse. During denoising, FS-SCA operates inside selected attention layers, while RCSS stitches denoised overlap predictions before the sampler update. After denoising, overlapped regions are removed, the cleaned latent segments are concatenated, and the frozen VAE decoder produces the final long video.

3.1 Problem Formulation

Let $\mathcal{P} = \{p_i\}_{i=1}^{M}$ denote an ordered sequence of prompts. We construct the long video by decomposing generation into overlapped latent segments. Let T denote the latent length of each segment, O the overlap size between neighboring segments, B the number of transition segments inserted between adjacent prompts, and L the number of additional dwell segments assigned to middle prompts. The total number of segments and the resulting long-video length are:

$$S = M + B(M-1) + L(M-2), \qquad T_{\text{long}} = T + (S-1)(T-O). \tag{1}$$

The first segment contributes T latent frames, and each subsequent segment contributes $T - O$ new frames due to overlap reuse.

To initialize neighboring segments consistently, we reuse the overlap noise:

$$\boldsymbol{\epsilon}_{s+1}[1{:}O] = \boldsymbol{\epsilon}_s[T - O + 1{:}T], \tag{2}$$

where $\boldsymbol{\epsilon}_s$ denotes the initial latent noise of the s-th segment. For each adjacent prompt pair (p_i, p_{i+1}), we insert transition segments. The cross-attention condition for the b-th transition segment is interpolated by a cosine schedule:

$$\omega_{i,b} = \frac{1 - \cos\left(\pi \frac{b+1}{B+1}\right)}{2}, \qquad \mathbf{c}_{i,b}^{\text{cross}} = (1 - \omega_{i,b})\mathbf{c}_i^{\text{cross}} + \omega_{i,b}\mathbf{c}_{i+1}^{\text{cross}}. \tag{3}$$

Following the practical design of multi-prompt sampling, non-cross-attention branches are inherited from the dominant prompt rather than fully interpolated, which avoids off-manifold mixed conditions.

The key question is how neighboring segments should be coupled at a prompt boundary. Isolated generation causes discontinuities, while full source sharing or coarse blending may preserve outdated semantics. FREQSYNC resolves this tension in the temporal frequency domain: low- and mid-frequency source components carry structure and motion continuity, whereas high-frequency target components preserve prompt-specific details.

3.2 Frequency-Synchronized Source-Conditioned Attention (FS-SCA)

FS-SCA addresses the attention-level source-conditioning part of boundary coupling. Consider two neighboring segments, where the source segment corresponds to the previous prompt and the target segment corresponds to the current prompt. In an attention layer of the pre-trained video diffusion generator, let $(\mathbf{Q}_t, \mathbf{K}_t, \mathbf{V}_t)$ denote the query, key, and value features of the target segment, and let $(\mathbf{K}_s, \mathbf{V}_s)$ denote the key and value features of the source segment. Rather than directly replacing target features with source features, FREQSYNC computes two attention paths:

$$\mathbf{Y}^{\text{tar}} = \text{Attn}(\mathbf{Q}_t, \mathbf{K}_t, \mathbf{V}_t), \qquad \mathbf{Y}^{\text{src}} = \text{Attn}(\mathbf{Q}_t, \mathbf{K}_s^{\rightarrow}, \mathbf{V}_s^{\rightarrow}). \tag{4}$$

The first path is the target self-conditioned path, which preserves prompt-specific semantics. The second path is the source-conditioned path, where $\mathbf{K}_s^{\rightarrow}$ and $\mathbf{V}_s^{\rightarrow}$ are source features aligned to the target boundary. Specifically, the tail overlap of the source segment is aligned to the head overlap of the target segment, so that the same global boundary frames are compared and fused. For the first few non-overlap guard frames, the final source-overlap representation is used as a structural reference. This tail-to-head alignment is important because local frame indices in neighboring segments do not correspond to the same global temporal positions.

We reshape the video feature part of $\mathbf{Y}^{\text{tar}}$ and $\mathbf{Y}^{\text{src}}$ into $T \times N \times d$ tensors, where T is the compressed temporal length and N is the number of spatial tokens. We then decompose each output into temporal low-, mid-, and high-frequency bands:

$$\mathbf{Y}^b = \mathcal{F}^{-1}(\mathbf{M}_b \odot \mathcal{F}(\mathbf{Y})), \qquad b \in \{\text{low}, \text{mid}, \text{high}\}, \tag{5}$$

where $\mathcal{F}$ denotes the temporal Fourier transform, $\mathbf{M}_b$ denotes a band mask, and $\odot$ denotes element-wise multiplication. This decomposition reflects the different roles of temporal frequency bands in boundary coupling: low-frequency components encode global layout, background, illumination, and coarse motion; mid-frequency components encode intermediate structure and local appearance; high-frequency components encode fine details and prompt-specific textures.

The fused video attention output is defined as:

$$\mathbf{Y}^{\text{sync}} = \mathbf{Y}^{\text{tar}} + G(\rho, n, \ell) \sum_{b \in \{\text{low},\text{mid},\text{high}\}} \lambda_b \left(\mathbf{Y}^{\text{src},b} - \mathbf{Y}^{\text{tar},b}\right). \tag{6}$$

Here, $G(\rho, n, \ell)$ controls where and when source-conditioned routing is active, while λ_b controls how strongly each temporal frequency band inherits source-side continuity cues.

Accordingly, we choose the routing strengths to satisfy $\lambda_{\text{low}} > \lambda_{\text{mid}} > \lambda_{\text{high}}$. In practice, λ_{low} is close to one near the boundary, λ_{mid} is moderate, and λ_{high} remains small. This design gives the target segment more source information where continuity matters, while preventing source high-frequency details from overwriting target prompt semantics.

To make the routing boundary-aware, we further modulate the routing strengths with a temporal boundary schedule. Let $\rho \in [0, 1]$ denote the normalized temporal position inside the boundary window, and let $g(n, \ell)$ denote a smooth denoising-step and layer gate. The boundary gate is:

$$G(\rho, n, \ell) = g(n, \ell) \left[\cos\left(\frac{\pi\rho}{2}\right)\right]^{2\gamma}, \tag{7}$$

where γ controls the decay from the source side to the target side. As a result, the gate is strongest near the beginning of the target segment and gradually decays as the target prompt takes over.

Residual clipping for stable source conditioning. While source-conditioned attention is essential for continuity, unconstrained correction may destabilize generation or over-constrain the target prompt. We therefore clip the frequency-fused correction relative to the target self-attention output. Let $\Delta = \mathbf{Y}^{\mathrm{sync}} - \mathbf{Y}^{\mathrm{tar}}$. The final output is:

$$\widehat{\mathbf{Y}} = \mathbf{Y}^{\mathrm{tar}} + \mathrm{clip}_r\big(\Delta; \mathbf{Y}^{\mathrm{tar}}\big), \tag{8}$$

where clip_r limits the residual's frame-wise energy to a fixed ratio of the target output, preserving source-conditioned cues without excessive attention-output shifts.

3.3 Residual-Clipped Spectral Stitching (RCSS)

RCSS addresses the prediction-level stitching part of boundary coupling. Frequency-Synchronized Source-Conditioned Attention improves semantic and structural handover inside the denoising network, but the final long video also depends on how neighboring denoised segment predictions are stitched. Directly modifying the noisy solver state can push samples away from the model distribution and may cause severe artifacts. Coarse overlap blending, on the other hand, may suppress target details or propagate source-side residue. Therefore, FREQSYNC operates only on denoised overlap predictions and applies frequency-aware fusion with residual clipping to all post-overlap corrections.

Let $\mathbf{Z}^{\mathrm{prev}} \in \mathbb{R}^{O\times C\times H\times W}$ denote the tail overlap of the previous segment, and let $\mathbf{Z}^{\mathrm{cur}} \in \mathbb{R}^{O\times C\times H\times W}$ denote the head overlap of the current segment. We decompose each overlap into temporal bands as in Eq. (5). The fused overlap is:

$$\begin{aligned}\mathbf{Z}^{\mathrm{fuse}} = {} & \alpha_{\mathrm{low}}\mathbf{Z}^{\mathrm{prev,low}} + (1-\alpha_{\mathrm{low}})\mathbf{Z}^{\mathrm{cur,low}} \\ & + \alpha_{\mathrm{high}}\mathbf{Z}^{\mathrm{prev,high}} + (1-\alpha_{\mathrm{high}})\mathbf{Z}^{\mathrm{cur,high}},\end{aligned} \tag{9}$$

where α_{low} follows a raised-cosine transition and α_{high} decays faster to avoid ghosting. We additionally apply a light low-frequency temporal smoothing pass inside the overlap to reduce boundary-local flicker while preserving high-frequency target details.

To handle the non-overlap guard frames immediately after the boundary, we propagate the low-frequency trend from the fused overlap. Let $\mathbf{Z}^{\mathrm{guard}}$ denote the first G non-overlap frames of the target segment. We compute a low-frequency memory $\mathbf{M}^{\mathrm{low}}$ from $\mathbf{Z}^{\mathrm{fuse}}$ and form a proposal:

$$\widetilde{\mathbf{Z}}^{\mathrm{guard}} = \mathbf{Z}^{\mathrm{guard,high}} + (1-\eta)\mathbf{Z}^{\mathrm{guard,low}} + \eta\mathbf{M}^{\mathrm{low}}, \tag{10}$$

where η is a decaying guard strength. The final guard correction is residual-clipped:

$$\widehat{\mathbf{Z}}^{\mathrm{guard}} = \mathbf{Z}^{\mathrm{guard}} + \mathrm{clip}_{r_g}\Big(\widetilde{\mathbf{Z}}^{\mathrm{guard}} - \mathbf{Z}^{\mathrm{guard}}; \mathbf{Z}^{\mathrm{guard}}\Big). \tag{11}$$

This design is intentionally conservative: it synchronizes low-frequency continuation near boundaries, but avoids direct modification of the solver state and limits the magnitude of each correction.

