MotionCFG: Boosting Motion Dynamics via Semantic Motion Sharpening

Byungjun Kim
KAIST
Daejeon 34141, Republic of Korea
bjkim@kaist.ac.kr

Soobin Um[†]
DGIST
Daegu 42988, Republic of Korea
soobin.um@dgist.ac.kr

Jong Chul Ye[†]
KAIST
Daejeon 34141, Republic of Korea
jong.ye@kaist.ac.kr

Abstract

Despite recent advancements in Text-to-Video (T2V) synthesis, generating high-fidelity and dynamic motion remains a significant challenge. Existing methods primarily rely on Classifier-Free Guidance (CFG), but standard CFG treats all semantic dimensions uniformly and provides no mechanism to selectively resolve ambiguity along the motion axis. To address this, we propose MotionCFG, a training-free approach that introduces selective guidance along the motion subspace of the condition embedding. Specifically, we construct a motion-perturbed negative condition by injecting Gaussian noise into motion-related embeddings, and steer generation away from it to selectively amplify motion intent. We show theoretically that this procedure implicitly approximates *Semantic Laplacian Sharpening*—a second-order correction that suppresses motion-ambiguous regions of the score landscape and amplifies well-resolved dynamic peaks. Combined with a piecewise guidance schedule that confines intervention to the early denoising steps, MotionCFG consistently improves motion dynamics across state-of-the-art T2V frameworks with negligible overhead. We further demonstrate that this Laplacian sharpening principle generalizes beyond motion, effectively steering complex, non-linear concepts such as precise object numerosity that are typically difficult to modulate via standard text-based guidance. Our project page is available on https://motion-cfg.vercel.app/.

1 Introduction

Text-to-Video (T2V) generation has seen remarkable progress with the advent of large-scale diffusion-based generative frameworks [4, 6, 33, 31], enabling the creation of realistic videos that faithfully reflect user-intended text descriptions. A key contributor to this success is Classifier-Free Guidance (CFG) [11], which strengthens text-conditioning signals by extrapolating the score function away from an unconditional null embedding.

Despite this progress, recent studies consistently report that T2V models struggle to produce videos with strong dynamic motion, often yielding near-static outputs even for explicitly motion-rich prompts (*e.g.*, "a cat jumping over a fence") [24, 26]. We attribute this to the indiscriminate nature of standard CFG: since it extrapolates the entire condition vector at once, semantic axes that are under-represented

[†]Corresponding authors.

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

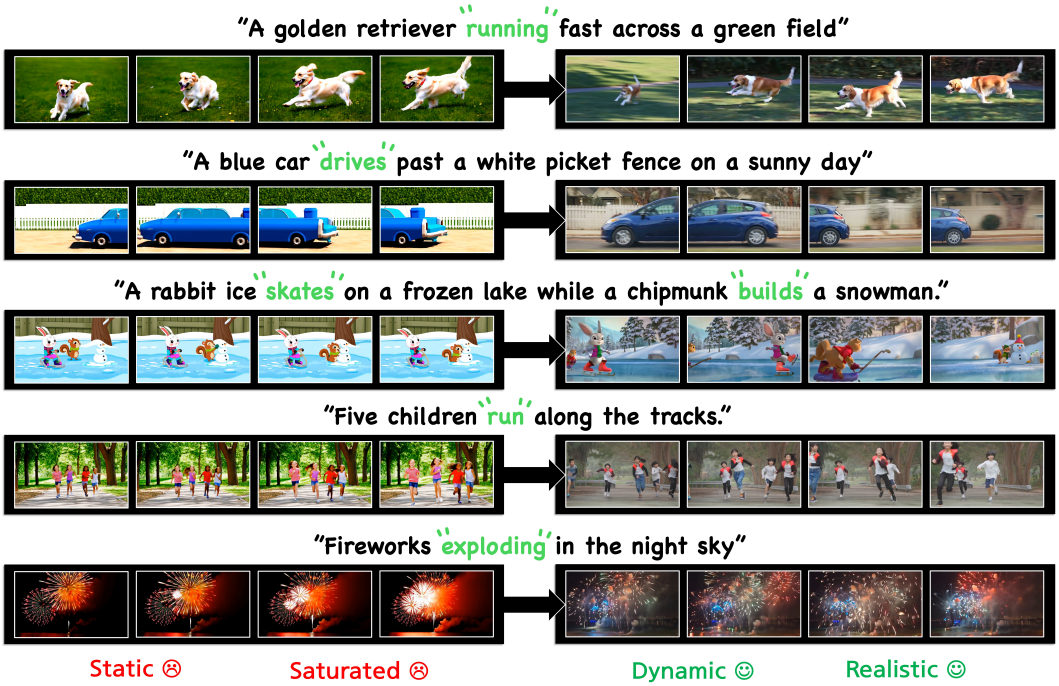


Figure 1: **Showcase of MotionCFG.** While Standard CFG (left) produces over-saturated colors and static outputs that often fail to reflect the intended actions, MotionCFG (right) resolves motion ambiguity by selectively sharpening motion-related embeddings, thereby yielding realistic and physically dynamic videos faithful to the prompt. Motion words are highlighted in green.

in the text encoder, such as motion [24, 22], can remain under-resolved when competing with more dominant dimensions such as object identity or scene layout. Incorporating explicit negative prompts (*e.g.*, "static") is a natural attempt, but often offers marginal gains; see Table 4 for instance.

In this work, we propose **MotionCFG**, a simple-yet-powerful framework that addresses motion ambiguity in a zero-shot, training-free fashion. While standard CFG guides generation uniformly away from an unconditional null embedding, MotionCFG introduces a *motion-targeted* guidance mechanism that operates selectively on the motion subspace of the condition. During inference, we identify motion-related tokens (*e.g.*, action verbs) from the input prompt, either manually or via a lightweight large language model (LLM), and inject Gaussian noise only into the embeddings of these motion tokens. The resulting motion-perturbed condition serves as the negative anchor, and we steer generation *away* from this perturbed neighborhood to selectively sharpen motion semantics without affecting other attributes. To balance motion dynamics with visual fidelity, we further introduce a Piecewise Guidance Scheduler that restricts our intervention to the early sampling steps, where the global motion layout is determined.

Beyond its simplicity, MotionCFG admits an interpretable theoretical viewpoint. We show that the procedure implicitly approximates the Laplacian of the score (or velocity [20]) field restricted to the motion subspace of the condition. We refer to this principle as *Semantic Laplacian Sharpening*: a correction that suppresses motion-ambiguous semantic regions and amplifies well-resolved dynamic peaks, extending prior data-space Laplacian correction [5] to the condition space. This perspective also predicts that the same mechanism generalizes to other ambiguous semantic dimensions beyond motion, which we empirically verify on object numerosity (See Section 4.3).

We validate MotionCFG on state-of-the-art T2V models including Wan2.1 [31] (1.3B and 14B) and CogVideoX [33], demonstrating consistent improvements in motion dynamics with marginal compromise in text-video alignment, all at negligible computational overhead over standard CFG. Furthermore, consistent with our theoretical framework, we show that the same Semantic Laplacian Sharpening principle extends beyond motion to other notoriously difficult-to-control attributes such as precise object numerosity [3], establishing MotionCFG as a general approach for resolving fine-grained semantic ambiguity in text-conditioned generation.

2 Related Work

Text-to-Video Generation. Text-to-Video (T2V) generation has evolved rapidly, transitioning from 3D-UNet based diffusion models [12, 27] to Flow-Matching (FM) frameworks [20] utilizing Diffusion Transformers (DiT). State-of-the-art open-source models like CogVideoX [33] and WAN [31] leverage large-scale datasets and 3D causal VAEs to achieve high spatiotemporal consistency. Despite their impressive visual quality, these models heavily rely on standard cross-attention mechanisms for text conditioning, which often fail to parse fine-grained physical dynamics, leading to motion ambiguity in complex motion generation.

Motion Dynamics in Video Generation. Current text-to-video (T2V) models frequently suffer from a static bias, where generated outputs exhibit minimal temporal variation or resemble static images with subtle camera drift [24, 28]. To address this, DEMO [24] enhances motion synthesis by decomposing text conditioning into content and motion components, but requires specialized training with text-motion and video-motion supervision. Recent training-free approaches attempt to improve motion dynamics during inference. State Guidance [28] introduces frame-wise scheduled guidance over state-transition prompts to encourage temporal scene progression, while VideoGuide [18] enhances temporal consistency through teacher-guided denoising trajectory interpolation. In contrast, our MotionCFG directly perturbs motion-related semantic embeddings to construct a motion-aware negative branch for classifier-free guidance. MotionCFG operates entirely at inference time without requiring architectural modifications, additional optimization with memory overhead, or auxiliary teacher models, while remaining applicable to general T2V generation settings.

Guidance Mechanisms in Diffusion Models. Classifier-Free Guidance (CFG) is the de facto standard for aligning diffusion outputs with text prompts [11]. While effective for static attributes, it struggles with motion ambiguity since extrapolating away from an empty string ($\varnothing$) does not inherently resolve the specific dynamic intent. Recent works have explored explicit negative prompts, gradient-based guidance, or attention-map manipulations. Orthogonal to these approaches, Laplacian Score Sharpening [5] shows that subtracting the Laplacian of the score can sharpen overly smooth score landscapes and suppress ambiguous inter-mode regions. Motivated by this perspective, we extend score sharpening from the data space to the semantic condition space: MotionCFG constructs its negative branch by perturbing motion-related text embeddings, yielding an efficient inference-time approximation to a Laplacian correction along motion-relevant semantic directions.

3 Method

3.1 Preliminaries

Text-to-Video Diffusion Models. Modern diffusion-based text-to-video (T2V) models [4, 6, 33, 31] operate in the latent space of a 3D variational autoencoder (VAE) [15], which consists of an encoder $E(\cdot)$ and a decoder $D(\cdot)$. A video $\boldsymbol{x}$ is first compressed into a low-dimensional latent representation $\boldsymbol{z}_0 = E(\boldsymbol{x})$. The diffusion forward process gradually corrupts $\boldsymbol{z}_0$ by adding Gaussian noise according to a variance schedule, producing a noisy latent $\boldsymbol{z}_t$ at timestep t [10]. A neural network is then trained to reverse this corruption by learning to predict the added noise $\epsilon_\theta(\boldsymbol{z}_t, \boldsymbol{c})$ [10], where $\boldsymbol{c}$ denotes the text condition embedded by a pretrained text encoder (*e.g.*, T5 [23]). Generation proceeds by iteratively denoising from pure noise $\boldsymbol{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ back to a clean latent $\boldsymbol{z}_0$ using the learned noise predictor, which is then decoded to pixel space via $\hat{\boldsymbol{x}} = D(\boldsymbol{z}_0)$.

Classifier-Free Guidance. Classifier-Free Guidance (CFG) [11] improves text-conditional generation by extrapolating between the unconditional and conditional noise predictions:

$$\tilde{\epsilon}_\theta(\boldsymbol{z}_t, \boldsymbol{c}, \varnothing) = \epsilon_\theta(\boldsymbol{z}_t, \varnothing) + \omega_{\text{std}} \left(\epsilon_\theta(\boldsymbol{z}_t, \boldsymbol{c}) - \epsilon_\theta(\boldsymbol{z}_t, \varnothing)\right), \tag{1}$$

where $\varnothing$ denotes the null (unconditional) embedding and $\omega_{\text{std}} > 1$ is the guidance scale. By extrapolating the noise prediction away from the unconditional estimate, CFG sharpens the conditional distribution and strengthens text alignment. However, this extrapolation operates on the entire condition vector $\boldsymbol{c}$ uniformly. Since the condition embedding encodes multiple semantic axes simultaneously (*e.g.*, object identity, scene layout), fine-grained axes such as motion can remain under-resolved when the extrapolation is dominated by other, more salient dimensions. This motivates a guidance mechanism that can selectively target the motion subspace of the condition.