3.4 Inference Procedure

During inference, FREQSYNC expands the prompt sequence into overlapped latent segments with transition conditions and performs iterative denoising over neighboring segment pairs. The complete inference procedure is summarized in Appendix A.1. In scheduled layers and timesteps, FS-SCA evaluates the target segment through both target self-conditioned and source-conditioned attention paths, then applies frequency-selective routing with residual clipping so that low- and mid-frequency continuity cues are inherited while high-frequency target details remain dominant. After each denoised prediction is obtained, RCSS fuses the boundary overlap and propagates conservative low-frequency continuation to boundary-adjacent guard frames. All corrections are applied to attention outputs or denoised predictions rather than directly to the noisy solver state, keeping the target branch as the semantic anchor while using the source segment only as a boundary-continuity reference.

Table 1: Quantitative comparison. Results are averaged over three seeds on the 130-case multi-prompt benchmark. FREQSYNC delivers the strongest overall performance, achieving the best scores on CSCV, Text-Image Similarity, T2V Alignment, and Physical Consistency, while remaining highly competitive on Motion Smoothness and Visual Quality with only marginal gaps to the best results. Best scores are in **bold**; second-best scores are underlined.

Method	CSCV (%)↑	Motion Smooth. (%)↑	Text-Image Similarity↑	Visual Quality↑	T2V Align.↑	Physical Consist.↑
MEVG [33]	67.22	95.66	29.56	1.970	2.773	2.280
Video-Infinity [42]	83.62	96.45	29.72	**3.247**	3.422	3.210
DiTCtrl [7]	86.11	**98.44**	30.33	3.147	3.758	3.228
FREQSYNC (Ours)	**86.43**	98.25	**30.53**	3.214	**3.849**	**3.240**

4 Experiments

4.1 Experimental Setup

Implementation details. We implement FREQSYNC on the pre-trained CogVideoX-2B backbone [50] in a fully training-free manner. Videos are generated at 480×720 resolution and 16 FPS. We use latent segments of 13 frames, an overlap size of 9, and 2 transition blocks between adjacent prompts. FS-SCA is applied to denoising steps $[2, 31)$ and transformer layers $[22, 30)$. Its temporal Fourier bands are split by normalized low- and mid-frequency cutoff ratios of 0.30 and 0.52, which define the low/mid/high bands used for frequency routing. Residual clipping is used for both FS-SCA attention fusion and RCSS guard correction. All experiments are performed on NVIDIA A800 GPUs.

Benchmark, baselines, and metrics. Following the MPVBench setting [7], we construct a 130-case multi-prompt benchmark covering ten transition modes: background, subject, camera, style, lighting, location, speed, emotion, clothing, and action. Each method is evaluated over three random seeds. We compare with MEVG [33], Video-Infinity [42], and DiTCtrl [7]; qualitative comparisons additionally include a commercial model, Kling-2.6 [1]. We report CSCV for transition smoothness, VBench Motion Smoothness [22], CLIP-based Text-Image Similarity [37], and VideoScore2 dimensions including Visual Quality, T2V Alignment, and Physical Consistency [15].

4.2 Quantitative Results

Table 1 reports the main quantitative comparison across transition smoothness, temporal motion quality, prompt alignment, and perceptual video quality. FREQSYNC achieves the strongest overall performance, ranking first on four out of six metrics and second on the remaining two with only marginal gaps to the best scores. Compared with MEVG and Video-Infinity [33, 42], our method substantially improves CSCV and motion-related quality, indicating more stable cross-segment evolution. Compared with DiTCtrl [7], FREQSYNC further improves CSCV from 86.11% to 86.43%, while maintaining a very close Motion Smoothness score of 98.25% versus 98.44%. More importantly, FREQSYNC achieves the highest Text-Image Similarity and T2V Alignment scores, suggesting that frequency-selective boundary coupling improves target-prompt activation without over-preserving outdated source semantics. Although Video-Infinity obtains the highest Visual Quality score, FREQSYNC remains close and achieves stronger transition smoothness, text alignment, and physical consistency. These results demonstrate that FREQSYNC provides the best overall boundary-coupling trade-off. Further per-seed results, significance tests, cross-backbone evaluations, and efficiency analyses are provided in Appendices D, E, F, and I.

4.3 Qualitative Results

Figure 4 presents a representative qualitative comparison on a challenging three-prompt sequence, where a dark blue sports car should move from a sunlit alpine road to a foggy highland road and finally to a night road with city glow, while preserving its identity and motion trajectory. MEVG [33] shows severe cross-segment discontinuities: the camera viewpoint, road layout, and vehicle scale change abruptly, and repeated or inconsistent subjects appear. Video-Infinity [42] produces weak



Figure 4: Qualitative comparison. On a three-prompt driving sequence, baselines suffer from viewpoint drift, weak prompt transition, ghosting, or duplicated traces. FREQSYNC preserves vehicle identity and motion while producing smoother semantic transitions.

dynamics and unstable visual details, with nearly static motion, flickering backgrounds, and ghosting artifacts. DiTCtrl [7] better maintains vehicle identity and global continuity, but its full-frequency source inheritance still introduces boundary artifacts during the foggy-to-night transition, including ghosting, duplicated visual traces, and an abrupt semantic switch. Kling-2.6 [1] generates realistic frames but shows weaker prompt following: the foggy road and final city-glow semantics are insufficiently expressed. In contrast, FREQSYNC preserves a stable car identity, viewpoint, and forward motion while progressively changing the background; the fog and night illumination emerge smoothly without duplicated subjects or strong ghosting. This comparison supports our frequency-synchronized boundary coupling design: low- and mid-frequency synchronization maintains structure and motion, while target-dominant high-frequency components preserve prompt-specific details. More qualitative results and the single-prompt long-video application are provided in Appendix K.

4.4 Diagnostic Analysis of Boundary Coupling

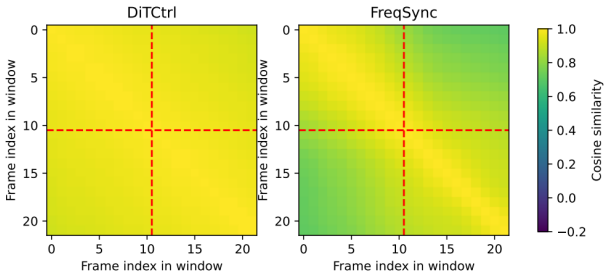


Figure 5: Temporal self-similarity around prompt boundaries. FREQSYNC preserves local diagonal continuity while reducing excessive cross-boundary similarity, supporting frequency-selective boundary coupling.

We further analyze the frequency-agnostic boundary coupling problem identified in Sec. 1. Figure 5 visualizes temporal self-similarity matrices around a prompt boundary. DiTCtrl [7] exhibits uniformly high cross-boundary similarity, indicating strong source-segment retention. While this helps smooth transitions, it may also preserve outdated source layouts and correlate with duplicated-subject artifacts, as observed in Figure 2. In contrast, FREQSYNC maintains strong local temporal continuity along the diagonal while reducing excessive cross-boundary similarity, suggesting a better balance between source continuity and target-prompt activation. This analysis supports our motivation: smoothness alone is not sufficient for multi-prompt generation. A successful method should selectively transfer continuity cues, preserve emerging target details, and stitch adjacent denoised predictions without high-energy boundary corrections. FREQSYNC enforces this balance through FS-SCA and RCSS,

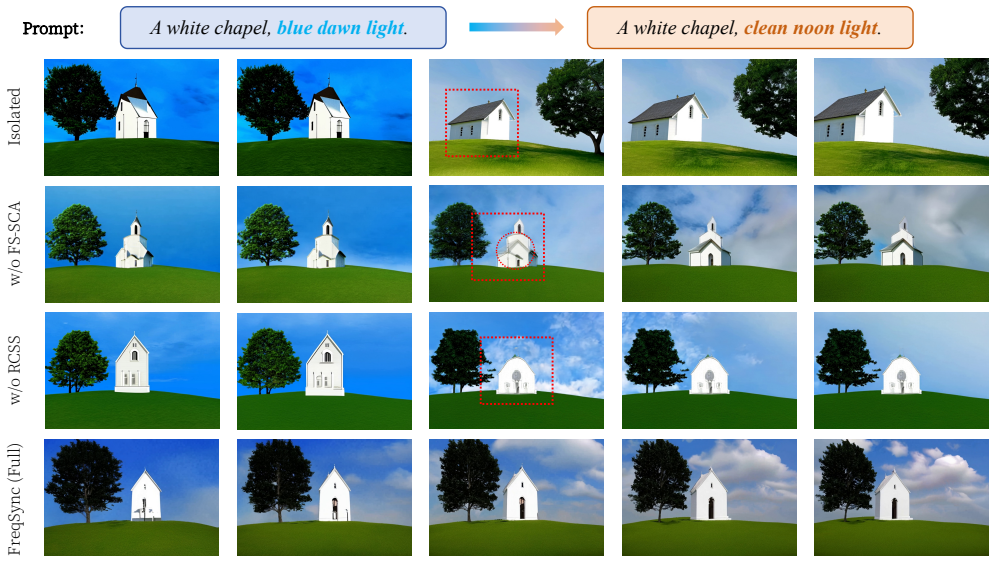


Figure 6: Visual ablation of FREQSYNC. Isolated generation produces a hard cut. RCSS alone smooths the overlap region but lacks attention-level semantic handover. FS-SCA alone preserves the global layout but changes the chapel structure. Full FREQSYNC combines both modules for smooth transitions and stable identity.