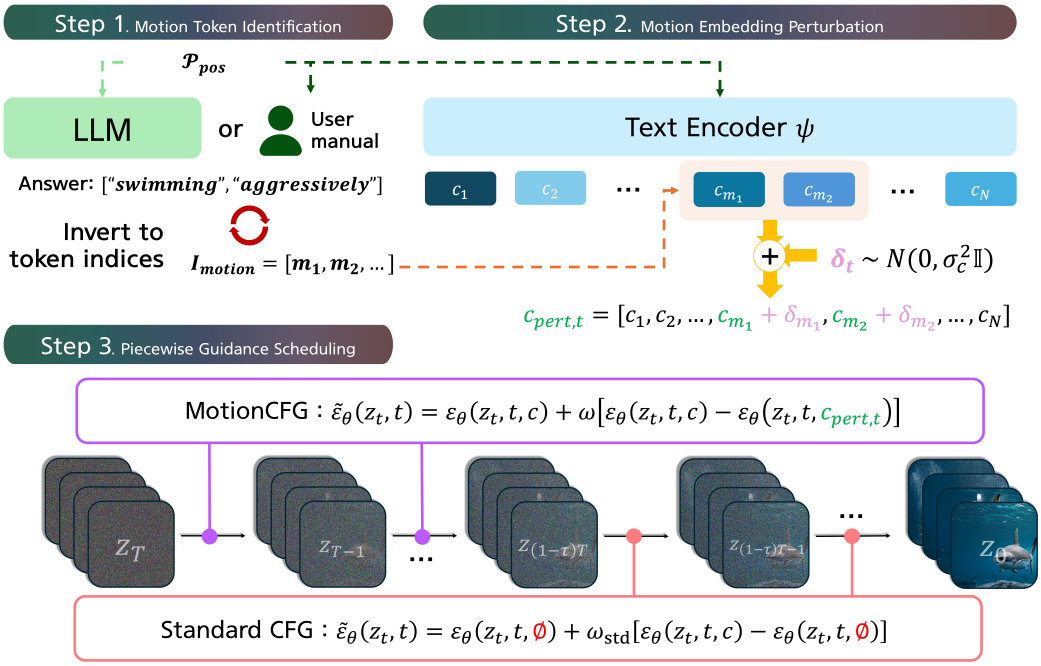


Figure 2: **Overview of the MotionCFG pipeline. Step 1:** Motion-related tokens in the prompt are identified via an LLM. **Step 2:** Gaussian noise (δ_t) is injected exclusively into these motion text embeddings to generate a perturbed condition ($c_{pert,t}$). **Step 3:** A piecewise guidance schedule applies MotionCFG during the early sampling steps to establish robust motion trajectories, before reverting to standard CFG to refine spatial details.

A natural first attempt is *negative prompting* [7]: replacing the null embedding in Eq. (1) with a text condition that explicitly describes static or low-motion scenes (*e.g.*, "static, still picture"), thereby steering the generation away from stillness. In practice, however, this coarse textual intervention yields only marginal gains over standard null-prompted CFG (see Table 4), as a single hand-crafted sentence cannot capture the fine-grained motion semantics distributed across individual tokens.

3.2 MotionCFG

We introduce MotionCFG, a training-free sampler that addresses the under-resolution of the motion axis by incorporating a motion-perturbed negative anchor. Specifically, given an input text prompt, let $\boldsymbol{c} \in \mathbb{R}^{L \times d}$ denote its embedding sequence obtained from the pretrained text encoder, where L is the sequence length and d is the embedding dimension. We first identify a set of motion-related tokenized indices $\mathcal{I}_{\text{motion}}$ (*e.g.*, action verbs such as "jumping" or "running"). At each denoising step t, we sample a fresh noise vector $\boldsymbol{n}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and construct a perturbed embedding $\boldsymbol{c}_{\text{pert},t}$ by injecting noise exclusively into the text embeddings at the identified indices:

$$\boldsymbol{c}_{\text{pert},t}[i] = \begin{cases} \boldsymbol{c}[i] + \sigma_{\boldsymbol{c}}\, \boldsymbol{n}_t, & \text{if } i \in \mathcal{I}_{\text{motion}} \\ \boldsymbol{c}[i], & \text{otherwise} \end{cases} \tag{2}$$

where $\sigma_{\boldsymbol{c}} > 0$ controls the perturbation magnitude. The guided noise prediction is then computed by steering generation away from this perturbed neighborhood:

$$\tilde{\epsilon}_\theta(\boldsymbol{z}_t, \boldsymbol{c}, \boldsymbol{c}_{\text{pert},t}) = \epsilon_\theta(\boldsymbol{z}_t, \boldsymbol{c}_{\text{pert},t}) + \omega\left(\epsilon_\theta(\boldsymbol{z}_t, \boldsymbol{c}) - \epsilon_\theta(\boldsymbol{z}_t, \boldsymbol{c}_{\text{pert},t})\right), \tag{3}$$

where $\boldsymbol{c}$ and $\boldsymbol{c}_{\text{pert},t}$ denote the conditions derived from the original and perturbed embeddings, respectively. Note that the anchor point is different from standard CFG in Eq. (1). Intuitively, while standard CFG extrapolates between the *unconditional* and conditional noise predictions to strengthen overall text alignment, MotionCFG steers away from a *motion-degraded* condition, thereby selectively sharpening the motion semantics without affecting the remaining attributes.

Why perturb motion embeddings? MotionCFG intervenes on the final contextualized T5 embeddings received by the denoiser. The selected positions provide entry points for the requested

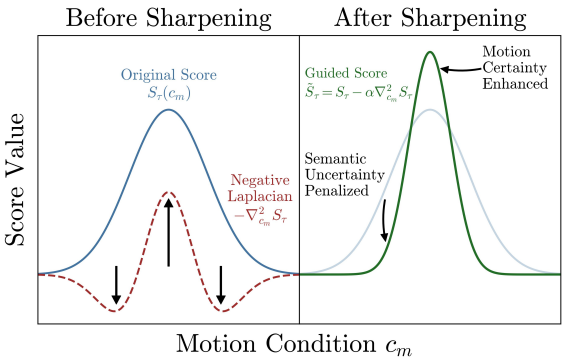


Figure 3: **Semantic Laplacian Sharpening.** Laplacian sharpening suppresses motion-ambiguous score regions and sharpens confident motion modes.

Algorithm 1 MotionCFG Sampling

Input: $\mathcal{P}, \epsilon_\theta, \mathcal{D}, T, \tau, \omega, \omega_{\text{std}}$
$\mathcal{I}_{\text{motion}} \leftarrow \text{MotionExtract}(\mathcal{P})$
$\boldsymbol{c} \leftarrow \text{Encode}(\mathcal{P})$
$\boldsymbol{c}_{\text{neg}} \leftarrow \text{Encode}(\mathcal{P}_{\text{neg}})$
$\boldsymbol{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T \dots 1$ **do**
  if $t > (1-\tau)T$ **then**
    $\boldsymbol{n}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\boldsymbol{c}_{\text{pert},t} \leftarrow \text{Perturb}(\boldsymbol{c}, \mathcal{I}_{\text{motion}}, \sigma_{\boldsymbol{c}}\boldsymbol{n}_t)$
    $\tilde{\epsilon} \leftarrow \epsilon_\theta(\boldsymbol{z}_t, \boldsymbol{c}) + \omega\big(\epsilon_\theta(\boldsymbol{z}_t, \boldsymbol{c}) - \epsilon_\theta(\boldsymbol{z}_t, \boldsymbol{c}_{\text{pert},t})\big)$
  else
    $\tilde{\epsilon} \leftarrow \epsilon_\theta(\boldsymbol{z}_t, \varnothing) + \omega_{\text{std}}\big(\epsilon_\theta(\boldsymbol{z}_t, \boldsymbol{c}) - \epsilon_\theta(\boldsymbol{z}_t, \varnothing)\big)$
  end if
  $\boldsymbol{z}_{t-1} \leftarrow \text{SchedulerStep}(\boldsymbol{z}_t, \tilde{\epsilon})$
end for
Return $\hat{\boldsymbol{x}} \leftarrow \mathcal{D}(\boldsymbol{z}_0)$

action, while motion arising from scene context, object interactions, camera behavior, and temporal dynamics remains modeled by the pretrained video generator. To examine whether these positions retain traces of the selected action tokens after contextualization, let $\mathbf{e}_i$ denote the input embedding at motion-token position i and $\mathbf{c}_j$ the final contextual embedding at position j. We measure each row's maximum cosine similarity to the selected input embeddings $s_j = \max_{i \in \mathcal{I}_{\text{motion}}} \cos(\mathbf{e}_i, \mathbf{c}_j)$.

Table 1 shows that the selected rows retain stronger traces of the original motion-token inputs despite contextual mixing. The highest-scoring row occurs at a selected position for every evaluated prompt, and most selected positions are recovered among the top-K rows. These measurements support the selected contextual rows as practical anchors for intervening on action-related conditioning, without requiring motion to be fully localized or disentangled at those positions. Appendix C.2 provides a controlled comparison of perturbing initial T5 input embeddings versus final contextual embeddings.

Table 1: **Motion-token localization.** We compare final contextual T5 rows with the selected input-token embeddings. $\text{TopK}(\mathbf{s})$ denotes the indices of the K highest-scoring rows, where $K = |\mathcal{I}_{\text{motion}}|$ and $\mathbf{s} = (s_1, \dots, s_L)$.

Measurement	Definition	Result
Motion-row similarity	$\frac{1}{K}\sum_{j \in \mathcal{I}_{\text{motion}}} s_j$	0.108
Other-row similarity	$\frac{1}{L-K}\sum_{j \notin \mathcal{I}_{\text{motion}}} s_j$	0.014
Top-1 localization	$\mathbf{1}\left[\arg\max_j s_j \in \mathcal{I}_{\text{motion}}\right]$	100%
Top-K recall	$\frac{\lvert \text{TopK}(\mathbf{s}) \cap \mathcal{I}_{\text{motion}} \rvert}{K}$	88.9%

Piecewise Guidance Scheduling. In video diffusion models, the global motion layout is established during the earliest denoising steps, while later steps refine high-frequency spatial details and textures [2, 14]. Applying Eq. (3) throughout the entire sampling process introduces unnecessary stochastic variance in the late stage, often resulting in visual artifacts. We therefore introduce a piecewise schedule controlled by a ratio $\tau \in [0, 1]$: MotionCFG is applied during the first τ fraction of the total steps to establish the motion trajectory, after which we revert to standard CFG (Eq. (1)) for spatial refinement. To jointly improve text alignment during the motion-encouraging stage, we anchor the baseline on $\boldsymbol{c}$ rather than $\boldsymbol{c}_{\text{pert},t}$:

$$\tilde{\epsilon}^*_\theta(\boldsymbol{z}_t, \boldsymbol{c}, \boldsymbol{c}_{\text{pert},t}) = \epsilon_\theta(\boldsymbol{z}_t, \boldsymbol{c}) + \omega\left(\epsilon_\theta(\boldsymbol{z}_t, \boldsymbol{c}) - \epsilon_\theta(\boldsymbol{z}_t, \boldsymbol{c}_{\text{pert},t})\right). \tag{4}$$

This two-stage approach with the $\boldsymbol{c}$-anchored formulation consistently outperforms the full-schedule variant ($\tau = 1.0$); see Figure 5 for details. The full procedure is summarized in Algorithm 1, and an overview of the pipeline is illustrated in Figure 2.

Motion Embeddings Identification. Identifying $\mathcal{I}_{\text{motion}}$ requires determining which parts of the text embeddings in the prompt carry motion semantics. Users can manually specify target verbs for best precision, but this becomes tedious when processing large prompt sets. To sidestep this, we automate the identification by querying a lightweight large language model (*e.g.*, Qwen2.5-7B-Instruct [32]) via in-context learning. Specifically, we feed the prompt with its tokenized indices (*e.g.*, `[0]A [1]cat [2]jumps`) and instruct the model to return the indices of words describing physical actions or dynamic changes. Importantly, the computational overhead introduced by this automated extraction is marginal compared to the overall video diffusion process. Detailed complexity analysis, along with the instruction template, is provided in the supplementary.