Table 2: Ablation study on a 40-case subset. Ablations are conducted on a representative 40-case subset covering all transition modes under a single random seed. CSCV is the primary boundary-transition metric. Segment-level Text-Image Similarity is reported as a prompt-fidelity reference and is not boundary-aware. Best results are in **bold** and second-best results are underlined.

Variant	FS-SCA	RCSS	CSCV (%)↑	Text-Image Sim.↑
Isolated	×	×	76.96	**30.89**
w/o FS-SCA (RCSS only)	×	✓	<u>85.73</u>	30.62
w/o RCSS (FS-SCA only)	✓	×	82.08	30.56
Full FREQSYNC	✓	✓	**89.14**	<u>30.71</u>

which jointly improve attention-level source conditioning and prediction-level boundary stitching. Further temporal-frequency analysis is provided in Appendix J.

4.5 Ablation Studies

We ablate the two core components of FREQSYNC: FS-SCA and RCSS. Ablation results are evaluated on a 40-case subset under a single random seed. As shown in Table 2 and Figure 6, isolated generation produces abrupt prompt-boundary changes and obtains the lowest CSCV score. Using RCSS alone improves CSCV from 76.96% to 85.73%, showing that prediction-level spectral stitching can smooth the overlap region, but the lack of FS-SCA still causes slight changes in object appearance and structural handover. Using FS-SCA alone improves CSCV to 82.08%, indicating that attention-level source conditioning helps preserve layout and transition continuity, although the chapel structure still changes without prediction-level stitching. The full FREQSYNC achieves the best CSCV score of 89.14% and produces the most coherent transition: the global layout, chapel position, and object appearance remain stable while the illumination follows the target prompt. Segment-level Text-Image Similarity serves as a prompt-fidelity reference; although isolated generation obtains a slightly higher score because it is not boundary-aware, full FREQSYNC achieves the best prompt alignment among coupled variants with much smoother transitions. Additional frequency-routing ablations and hyperparameter sensitivity analyses are provided in Appendices G and H, respectively.

4.6 Human Evaluation

To complement automatic metrics, we conduct a blind human evaluation with 30 participants on 40 anonymized cases covering all ten transition modes and three random seeds. Four methods are

Table 3: Blind human evaluation on 40 anonymized cases with 30 participants. Mean five-point scores are reported, where higher is better in every column. For source leakage and artifacts, higher scores mean fewer artifacts. Best scores are in **bold**.

Method	Target semantic correctness↑	Source leakage and artifacts↑	Transition smoothness↑	Visual quality↑	Overall preference↑
MEVG [33]	1.67	1.53	1.70	1.83	1.63
Video-Infinity [42]	2.70	2.57	2.83	2.97	2.70
DiTCtrl [7]	3.33	3.07	2.80	2.90	3.17
FREQSYNC (Ours)	**4.30**	**4.23**	**4.27**	**4.07**	**4.30**

evaluated on a five-point scale across five dimensions: target semantic correctness, source leakage and artifacts, transition smoothness, visual quality, and overall preference. Higher scores indicate better performance, with higher scores for source leakage and artifacts reflecting fewer unwanted artifacts. As shown in Table 3, FREQSYNC achieves the highest mean scores across all five dimensions, with particularly strong improvements in target semantic correctness, artifact suppression, and transition smoothness. These results further demonstrate that FREQSYNC effectively improves temporal continuity while preserving newly introduced prompt semantics, consistent with the quantitative and qualitative evaluations. Additional protocol details are provided in Appendix C.4.

5 Conclusion

We present FREQSYNC, a training-free framework for multi-prompt long video generation based on frequency-synchronized boundary coupling. Starting from the observation that weak boundary coupling causes abrupt transitions whereas overly strong or frequency-agnostic coupling may preserve outdated source semantics and introduce ghosting or duplicated subjects, FREQSYNC decomposes prompt transitions into attention-level source conditioning and prediction-level boundary stitching. FS-SCA transfers low- and mid-frequency continuity cues while preserving target-dominant details, and RCSS aligns denoised overlap predictions under residual constraints for stable stitching. Extensive quantitative, qualitative, diagnostic, and ablation results demonstrate that FREQSYNC improves transition smoothness, prompt alignment, and overall video quality without additional training, providing an effective solution for coherent multi-prompt long video generation.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (No. 62576109), the Scientific and Technological Innovation Action Plan of the Shanghai Science and Technology Committee (No. 25511104402), and the AI for Science Program of the Shanghai Municipal Commission of Economy and Informatization (No. 2025-GZL-RGZN-BTBX-02006). The authors declare no competing interests.

References

[1] Kling. https://klingai.kuaishou.com/, 2024.

[2] Hritik Bansal, Yonatan Bitton, Michal Yarom, Idan Szpektor, Aditya Grover, and Kai-Wei Chang. Talc: Time-aligned captions for multi-scene text-to-video generation. *arXiv preprint arXiv:2405.04682*, 2024.

[3] 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 Asia 2024 Conference Papers*, pages 1–11, 2024.

[4] 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.

[5] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22563–22575, 2023.

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

[7] Minghong Cai, Xiaodong Cun, Xiaoyu Li, Wenze Liu, Zhaoyang Zhang, Yong Zhang, Ying Shan, and Xiangyu Yue. Ditctrl: Exploring attention control in multi-modal diffusion transformer for tuning-free multi-prompt longer video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7763–7772, 2025.

[8] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22560–22570, 2023.

[9] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM transactions on Graphics (TOG)*, 42(4):1–10, 2023.

[10] 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.

[11] Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7310–7320, 2024.

[12] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[13] 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 *European Conference on Computer Vision*, pages 205–224. Springer, 2024.

[14] 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.

[15] Xuan He, Dongfu Jiang, Ping Nie, Minghao Liu, Zhengxuan Jiang, Mingyi Su, Wentao Ma, Junru Lin, Chun Ye, Yi Lu, et al. Videoscore2: Think before you score in generative video evaluation. *arXiv preprint arXiv:2509.22799*, 2025.

[16] Roberto Henschel, Levon Khachatryan, Hayk Poghosyan, Daniil Hayrapetyan, Vahram Tadevosyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Streamingt2v: Consistent, dynamic, and extendable long video generation from text. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2568–2577, 2025.

[17] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[19] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

[20] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35: 8633–8646, 2022.

[21] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025.

[22] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024.

[23] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15954–15964, 2023.

[24] Jihwan Kim, Junoh Kang, Jinyoung Choi, and Bohyung Han. Fifo-diffusion: Generating infinite videos from text without training. *Advances in Neural Information Processing Systems*, 37: 89834–89868, 2024.

[25] Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, et al. Videopoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*, 2023.

[26] 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.

[27] Zhuoling Li, Hossein Rahmani, Qiuhong Ke, and Jun Liu. Longdiff: Training-free long video generation in one go. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17789–17798, 2025.

[28] Han Lin, Abhay Zala, Jaemin Cho, and Mohit Bansal. Videodirectorgpt: Consistent multi-scene video generation via llm-guided planning. *arXiv preprint arXiv:2309.15091*, 2023.

[29] Fuchen Long, Zhaofan Qiu, Ting Yao, and Tao Mei. Videostudio: Generating consistent-content and multi-scene videos. In *European Conference on Computer Vision*, pages 468–485. Springer, 2024.

[30] Yu Lu, Yuanzhi Liang, Linchao Zhu, and Yi Yang. Freelong: Training-free long video generation with spectralblend temporal attention. *Advances in Neural Information Processing Systems*, 37: 131434–131455, 2024.

[31] Xin Ma, Yaohui Wang, Xinyuan Chen, Gengyun Jia, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024.

[32] Jisu Nam, Soowon Son, Dahyun Chung, Jiyoung Kim, Siyoon Jin, Junhwa Hur, and Seungryong Kim. Emergent temporal correspondences from video diffusion transformers. *arXiv preprint arXiv:2506.17220*, 2025.

[33] 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 *European Conference on computer vision*, pages 401–418. Springer, 2024.

[34] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[35] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

[36] Haonan Qiu, Menghan Xia, Yong Zhang, Yingqing He, Xintao Wang, Ying Shan, and Ziwei Liu. Freenoise: Tuning-free longer video diffusion via noise rescheduling. *arXiv preprint arXiv:2310.15169*, 2023.

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[39] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

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

[41] Wenhao Sun, Rong-Cheng Tu, Yifu Ding, Zhao Jin, Jingyi Liao, Shunyu Liu, and Dacheng Tao. Vorta: Efficient video diffusion via routing sparse attention. *arXiv preprint arXiv:2505.18809*, 2025.

[42] Zhenxiong Tan, Xingyi Yang, Songhua Liu, and Xinchao Wang. Video-infinity: Distributed long video generation. *arXiv preprint arXiv:2406.16260*, 2024.

[43] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. *arXiv preprint arXiv:2210.02399*, 2022.

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

[45] 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.

[46] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023.

[47] Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, et al. Lavie: High-quality video generation with cascaded latent diffusion models. *International Journal of Computer Vision*, 133(5):3059–3078, 2025.

[48] 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*, pages 378–394. Springer, 2024.

[49] 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. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3184–3194, 2025.

[50] 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.

[51] Lvmin Zhang, Shengqu Cai, Muyang Li, Gordon Wetzstein, and Maneesh Agrawala. Frame context packing and drift prevention in next-frame-prediction video diffusion models. *arXiv preprint arXiv:2504.12626*, 2025.