3.3 Connection to Semantic Laplacian Sharpening

Here we provide a theoretical insight into why MotionCFG is effective. In particular, we show that MotionCFG can be interpreted as an efficient, single-sample approximation of *Semantic Laplacian Sharpening*, a second-order correction that sharpens the score landscape along the motion dimensions of the condition space. For convenience, here we work with the score notation $\mathbf{S}_\theta \coloneqq -\boldsymbol{\epsilon}_\theta/\sigma_t$ [29, 8], which is equivalent to the noise prediction $\boldsymbol{\epsilon}_\theta$ up to a timestep-dependent scalar determined by the noise schedule σ_t. Under this notation, the MotionCFG update in Eq. (4) becomes

$$\tilde{\mathbf{S}}^*_\theta(\boldsymbol{z}_t, \boldsymbol{c}, \boldsymbol{c}_{\text{pert},t}) = \mathbf{S}_\theta(\boldsymbol{z}_t, \boldsymbol{c}) + \omega\left(\mathbf{S}_\theta(\boldsymbol{z}_t, \boldsymbol{c}) - \mathbf{S}_\theta(\boldsymbol{z}_t, \boldsymbol{c}_{\text{pert},t})\right). \tag{5}$$

To establish the connection to Laplacian sharpening, we first consider regularizing the score function by subtracting its Laplacian with respect to the motion condition:

$$\mathbf{S}^*_\theta(\boldsymbol{z}_t, \boldsymbol{c}) = \mathbf{S}_\theta(\boldsymbol{z}_t, \boldsymbol{c}) - \gamma\, \nabla^2_{\boldsymbol{c}_m} \mathbf{S}_\theta(\boldsymbol{z}_t, \boldsymbol{c}), \tag{6}$$

where $\boldsymbol{c}_m \in \mathbb{R}^{|\mathcal{I}_{\text{motion}}| \times d}$ denotes the rows of $\boldsymbol{c}$ indexed by $\mathcal{I}_{\text{motion}}$, and $\gamma > 0$ controls the regularization strength. Intuitively, the Laplacian $\nabla^2_{\boldsymbol{c}_m} \mathbf{S}_\theta$ measures how sensitive the score is to variations in the motion condition: it is near-zero in flat, ambiguous regions where multiple motion interpretations coexist, and strongly negative at sharp peaks where the motion intent is well-resolved. By subtracting this term, Eq. (6) penalizes the former and amplifies the latter, effectively sharpening the score landscape along the motion axis. This extends the Laplacian Score Sharpening of Chandran *et al*. [5], who apply an analogous correction in the *data space* to reduce visual hallucinations, to the *condition space*, hence the name *Semantic* Laplacian Sharpening. See Figure 3 for illustration of the concept.

The problem is that computing $\nabla^2_{\boldsymbol{c}_m} \mathbf{S}_\theta$ exactly requires evaluating the trace of the Hessian via second-order backpropagation through the denoising network, which is prohibitively expensive for large-scale video models. Remarkably, MotionCFG bypasses this entirely, offering an efficient proxy for Eq. (6) as formalized below.

Proposition 1. *Let $\boldsymbol{n}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ be the noise realization in Eq. (2). Assuming $\mathbf{S}_\theta(\boldsymbol{z}_t, \cdot)$ is three-times differentiable with respect to the condition, the expected guided score in Eq. (5) satisfies*

$$\mathbb{E}_{\boldsymbol{n}_t}\left[\tilde{\mathbf{S}}^*_\theta(\boldsymbol{z}_t, \boldsymbol{c}, \boldsymbol{c}_{pert,t})\right] \;=\; \mathbf{S}_\theta(\boldsymbol{z}_t, \boldsymbol{c}) \;-\; \frac{\omega\, \sigma_{\boldsymbol{c}}^2}{2}\, \nabla^2_{\boldsymbol{c}_m} \mathbf{S}_\theta(\boldsymbol{z}_t, \boldsymbol{c}) \;+\; O(\sigma_{\boldsymbol{c}}^3), \tag{7}$$

where $\nabla^2_{\boldsymbol{c}_m}$ is the Laplacian with respect to $\boldsymbol{c}_m$.

See Section B for the proof. Notice that Eq. (7) recovers the Semantic Laplacian Sharpening target in Eq. (6) with $\gamma = \omega\sigma_{\boldsymbol{c}}^2/2$, up to third-order residuals. This means that the MotionCFG update in Eq. (5) is interpretable as an unbiased estimator of the Laplacian-regularized score, requiring only one extra forward pass (*i.e*., the same cost as standard CFG).

While Proposition 1 holds in expectation under idealized assumptions, the practical MotionCFG update involves several approximations, including our use of single-sample Monte Carlo for $\boldsymbol{n}_t$ and the residual term $O(\sigma_{\boldsymbol{c}}^3)$. To assess whether the Laplacian sharpening interpretation remains meaningful in practice, we conduct an empirical analysis on synthetic toy data, finding that the qualitative sharpening behavior predicted by the framework is preserved across our practical design choices. See Section C.1 for details.

4 Experiments

Models and Datasets. To verify the robustness of our approach across various model scales, we conduct experiments on several state-of-the-art T2V frameworks, including Wan2.1-T2V (1.3B and 14B) and CogVideoX-5B. For quantitative and qualitative evaluation, we utilize the T2V-Compbench [30] dataset. However, as T2V-Compbench does not explicitly prioritize or categorize high-magnitude motion dynamics, we curate an additional set of motion-intensive prompts using Gemini to specifically assess dynamic fidelity. A comprehensive description of these curated prompts and their categories is provided in the supplementary.

Baselines. We compare MotionCFG against several training-free sampling strategies. *Baseline* denotes *Negative Prompting*, which replaces the null embedding in standard CFG with a text condition describing static scenes (*e.g*., "static, still picture"); details on prompt selection are provided in

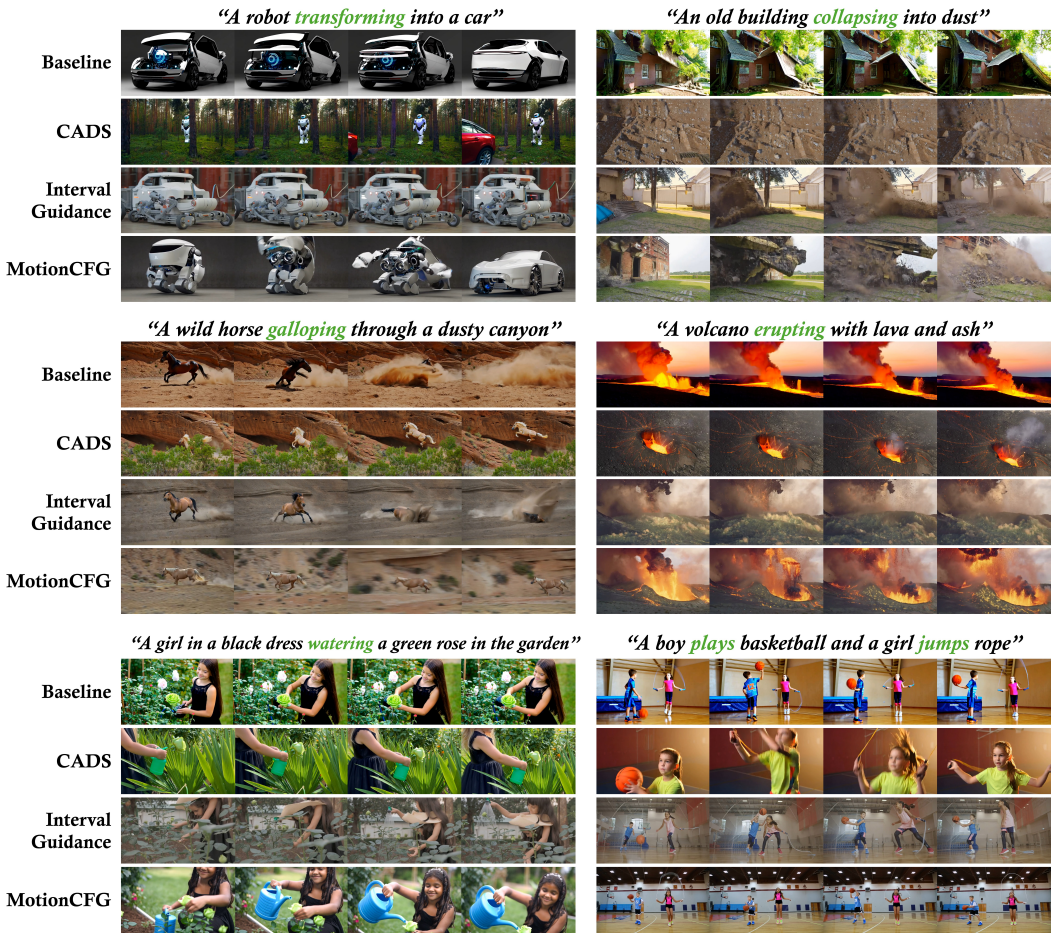


Figure 4: **Qualitative comparison on Wan2.1.** Each row shows uniformly sampled frames from a generated video. "Baseline" denotes negative-prompted CFG. While existing approaches produce near-static outputs or visual artifacts, MotionCFG generates physically plausible dynamics faithful to the highlighted motion tokens (green).

the supplementary. *CADS* [25] and *Interval Guidance* [17] are state-of-the-art diversity-enhancing samplers originally proposed for image generation. We additionally compare against *State Guidance (SG)* [28], which applies frame-wise scheduled guidance over state-transition prompts, and *VideoGuide* [18], which improves temporal consistency via teacher-guided denoising trajectory interpolation during sampling. We further include *FlowMo* [26], a concurrent training-free motion-enhancing approach that performs step-wise gradient-based optimization in the latent space to reduce temporal variance. Unlike MotionCFG, FlowMo requires per-step gradient computation, introducing substantial computational overhead; see Appendix C.3 for detailed analysis.

Evaluation Metrics. To comprehensively evaluate the generated videos, we employ a diverse set of metrics focusing on two main aspects: *visual quality with semantic alignment*, and *motion dynamics*. For visual quality and semantic alignment, we evaluate videos at both the frame and video levels. For frame-wise evaluation, we extract individual frames from the generated video, independently calculate **CLIPScore** [9] and **PickScore** [16], and report the averaged scores for each video. To assess the global spatio-temporal alignment between the text prompt and the entire video sequence, we employ **X-CLIP** [21], a dedicated video-language understanding metric.

To quantitatively evaluate the magnitude, complexity, and temporal dynamics of the generated motions, we adopt the evaluation protocol proposed in the **DEVIL** benchmark [19]. Specifically, we report **FLOW** (average pixel-level motion) and semantic variations using **Info DINO** and **DINO Segm**. We also report the unified **DEVIL Score**. Notably, this score is derived via regression after measuring a broader, comprehensive set of multiple dynamic indicators to provide a holistic

Table 2: **Quantitative comparison on Gemini-generated prompts**. "Baseline" denotes negative-prompted CFG. MotionCFG (Ours) consistently demonstrates superior performance in temporal dynamics across various base models. Upward arrows (↑) indicate higher is better, while downward arrows (↓) indicate lower is better.