[52] Peiyuan Zhang, Yongqi Chen, Haofeng Huang, Will Lin, Zhengzhong Liu, Ion Stoica, Eric Xing, and Hao Zhang. Vsa: Faster video diffusion with trainable sparse attention. *arXiv preprint arXiv:2505.13389*, 2025.

[53] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.

[54] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022.

Appendix

- In Appendix A, we provide additional algorithmic and implementation details of FREQSYNC, including the inference pseudocode, the placement of RCSS before the sampler update, the supplementary code organization, and the configuration hyperparameters.
- In Appendix B, we provide a stability analysis of the residual clipping mechanism used in FS-SCA and RCSS, showing that each introduced correction is locally bounded relative to a reference representation.
- In Appendix C, we provide additional evaluation details, including case-level aggregation, prompt-level text alignment, and the random seed setting used for quantitative evaluation.
- In Appendix D, we provide additional quantitative results, including per-seed results for all compared methods under three random seeds.
- In Appendix E, we report paired significance tests on 100 balanced cases, including bootstrap confidence intervals and Holm-adjusted tests.
- In Appendix F, we evaluate Wan2.1 and VideoCrafter2 to examine cross-architecture transfer and its limitations.
- In Appendix G, we compare low/mid/high source-routing controls, including reverse and uniform routing.
- In Appendix H, we report OFAT and cutoff–layer interaction tests.
- In Appendix I, we report sampling time, GPU memory, and counted FLOPs.
- In Appendix J, we provide additional diagnostic visualizations that analyze temporal-frequency energy around prompt boundaries.
- In Appendix K, we provide additional qualitative visualizations, including more generated examples of FREQSYNC, a single-prompt long-video generation application, and additional qualitative comparisons with Kling-2.6, MEVG, Video-Infinity, and DiTCtrl [1, 33, 42, 7].
- In Appendix L, we discuss the broader impacts of FREQSYNC, including potential positive applications and possible misuse risks of coherent long-video generation.
- In Appendix M, we summarize the existing models, baselines, benchmarks, and evaluation tools used in this work, together with their corresponding licenses or usage terms.
- In Appendix N, we discuss the limitations of FREQSYNC, including dependence on the underlying pre-trained generator, challenges under large semantic changes, additional inference overhead, and inherited model risks.

A Algorithmic and Implementation Details

A.1 Inference Pseudocode

Algorithm 1 summarizes the inference procedure of FREQSYNC. Unlike the method overview in the main paper, this pseudocode clarifies the order in which prompt expansion, overlapped latent initialization, FS-SCA, RCSS, sampler update, and final latent concatenation are applied.

A.2 Why RCSS Is Applied Before the Sampler Update

RCSS is applied to denoised overlap predictions rather than directly to the noisy solver state. A diffusion sampler updates the noisy latent according to the current denoised prediction and the noise schedule. Directly modifying the noisy state may introduce uncontrolled perturbations that are further propagated by later denoising steps. In contrast, modifying denoised predictions allows boundary corrections to enter the update through the original sampler rule. Therefore, RCSS refines the model prediction at the boundary while keeping the solver dynamics unchanged. This design makes the boundary correction more conservative and reduces the risk of off-manifold latent updates.

Algorithm 1 FREQSYNC Inference

Require: Prompt sequence $\mathcal{P} = \{p_i\}_{i=1}^{M}$, frozen video diffusion generator ϵ_θ, segment length T, overlap length O, transition blocks B, dwell segments L

Compute S and T_{long} by Eq. (1).
Build transition conditions $\mathbf{c}_{i,b}^{\text{cross}}$ by Eq. (3).
Initialize latent segments $\{\mathbf{x}_s^{(N)}\}_{s=1}^{S}$ with overlap noise reuse by Eq. (2).
for each denoising step $n = N, \ldots, 1$ **do**
  for each neighboring segment pair $(s, s+1)$ **do**
    Run ϵ_θ on the neighboring segments.
    if (n, ℓ) is scheduled for FS-SCA **then**
      Compute $\mathbf{Y}^{\text{tar}}$ and $\mathbf{Y}^{\text{src}}$ by Eq. (4) with tail-to-head aligned source K/V.
      Decompose attention outputs into temporal bands by Eq. (5).
      Obtain $\mathbf{Y}^{\text{sync}}$ using the boundary gate in Eq. (7) and the routing rule in Eq. (6).
      Obtain the clipped attention output $\widehat{\mathbf{Y}}$ by Eq. (8).
    end if
    Obtain denoised predictions for the neighboring segments.
    Extract $\mathbf{Z}^{\text{prev}}$ and $\mathbf{Z}^{\text{cur}}$ from the boundary overlap.
    Fuse the denoised overlap $\mathbf{Z}^{\text{fuse}}$ by Eq. (9).
    Apply low-frequency guard continuation by Eq. (10).
    Apply residual-clipped guard correction by Eq. (11).
    Update noisy latent states with the original diffusion sampler.
  end for
end for
Remove duplicated overlaps and concatenate the cleaned latent segments.
Decode the final latent sequence with the frozen VAE decoder.
return Generated multi-prompt long video.

A.3 Supplementary Code and Configuration

We include partial source code, the corresponding configuration file, and representative prompt examples in the supplementary material. The code is organized into three files. `freqsync_modules.py` contains the frequency-domain utilities shared by FS-SCA and RCSS, including temporal band decomposition, source-aligned K/V construction, spectral fusion, and residual clipping. `freqsync_attention.py` contains the attention-layer integration of FS-SCA, where the target segment is processed through both target self-conditioned and source-conditioned attention paths. `freqsync_sampler.py` contains the multi-prompt inference logic, including prompt-sequence expansion, overlap noise reuse, and RCSS before the sampler update. The file `config.yaml` lists the main sampling and frequency-routing hyperparameters used by FREQSYNC. We also include `sample_prompts.json`, which provides representative prompt-sequence examples for each transition mode and illustrates the input format for multi-prompt generation.

A.4 Configuration Hyperparameters

Table 4 summarizes the main hyperparameters included in `config.yaml`. Basic inference parameters determine how an input prompt sequence is decomposed into overlapped latent segments. FS-SCA parameters specify where attention-level source conditioning is applied and how source information is routed across temporal frequency bands. RCSS parameters control prediction-level spectral stitching around the overlap, including low-pass seam alignment, post-overlap guard synchronization, and residual-energy clipping.

The source routing strengths are intentionally larger for low- and mid-frequency bands, which mainly carry layout, trajectory, and coarse motion continuity across prompt boundaries. The high-frequency source strength is kept small so that fine target-prompt details remain dominant. Residual clipping is applied in both FS-SCA and RCSS to prevent overly strong boundary corrections and reduce semantic contamination or stitching artifacts.

Table 4: Main hyperparameters in the supplementary configuration file. The table lists the core parameters included in `config.yaml`. FS-SCA controls frequency-selective attention-level source conditioning, while RCSS controls residual-constrained prediction-level boundary stitching.

Module	Hyperparameter	Value	Description
Basic inference	latent frames per segment	13	Number of latent frames generated for each segment.
Basic inference	sampling FPS	16	Frame rate used when saving the generated video.
Basic inference	overlap size	9	Number of shared latent frames between adjacent segments.
Basic inference	transition blocks	2	Number of transition segments inserted between neighboring prompts.
Basic inference	seed	42	Default seed used by the example configuration.
FS-SCA	active attention layers	$[22, 30)$	Transformer layers where source-conditioned attention is applied.
FS-SCA	active denoising steps	$[2, 31)$	Denoising steps where FS-SCA is enabled.
FS-SCA	low-frequency cutoff	0.30	Temporal cutoff used to extract stable low-frequency structure.
FS-SCA	mid-frequency cutoff	0.52	Temporal cutoff separating mid-frequency motion from high-frequency details.
FS-SCA	source low-frequency strength	0.98	Weight for transferring source-side low-frequency continuity cues.
FS-SCA	source mid-frequency strength	0.68	Weight for transferring source-side mid-frequency motion cues.
FS-SCA	source high-frequency strength	0.035	Small source weight retained in high-frequency bands to keep target details dominant.
FS-SCA	attention-output residual clip	0.32	Maximum relative residual energy allowed for attention-output correction.
RCSS	boundary low-pass ratio	0.23	Low-pass cutoff used for spectral seam alignment in denoised predictions.
RCSS	boundary smoothing strength	0.12	Strength of temporal smoothing applied near the overlap.
RCSS	boundary smoothing steps	32	Number of early/middle denoising steps using boundary smoothing.
RCSS	guard frames	4	Number of post-overlap frames receiving low-frequency continuation.
RCSS	guard strength	0.52	Strength of the post-overlap low-frequency guard.
RCSS	guard residual clip	0.18	Maximum relative residual energy allowed for guard-frame correction.

B Bounded Residual Correction

This section provides a simple stability analysis for the residual clipping used in FS-SCA and RCSS. The purpose is not to prove global convergence of the diffusion sampler, but to show that each correction introduced by FREQSYNC is locally bounded relative to a reference representation.

B.1 RMS-Norm Residual Bound

Let $\mathbf{B}$ denote the base representation, $\widetilde{\mathbf{B}}$ denote the proposed corrected representation, and $\mathbf{R}$ denote the reference representation. Define the residual as

$$\Delta = \widetilde{\mathbf{B}} - \mathbf{B}. \tag{12}$$

The residual-clipped correction used in FREQSYNC can be written as

$$\widehat{\mathbf{B}} = \mathbf{B} + s\Delta, \qquad s = \min\left(1, \frac{r \cdot \mathrm{RMS}(\mathbf{R})}{\mathrm{RMS}(\Delta) + \epsilon}\right), \tag{13}$$

where r is the residual clipping ratio and ϵ is a small constant for numerical stability. The resulting correction satisfies

$$\mathrm{RMS}\left(\widehat{\mathbf{B}} - \mathbf{B}\right) \le r \cdot \mathrm{RMS}(\mathbf{R}). \tag{14}$$

Proof. Since $\widehat{\mathbf{B}} - \mathbf{B} = s\Delta$, we have

$$\mathrm{RMS}\left(\widehat{\mathbf{B}} - \mathbf{B}\right) = s\,\mathrm{RMS}(\Delta). \tag{15}$$

If $s = 1$, then by the definition of s,

$$\mathrm{RMS}(\Delta) \le r\,\mathrm{RMS}(\mathbf{R}). \tag{16}$$

If $s < 1$, then

$$s\,\mathrm{RMS}(\Delta) = \frac{r\,\mathrm{RMS}(\mathbf{R})}{\mathrm{RMS}(\Delta) + \epsilon}\,\mathrm{RMS}(\Delta) \le r\,\mathrm{RMS}(\mathbf{R}). \tag{17}$$

Thus, in both cases,

$$\mathrm{RMS}\left(\widehat{\mathbf{B}} - \mathbf{B}\right) \le r\,\mathrm{RMS}(\mathbf{R}). \tag{18}$$

□

Implication for FS-SCA. For FS-SCA, $\mathbf{B}$ is the target self-conditioned attention output, $\widetilde{\mathbf{B}}$ is the frequency-fused source-conditioned proposal, and $\mathbf{R}$ is the target attention output. Therefore, the source-conditioned attention correction is bounded by a fixed ratio of the target-path activation energy. This prevents source-conditioned attention from fully overriding the target branch.

Implication for RCSS. For RCSS, $\mathbf{B}$ is the original denoised guard prediction, $\widetilde{\mathbf{B}}$ is the low-frequency continuation proposal, and $\mathbf{R}$ is the original denoised guard prediction. Therefore, the guard-frame correction is bounded by a fixed ratio of the local denoised prediction energy. This explains why RCSS can improve boundary continuation without applying unconstrained modifications to the solver state.

C Additional Evaluation Details

C.1 Case-Level Aggregation

The benchmark contains cases with different numbers of prompts. Some cases contain two prompts, while others contain three or more prompts. Directly averaging over all generated segments would give larger weights to cases with more prompts. Therefore, for segment-level metrics, we first average the scores within each case and then average over all cases:

$$S_{\text{case}} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \left(\frac{1}{N_c} \sum_{j=1}^{N_c} S_{c,j} \right), \tag{19}$$

where $\mathcal{C}$ is the set of benchmark cases, N_c is the number of evaluated segments in case c, and $S_{c,j}$ is the score of the j-th segment. This aggregation ensures that each case contributes equally regardless of the number of prompts.

C.2 Prompt-Level Text Alignment

For Text-Image Similarity [37], all videos are generated using extended prompts, while the evaluation references differ across experiments. Tables 1, 2, 5, and 6 use the corresponding short prompts, whereas the remaining experiments use extended prompts. Each segment is evaluated against its corresponding prompt or transition description rather than the full prompt sequence. Consequently, absolute similarity scores across these settings are not directly comparable, while comparisons within each experiment follow a consistent evaluation protocol.

C.3 Random Seeds

All quantitative results in the main paper are averaged over three random seeds. This reduces the influence of favorable or unfavorable random initialization and provides a more stable estimate of each method's performance. The same seed set is used for all compared methods whenever the baseline implementation supports seed control.

C.4 Blind Human Study Protocol

Our blind study includes 30 participants and 40 anonymized cases covering all ten prompt-transition modes and three random seeds. We evaluate MEVG [33], Video-Infinity [42], DiTCtrl [7], and FREQSYNC using five dimensions, each scored on a five-point scale: (i) target semantic correctness, (ii) source leakage and artifacts, (iii) transition smoothness, (iv) visual quality, and (v) overall preference. For criterion (ii), a higher score means fewer unwanted source remnants, ghosting, duplicated subjects, or artifacts, so all five reported metrics share the same higher-is-better direction. Table 3 reports the mean ratings. We treat this study as complementary perceptual evidence rather than a replacement for matched automatic metrics.

D Additional Quantitative Results

Table 5 reports the per-seed results of all compared methods under three random seeds, namely 42, 123, and 999. We also report the averaged scores across these three seeds, which are used in the main paper. The average scores are computed from the original unrounded per-seed results, while the displayed per-seed values are rounded for readability. Therefore, recomputing averages directly from the displayed rounded values may lead to minor differences in the last decimal place. This per-seed breakdown provides a more detailed view of the stability of each method across different random initializations.

Table 5: Per-seed quantitative results of all compared methods. We report the results of MEVG [33], Video-Infinity [42], DiTCtrl [7], and FREQSYNC under three random seeds: 42, 123, and 999. The shaded average rows correspond to the results reported in the main paper. CSCV and Motion Smoothness are reported as percentages.

Method	Seed	CSCV (%)↑	Motion Smooth. (%)↑	Text-Image Similarity↑	Visual Quality↑	T2V Align.↑	Physical Consist.↑
MEVG [33]	42	68.09	95.54	29.86	1.957	2.778	2.261
MEVG [33]	123	67.23	96.04	29.29	2.008	2.868	2.306
MEVG [33]	999	66.34	95.39	29.52	1.946	2.674	2.272
MEVG [33]	Average	67.22	95.66	29.56	1.970	2.773	2.280
Video-Infinity [42]	42	83.72	96.53	29.92	3.255	3.511	3.200
Video-Infinity [42]	123	83.63	96.38	29.75	3.210	3.347	3.162
Video-Infinity [42]	999	83.52	96.42	29.49	3.275	3.409	3.268
Video-Infinity [42]	Average	83.62	96.45	29.72	**3.247**	3.422	3.210
DiTCtrl [7]	42	86.46	98.48	30.35	3.129	3.753	3.230
DiTCtrl [7]	123	87.45	98.59	30.48	3.171	3.801	3.275
DiTCtrl [7]	999	84.42	98.25	30.14	3.142	3.721	3.181
DiTCtrl [7]	Average	<u>86.11</u>	**98.44**	<u>30.33</u>	3.147	<u>3.758</u>	<u>3.228</u>
FREQSYNC (Ours)	42	87.39	98.19	30.50	3.244	3.864	3.327
FREQSYNC (Ours)	123	87.00	98.42	30.58	3.203	3.854	3.191
FREQSYNC (Ours)	999	84.91	98.15	30.51	3.195	3.829	3.203
FREQSYNC (Ours)	Average	**86.43**	<u>98.25</u>	**30.53**	<u>3.214</u>	**3.849**	**3.240**

Table 6: Paired 100-case analysis against DiTCtrl. Differences are FreqSync minus DiTCtrl; CSCV differences are in percentage points (pp). All listed confidence intervals exclude zero. Minor differences from subtracting displayed scores arise from rounding.

Metric	FreqSync	DiTCtrl	Difference	95% CI	Holm p
CSCV (%)	**87.5133**	85.7334	+1.7799	[0.8623, 2.7845]	0.00124
Text–Image Similarity	**30.5638**	30.3236	+0.2402	[0.0244, 0.4653]	0.04325
Visual Quality	**3.2980**	3.0855	+0.2124	[0.1536, 0.2725]	$< 10^{-4}$
T2V Alignment	**3.9049**	3.7341	+0.1708	[0.1003, 0.2429]	0.00012
Physical Consistency	**3.3885**	3.1571	+0.2314	[0.1396, 0.3251]	$< 10^{-4}$

E Paired Statistical Significance

To assess the reliability of improvements over DiTCtrl [7], we evaluate 100 balanced cases covering all ten transition modes, matched on prompts, cases, random seed, and generation protocol. We form stratified paired-bootstrap 95% confidence intervals (CIs), apply case-level sign-flip tests, and correct the resulting p-values with the Holm procedure. Table 6 reports five metrics for which the corrected comparisons are available. The CSCV gain is +1.78 pp with a 95% CI of $[+0.86, +2.78]$ pp and Holm-adjusted $p = 0.00124$. All five bootstrap confidence intervals exclude zero, and the corresponding Holm-adjusted p-values are below 0.05.

F Cross-Backbone Evaluation

We extend the evaluation to two pre-trained video generators: Wan2.1-1.3B [44], based on DiT, and VideoCrafter2 [11], based on a U-Net. Within each backbone, DiTCtrl [7] and FREQSYNC use the

Table 7: Cross-backbone evaluation. All metrics are higher-is-better; CSCV and Motion Smoothness are percentages. Best results are bold within each backbone.

Backbone	Method	CSCV (%)	Motion (%)	Text–Image	VQ	T2V	Physical
VideoCrafter2	DiTCtrl	81.04	97.15	34.01	**3.095**	**3.665**	**3.057**
VideoCrafter2	FREQSYNC	**86.50**	**99.35**	**34.23**	2.836	3.576	2.842
Wan2.1-1.3B	DiTCtrl	84.05	**98.25**	31.91	3.267	3.900	3.404
Wan2.1-1.3B	FREQSYNC	**87.00**	98.04	**33.50**	**3.872**	**4.095**	**3.809**

Table 8: Controlled 40-case frequency-routing ablations. Mean $\pm$ standard deviation; all scores are higher-is-better. The routing tuple gives source weights (L, M, H).