Model	Method	Framewise		Full Video	Dynamics / Temporal				
		CLIPScore↑	PickScore↑	X-CLIP↑	SSIM↓	FLOW↑	Info DINO↑	DINO Segm.↑	DEVIL↑
Wan2.1-1.3B	Baseline	0.2535	0.2092	22.2128	0.7193	4.0962	0.0555	0.3842	0.3069
	CADS	0.2496	0.2104	22.7687	**0.6476**	2.2151	0.0294	0.3553	0.1973
	IG	0.2392	0.2058	23.0677	0.7116	3.7564	0.0635	0.4420	0.3147
	FlowMo	**0.2622**	**0.2120**	22.4004	0.7669	3.5846	0.0447	0.3558	0.3032
	VideoGuide	0.2554	0.2118	22.7258	0.6287	4.1330	0.0366	0.3780	0.2636
	SG	**0.2535**	**0.2089**	22.2330	0.7359	3.7479	0.0572	0.3853	0.3071
	Ours	0.2488	0.2089	**23.3881**	0.6547	**4.7320**	**0.0650**	**0.4586**	**0.3374**
Wan2.1-14B	Baseline	**0.2651**	0.2136	21.7438	0.8398	2.9466	0.0430	0.2853	0.2324
	CADS	0.2516	0.2112	22.1356	0.7735	2.4108	0.0467	0.3554	0.2410
	IG	0.2518	0.2112	**23.0622**	0.8207	2.6495	0.0571	0.3931	0.2732
	FlowMo	0.2626	**0.2146**	22.1167	**0.7561**	3.9944	0.0374	0.3507	0.2822
	Ours	0.2519	0.2098	22.7080	0.7901	**6.0557**	**0.0772**	**0.4293**	**0.3304**
CogVideoX-5B	Baseline	**0.2627**	**0.2140**	**23.4035**	0.8243	2.6547	0.0384	0.3109	0.2415
	CADS	0.2534	0.2116	22.7322	**0.7651**	2.9304	0.0494	**0.4470**	0.2884
	IG	0.2539	0.2105	22.6066	0.7665	**2.9594**	0.0512	0.4368	0.2955
	VideoGuide	0.2693	0.2176	22.9130	0.8717	1.5847	0.0216	0.2692	0.1774
	Ours	0.2550	0.2115	22.7820	0.7666	2.9415	**0.0522**	0.4360	**0.2976**

Table 3: **Quantitative comparison on T2V-CompBench**. "Baseline" indicates negative-prompted CFG. We observe that MotionCFG significantly improves temporal consistency and motion dynamics compared to existing guidance techniques. Upward arrows (↑) indicate higher is better, while downward arrows (↓) indicate lower is better.

Model	Method	Framewise		Full Video	Dynamics / Temporal				
		CLIPScore↑	PickScore↑	X-CLIP↑	SSIM↓	FLOW↑	Info DINO↑	DINO Segm.↑	DEVIL↑
Wan2.1-1.3B	Baseline	0.2651	0.2108	22.7719	0.7720	3.0671	0.0463	0.3517	0.2528
	CADS	**0.2654**	**0.2127**	**24.1668**	0.7018	1.7996	0.0348	0.3536	0.2022
	IG	0.2552	0.2090	23.9607	0.6785	4.1116	0.0601	0.4761	0.3077
	VideoGuide	0.2675	0.2113	23.3390	0.6581	4.1287	0.0459	0.3992	0.2585
	SG	0.2631	0.2088	22.4492	0.6581	4.1287	0.0459	0.3992	0.2579
	Ours	0.2550	0.2085	23.7080	**0.6378**	**6.5136**	**0.0726**	**0.5034**	**0.3531**
Wan2.1-14B	Baseline	**0.2755**	**0.2147**	23.1646	0.8770	1.7147	0.0378	0.2561	0.1995
	CADS	0.2630	0.2121	23.1093	0.7853	3.5325	0.0438	0.3464	0.2286
	IG	0.2579	0.2092	**23.2270**	0.8223	1.7266	0.0571	0.3893	0.2581
	Ours	0.2593	0.2097	23.0819	**0.8025**	**4.4839**	**0.0727**	**0.4213**	**0.3057**

assessment. Additionally, we employ **SSIM** to measure structural variation across adjacent frames, where a lower score indicates higher dynamic changes.

4.1 Main Results

Qualitative Evaluation. Figure 4 compares generated videos of MotionCFG against the baselines. Standard CFG with negative prompting ("Baseline") often produces nearly static frames, while CADS and Interval Guidance introduce artifacts such as over-saturation or subject morphing. In contrast, MotionCFG generates vigorous motion trajectories, such as the structural collapse of the building and the galloping motion of the horse, while preserving subject consistency and visual fidelity. Since static frames cannot fully convey these temporal improvements, we provide high-resolution video comparisons in the project page.

Quantitative Results. Tables 2 and 3 present quantitative comparisons across all base models and prompt sets. MotionCFG consistently achieves high scores in FLOW, DINO Segm., and DEVIL, confirming its effectiveness in enhancing temporal dynamics at multiple granularities. Notably, on Wan2.1-14B, MotionCFG more than doubles the FLOW score of the baseline (6.06 vs. 2.95) while maintaining comparable X-CLIP. CADS tends to suppress motion (lower FLOW than baseline in most settings), and Interval Guidance offers moderate gains but at the cost of reduced CLIPScore. In contrast, MotionCFG amplifies motion dynamics with only marginal trade-offs in framewise fidelity metrics, demonstrating a favorable balance between dynamics and visual quality.

Figure 5 visualizes the text fidelity–motion dynamics trade-off by sweeping each method's hyperparameters. Observe that MotionCFG occupies the upper-right region across all three dynamics metrics, achieving stronger motion with minimal X-CLIP degradation. Baseline and Interval Guidance cluster