Variant	Source routing (L, M, H)	CSCV (%)	Text–Image Similarity	T2V Alignment
Full FREQSYNC	$(0.98, 0.68, 0.035)$	$\mathbf{89.139 \pm 6.774}$	34.013 ± 3.019	$\mathbf{3.951 \pm 0.370}$
Low+Mid-only	$(0.98, 0.68, 0)$	88.956 ± 7.121	$\mathbf{34.027 \pm 3.028}$	3.891 ± 0.369
High-only	$(0, 0, 0.98)$	86.302 ± 7.335	33.913 ± 3.064	3.805 ± 0.418
Reverse Routing	$(0.035, 0.68, 0.98)$	85.795 ± 8.250	33.885 ± 2.939	3.821 ± 0.349
Uniform Routing	$(0.565, 0.565, 0.565)$	88.380 ± 7.506	33.964 ± 3.075	3.903 ± 0.411
w/o FS-SCA	—	85.693 ± 8.385	33.570 ± 2.902	3.905 ± 0.361

same cases, seed, checkpoint, resolution, sampling steps, classifier-free guidance, window length, and overlap. This comparison tests transfer beyond CogVideoX-2B.

As shown in Table 7, CSCV improves by 5.46 pp on VideoCrafter2 (95% CI $[3.85, 7.13]$ pp) and by 2.94 pp on Wan2.1 (95% CI $[0.27, 5.50]$ pp). On Wan2.1, FREQSYNC improves five of six metrics. On VideoCrafter2, boundary continuity, motion smoothness, and text–image alignment improve, while the three perceptual-quality metrics decrease. The factorized U-Net pathways and shorter four-frame overlap in this setting require stronger temporal-memory and stitching corrections, which may suppress details while improving continuity. The cross-backbone findings therefore support transfer of the boundary-coupling principle while also exposing architecture-dependent trade-offs.

G Controlled Frequency-Routing Ablations

To test the role of frequency allocation directly, we run six controlled variants on the same 40 cases, spanning all ten transition modes with matched prompts, seed, generation protocol, and evaluation. In Table 8, (L, M, H) denotes the source-conditioned routing strengths applied to low-, mid-, and high-frequency components of FS-SCA. Setting a source weight to zero removes that *source correction*, not the corresponding native target-frequency component. Accordingly, "high-only" is not a removal of low-frequency information from the target branch.

Low+Mid-only nearly matches the full method, while High-only and Reverse Routing lower CSCV by 2.84 and 3.34 pp, respectively. Uniform Routing controls for mean source strength but lowers all three metrics relative to the full model. Removing FS-SCA decreases CSCV by 3.446 pp and Text–Image Similarity by 0.443. This controlled intervention supports asymmetric source-low/mid and target-high routing rather than an explanation based only on stronger uniform coupling. It also extends the qualitative boundary diagnosis beyond a single failure example.

H Hyperparameter Sensitivity

We assess local robustness around the fixed default configuration with a paired one-factor-at-a-time (OFAT) sweep and a separate cutoff–layer interaction study. The same default configuration is used across cases within each study, without per-case tuning. For the short temporal Fourier axis, the sweep logs describe selected partitions using the discrete "DC + non-DC bins" convention; the implementation settings in Sec. 4.1 report normalized cutoffs. Each study is evaluated under its corresponding experimental protocol, and comparisons are made within the same setting.

Tables 9 and 10 show that FREQSYNC maintains stable performance under variations in frequency cutoffs, routing strengths, and active layers, with no alternative configuration consistently outper-

Table 9: Paired OFAT sensitivity analysis. Absolute scores are shown for each configuration; the default is the reference. "DC+k" is the logged temporal-frequency-bin partition. No tested alternative dominates the default across all metrics.

Family	Setting	CSCV (%)	Text–Image	VQ	T2V	Physical	Motion (%)
Default	low=2, low+mid=3, 8 layers	88.993	34.083	3.395	3.984	3.419	98.184
Low cutoff	DC+1 non-DC	89.326	34.161	3.414	3.987	3.377	98.191
Low cutoff	DC+3 non-DC	88.890	34.092	3.410	3.960	3.453	98.183
Low+mid cutoff	DC+2, mid collapsed	88.625	34.112	3.300	3.862	3.295	98.190
Low+mid cutoff	DC+4	88.845	34.078	3.311	3.925	3.352	98.182
Routing	$0.8\times$	88.315	34.123	3.298	4.014	3.302	98.191
Routing	$1.2\times$	89.228	34.100	3.324	3.926	3.323	98.184
Active layers	$[20, 30)$, 10 layers	88.069	34.108	3.425	4.023	3.338	98.161
Active layers	$[24, 30)$, 6 layers	88.967	34.082	3.333	3.865	3.445	98.206

Table 10: Cutoff–layer interaction sweep. Every cell lists CSCV (%)/T2V Alignment. These interaction results come from a separate paired study and should not be combined numerically with the OFAT table.

Low partition	10 layers $[20, 30)$	8 layers $[22, 30)$	6 layers $[24, 30)$
DC+1	87.514 / 4.000	88.476 / 3.799	88.739 / 3.846
DC+2	87.000 / 3.948	88.289 / 3.918	89.494 / 3.866
DC+3	87.491 / 4.073	88.070 / 3.694	88.697 / 3.859

forming the default. All cutoff–layer interaction 95% CIs include zero, indicating no statistically significant interaction or evidence of fragile parameter dependence. The three-band design provides a balanced performance across metrics, while only the $0.8\times$ routing setting shows a mild degradation on fast-motion cases. Overall, FREQSYNC is not highly sensitive to these hyperparameters around the default configuration and requires no per-case tuning.

I Inference Efficiency

We compare FREQSYNC with DiTCtrl [7] on the same NVIDIA A800-SXM4-80GB GPU using CogVideoX-2B, 480×720 resolution, 50 denoising steps, and matched generation settings. Each 2–5-prompt sequence receives one warm-up and three CUDA-synchronized timed runs. We time the core latent-sampling interval and exclude shared model loading, text encoding, VAE decoding, and video I/O. FLOPs are counted using one multiply–accumulate as two floating-point operations; reported memory is peak allocated GPU memory. This protocol compares the proposed sampling components under controlled conditions, not end-to-end wall-clock video-delivery latency.

As shown in Table 11, peak allocated GPU memory increases by only 0.23 GiB (233 MiB, 1.37%). Across 2–5 prompts, the sampling-time overhead remains within 8.07–9.07%, while the counted-FLOP overhead stays at 4.697% and peak memory remains nearly constant. Overall, FREQSYNC achieves improved generation quality with modest and stable inference overhead, without introducing additional trainable parameters or training costs.

J Additional Diagnostic Visualizations

We also analyze temporal-frequency energy around prompt boundaries. Figure 7 reports normalized temporal FFT energy under different boundary window radii. FREQSYNC exhibits stronger low-frequency energy near the boundary, which indicates more stable structural and motion continuation. Meanwhile, its high-frequency variation is consistently lower than DiTCtrl [7], suggesting that the transition contains fewer abrupt high-frequency changes. This behavior is consistent with the design of FS-SCA and RCSS: low- and mid-frequency components are used to carry continuity cues, while high-frequency components remain target-dominant and are constrained to reduce boundary-local flicker, ghosting, and duplicated traces.

Table 11: Inference cost as prompt count increases. Runtime is mean ± standard deviation over three timed runs (seconds); memory is peak allocated GiB. Counted FLOPs are in TFLOPs per sequence. Percentage increases are relative to DiTCtrl.

Prompts	Sampling time (s)			Memory (GiB)			Counted TFLOPs		
	DiTCtrl	FreqSync	Δ	DiTCtrl	FreqSync	Δ	DiTCtrl	FreqSync	Δ
2	461.05 ± 0.28	502.85 ± 2.41	+9.07%	16.51	16.74	+1.39%	71,998.39	75,379.93	+4.697%
3	923.07 ± 0.39	1000.50 ± 0.69	+8.39%	16.53	16.76	+1.39%	143,996.77	150,759.87	+4.697%
4	1383.91 ± 0.47	1500.52 ± 1.76	+8.43%	16.56	16.79	+1.37%	215,995.16	226,139.80	+4.697%
5	1851.45 ± 0.92	2000.84 ± 0.97	+8.07%	16.59	16.81	+1.37%	287,993.55	301,519.73	+4.697%

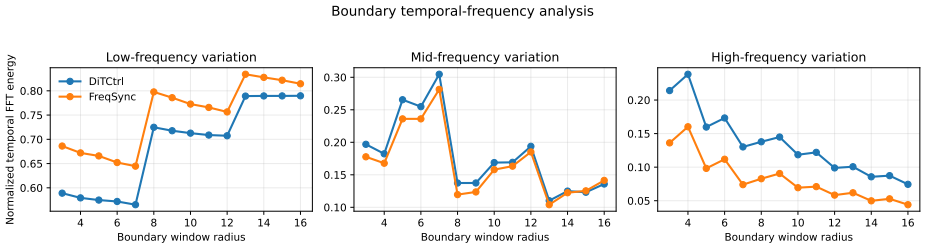


Figure 7: Boundary temporal-frequency analysis. FREQSYNC shows stronger low-frequency continuation and lower high-frequency variation around prompt boundaries. This supports our frequency-synchronized boundary coupling design, where low-frequency components preserve structural continuity and high-frequency instability is suppressed near transitions.

K Additional Qualitative Visualizations

The main paper presents a representative qualitative comparison and a visual ablation study. In this section, we provide additional qualitative visualizations from three perspectives. First, we show more generated examples of FREQSYNC under diverse multi-prompt sequences and transition types. Second, we present a single-prompt long-video generation application, where the same prompt is assigned to all segments and FREQSYNC is used to extend the video duration. Third, we provide additional qualitative comparisons with Kling-2.6 [1], MEVG [33], Video-Infinity [42], and DiTCtrl [7] on challenging three-prompt sequences.

K.1 Additional Generated Examples

Figure 8 shows two-prompt examples with a single prompt boundary, while Figure 9 shows three-prompt examples with two consecutive prompt boundaries. These examples cover diverse changes in background, lighting, location, subject, and action conditions. For each sequence, we visualize uniformly sampled frames together with the corresponding prompt transition labels. The results further show that FREQSYNC can preserve temporal continuity while allowing target-prompt details to emerge progressively and remain visually coherent throughout the sequence.

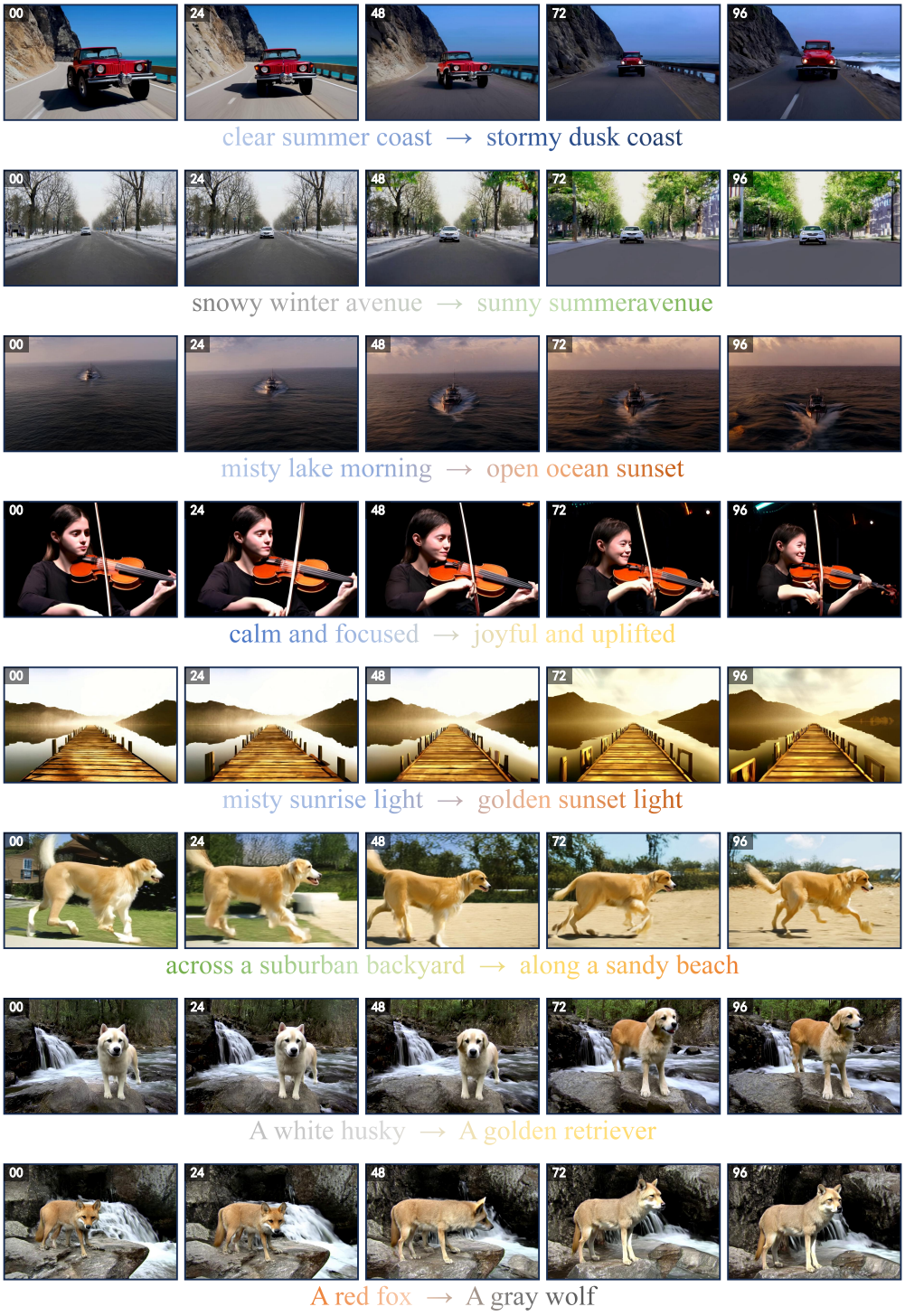


Figure 8: Additional two-prompt generated examples. We show eight representative two-prompt sequences. Each row corresponds to one generated video and visualizes key frames across a single prompt boundary. Across diverse transition types, FREQSYNC preserves the main subject, scene layout, and motion trajectory while allowing the target-prompt condition to emerge progressively. These examples further illustrate that the frequency-synchronized boundary coupling used by FS-SCA and RCSS can reduce abrupt visual changes without suppressing prompt-specific details.

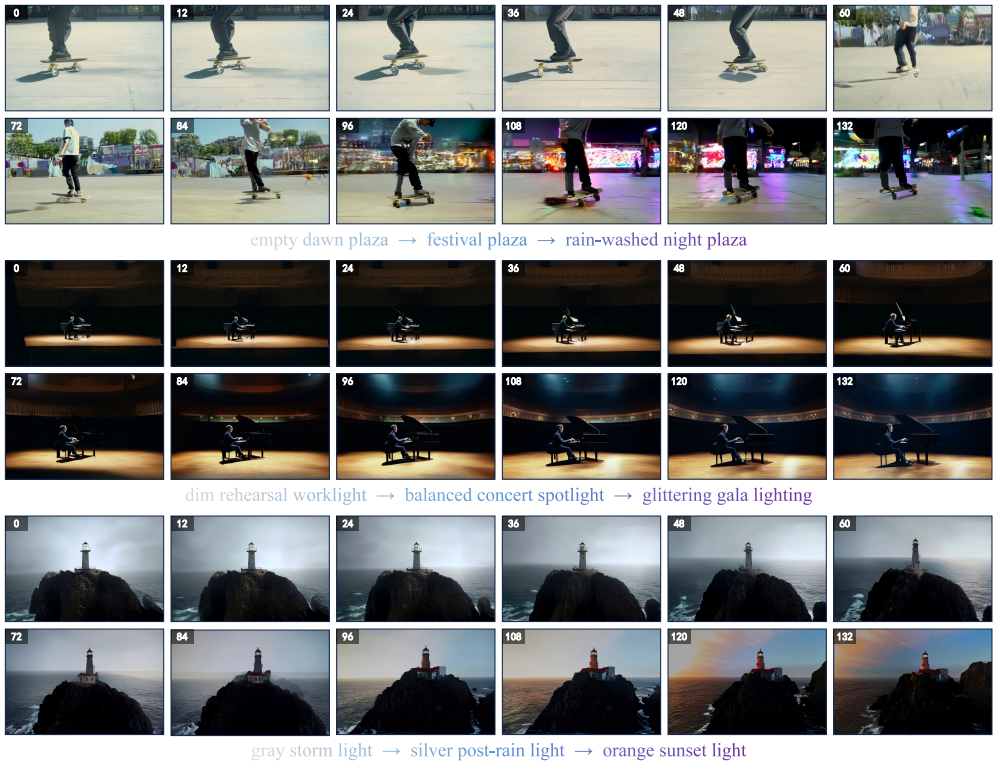


Figure 9: Additional three-prompt generated examples. We show three representative three-prompt sequences, where each example spans two rows to cover a longer temporal range and two consecutive prompt boundaries. FREQSYNC maintains coherent temporal evolution across multiple semantic changes, such as scene, illumination, and action transitions. The results suggest that the proposed boundary coupling is not limited to a single transition, but can also support longer prompt sequences with successive target-prompt activations.

K.2 Single-Prompt Long-Video Generation

Although FREQSYNC is designed for multi-prompt long video generation, it can also be applied to single-prompt long-video generation by assigning the same prompt to every segment. In this setting, there is no semantic prompt transition, but adjacent segments still need to be connected smoothly to avoid temporal seams, flicker, or layout drift. Figure 10 shows four single-prompt long-video examples generated by FREQSYNC. The results demonstrate that the proposed boundary coupling can also support temporally coherent long-video extension under a fixed prompt, maintaining stable scene content and smooth motion across segment boundaries.

K.3 Additional Qualitative Comparisons

We further provide two additional qualitative comparisons against Kling-2.6 [1], MEVG [33], Video-Infinity [42], and DiTCtrl [7]. As shown in Figures 11 and 12, the compared methods often exhibit weak prompt progression, inconsistent subject appearance, duplicated visual traces, or abrupt boundary changes. In contrast, FREQSYNC maintains more stable temporal trajectories and smoother prompt transitions while preserving target-prompt semantics.

L Broader Impacts

FREQSYNC aims to improve training-free multi-prompt long video generation by producing smoother prompt transitions and more coherent long videos. This may benefit creative video production, storyboarding, visual prototyping, education, and research on controllable video generation, especially

because the method does not require additional model training. At the same time, improvements in text-to-video generation may also introduce potential risks. More coherent long videos could be misused to create misleading synthetic media, fabricated events, or deceptive visual narratives. Since FREQSYNC builds on pre-trained video generators, it may also inherit biases, unsafe behaviors, or inappropriate generation tendencies from the underlying model and its training data. In addition, when generated videos depict people, locations, or sensitive events, misuse may raise concerns related to misinformation, impersonation, privacy, and public trust. We emphasize that FREQSYNC is a research method for studying prompt-boundary coupling rather than a deployed generation system. We recommend that practical deployments combine such methods with standard safety filters, watermarking or provenance mechanisms, content moderation, and usage policies that restrict malicious applications. We also encourage users to clearly disclose synthetic content when generated videos are used in public-facing contexts.

M Licenses and Existing Assets

We use several existing models, codebases, benchmarks, and evaluation tools in this work. All external assets are used only for research and evaluation purposes, and we cite their original papers or official releases in the main text. We do not redistribute third-party model weights, benchmark code, or evaluation packages as part of this submission. When using existing assets, we follow their official licenses and terms of use.

Table 12: Existing assets used in this work. We cite the original sources in the main paper and follow their corresponding licenses or usage terms.

Asset	Usage in this work	License / Terms
CogVideoX-2B [50]	Pre-trained video generation backbone	Apache-2.0 License
DiTCtrl / MPVBench [7]	Baseline method and multi-prompt evaluation setting	Apache-2.0 License for released inference code
VBench [22]	Motion Smoothness evaluation	Apache-2.0 License
CLIP [37]	Text-Image Similarity and CSCV-related feature similarity	MIT License
VideoScore2 [15]	Visual Quality, T2V Alignment, and Physical Consistency evaluation	MIT License
MEVG [33]	Baseline comparison	Original paper and official project terms
Video-Infinity [42]	Baseline comparison	Original paper and official project terms
Kling-2.6 [1]	Qualitative comparison with a commercial system	Official service terms

N Limitations

FREQSYNC has several limitations. First, it relies on the generation capacity of the underlying pre-trained video diffusion model. If the base model fails to synthesize a rare object, complex interaction, or unusual camera motion, FREQSYNC cannot fully recover the missing content. Second, extremely large semantic changes may still produce imperfect transitions, especially when adjacent prompts require incompatible layouts, object counts, or camera trajectories. Third, FS-SCA introduces additional inference overhead because it evaluates an extra source-conditioned attention path in scheduled layers and denoising steps. Finally, as with other text-to-video generation methods, generated videos may inherit biases or unsafe behaviors from the underlying model and training data. We recommend standard safety filtering and responsible deployment practices for open-ended use.

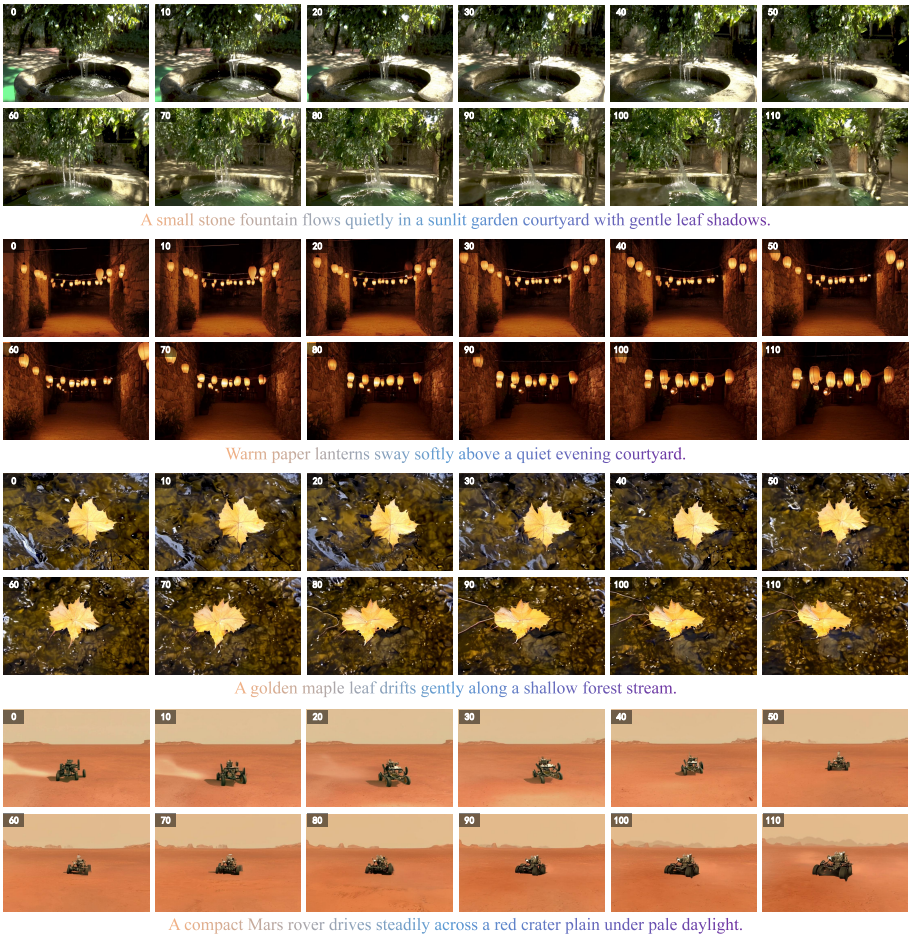


Figure 10: Single-prompt long-video generation with FREQSYNC. We apply FREQSYNC to single-prompt long-video generation by assigning the same prompt to all segments. The four examples show a sunlit courtyard fountain, warm paper lanterns, a drifting maple leaf, and a Mars rover driving across a red crater plain. Even without semantic prompt changes, long-video generation still requires stable cross-segment continuation. FREQSYNC produces smooth motion and coherent visual content over extended durations, suggesting that its boundary-coupling mechanism can also benefit single-prompt video extension.



Figure 11: Additional qualitative comparison on a skateboarder sequence. The base prompt describes a young skateboarder gliding forward through an open public square, while the segment prompts transition from an empty dawn plaza to a festival plaza and then to a rain-washed night plaza. Kling-2.6 [1] produces visually stable frames but shows limited scene progression. MEVG [33] and Video-Infinity [42] suffer from identity, viewpoint, or motion inconsistency. DiTCtrl [7] improves cross-segment continuity but introduces ghosting and blurred boundary artifacts. In contrast, FREQSYNC preserves the skateboarder's motion trajectory while progressively changing the plaza environment and illumination.

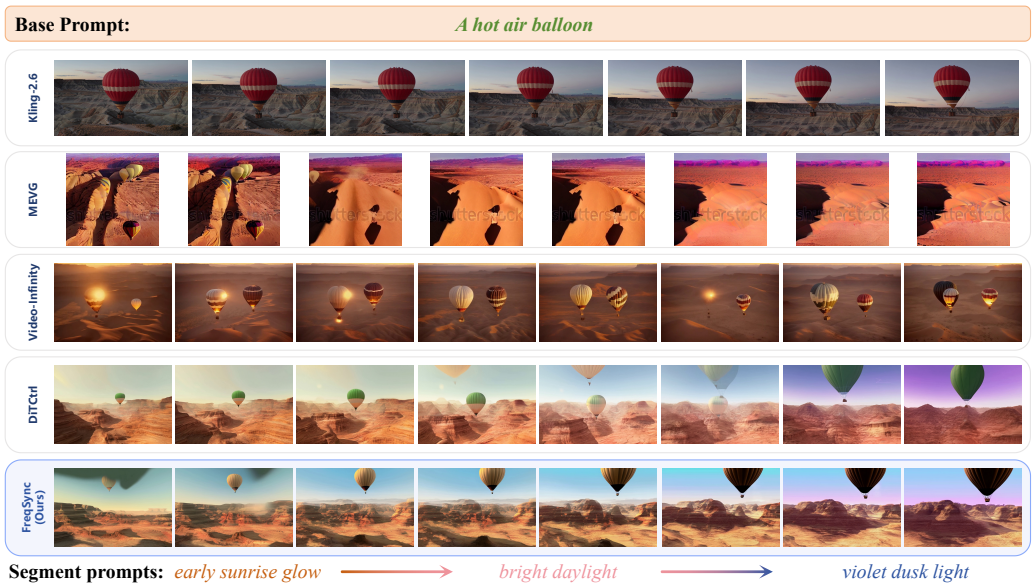


Figure 12: Additional qualitative comparison on a hot-air-balloon sequence. The base prompt describes a hot air balloon, and the segment prompts change the lighting condition from early sunrise glow to bright daylight and then to violet dusk light. Kling-2.6 [1] preserves the balloon appearance but shows limited temporal and illumination transition. MEVG [33] and Video-Infinity [42] produce noticeable subject or scene inconsistencies, including multiple balloons, unstable scale, or abrupt viewpoint changes. DiTCtrl [7] maintains partial continuity but shows over-preserved source-side evidence and boundary-local artifacts. FREQSYNC better preserves the balloon identity and desert layout while producing a smoother lighting transition across the prompt sequence.

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 abstract and introduction clearly state the main claims, which accurately reflect the contributions and scope of our work.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss limitations in Appendix N.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: Appendix B provides the assumptions and proof for the bounded residual correction property used by FS-SCA and RCSS.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We describe the algorithm, inference procedure, benchmark construction, evaluation metrics, and main hyperparameters in Secs. 3, 4, and Appendix A.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: The code and configurations are released at: `https://yipanxu-22.github.io/FreqSync-project/`.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: In Sec. 4.1 and Appendix A.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Statistical significance tests and confidence intervals are reported in Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: GPU specifications and inference costs are reported in Sec. 4.1 and Appendix I.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have read and complied with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss potential positive uses, misuse risks, inherited model biases, and mitigation strategies in Appendix L.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We cite the creators of all models, baselines, benchmarks, and evaluation tools used in our paper, and summarize their licenses or usage terms in Appendix M.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: The code and configurations are released at: `https://yipanxu-22.github.io/FreqSync-project/`.

    Guidelines:

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

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

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

    Answer: [No]

    Justification: Full participant instructions and compensation details are not provided, but the human evaluation protocol is described in Sec. 4.6 and Appendix C.4.

    Guidelines:

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

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

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

    Answer: [No]

    Justification: Formal IRB approval or equivalent ethics review was not obtained.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs are not used as an important, original, or non-standard component of the core method or experiments.

    Guidelines:

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