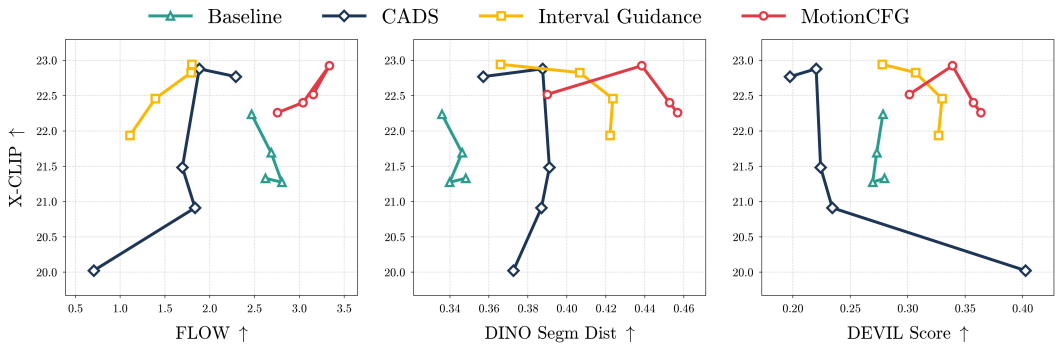


Figure 5: **Trade-off analysis between text fidelity and motion dynamics.** Each point corresponds to a different hyperparameter choice (*e.g.*, ratio τ). MotionCFG (red) achieves higher motion scores (FLOW, Dino Segm Dist, DEVIL) with minimal X-CLIP degradation, while baselines either sacrifice fidelity or fail to improve dynamics.

in a narrow range, indicating limited controllability over dynamics. CADS can reach high FLOW and DINO Segm Dist values, but only at a severe X-CLIP cost; its curve drops steeply as dynamics increase. In contrast, MotionCFG's Pareto frontier remains nearly flat, demonstrating that motion intensity can be scaled without sacrificing semantic fidelity.

4.2 Ablation Studies

Table 4 isolates the effect of the perturbation target in the text embedding space. Null- and negative-prompted baselines show limited dynamics, indicating that replacing the unconditional branch with a generic static negative prompt is insufficient to resolve motion ambiguity. Perturbing embeddings at non-motion token positions improves several dynamic metrics, but causes a noticeable drop in X-CLIP, suggesting that perturbations outside the motion subspace disrupt static semantics such as object identity or scene layout. In contrast, perturbing embeddings at motion-related token positions consistently improves dynamic metrics while maintaining competitive text alignment, yielding the strongest fidelity–dynamics trade-off. Additional ablations including LLM-refined positive/negative prompts, σ_c, τ, and other factors are provided in Section C.2.

Table 4: **Ablation on perturbation target.** We compare null/negative prompting with perturbations applied to text embeddings at non-motion or motion-related token positions. Perturbing motion-related embeddings yields the strongest dynamics while preserving competitive text alignment.

Method	X-CLIP ↑	I-DINO ↑	D-Segm. ↑	DEVIL ↑
Null-Prompted	**22.2696**	0.0597	0.3594	0.3083
Neg-Prompted	22.2128	0.0555	0.3842	0.3069
Non-motion ($\mathcal{I}^c_{\text{motion}}$)	21.9169	<u>0.0805</u>	<u>0.4208</u>	<u>0.3320</u>
Motion ($\mathcal{I}_{\text{motion}}$)	<u>22.2222</u>	**0.0887**	**0.4990**	**0.4001**

4.3 Extension to Object Counting

To demonstrate the versatility of our framework, we investigate whether the semantic perturbation mechanism of MotionCFG can be generalized beyond motion dynamics to other specific attributes, such as numerical quantity. In this experiment, we apply perturbations to the embeddings of quantity-related words instead of motion-related ones. We evaluate the object counting accuracy using the open-world counting framework proposed in CountVid [1], calculating the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) between the generated videos and ground truth counts.

Table 5: **Object counting performance.** MotionCFG configurations show lower MAE and RMSE.

Method	MAE ↓	RMSE ↓
Baseline ($w = 6.0$)	6.2543	8.3558
IG ($w = 6.0, r = 0.1$)	4.8911	12.0896
Ours ($\tau = 0.2, \sigma_c = 0.1$)	4.1411	11.5924
Ours ($\tau = 0.4, \sigma_c = 0.1$)	4.2056	9.1235
Ours ($\tau = 0.2, \sigma_c = 0.5$)	**2.3952**	**3.6290**

Prompt: *"Three horses gallop across the beach."*



(a) Interval Guidance (b) Ours

Figure 6: **Qualitative comparison of subject count consistency.** While the Interval Guidance (a) hallucinated an additional horse, our method (b) consistently maintained the requested count of three horses throughout the video.

As summarized in Table 5, MotionCFG demonstrates a clear advantage in numerical alignment compared to the baseline and Interval Guidance (IG). Even with preliminary hyperparameter settings, our method effectively reduces the counting error. For instance, the configuration with $w = 6.0, \tau = 0.2, \sigma_c = 0.5$ achieves an MAE of 2.3952, which is a significant improvement over the baseline MAE of 6.2543. These results suggest that perturbing attribute-specific embeddings during the guidance process can effectively steer the generative model toward precise attribute formation. The variance in performance across different noise scales indicates a need for attribute-specific scheduling. Since different semantic properties likely emerge at different stages of the diffusion process, future work should explore optimal temporal windows and noise intensities tailored to specific attributes beyond motion and quantity.

5 Conclusion

We introduced MotionCFG, a training-free guidance mechanism that addresses motion ambiguity in text-to-video (T2V) generation. By constructing motion-perturbed conditions and steering generation away from them, MotionCFG selectively sharpens motion semantics while leaving other attributes intact. We further showed that this procedure admits an interpretable theoretical viewpoint as Semantic Laplacian Sharpening, a second-order correction restricted to the motion subspace of the condition. Combined with a piecewise guidance schedule, MotionCFG consistently improves motion dynamics across state-of-the-art T2V frameworks with negligible overhead, and the same principle generalizes to other ambiguous semantic attributes such as precise object numerosity.

Limitations and Future Work. MotionCFG relies on an external LLM for automatic motion token identification, introducing a lightweight but non-negligible dependency; replacing it with attention-based saliency or similar attribution methods could further streamline the pipeline. Extending the framework to additional semantic axes and developing adaptive scheduling that automatically identifies the optimal intervention window are promising directions for future work.

Furthermore, our theoretical connection to Semantic Laplacian Sharpening holds in expectation, whereas practical guidance relies on a single stochastic perturbation at each denoising step. This approximation may introduce variability across samples, particularly for complex motion descriptions. Further investigation into variance reduction and more robust semantic conditioning could improve the consistency of motion generation.

Broader Impact. By improving motion controllability in text-to-video generation, MotionCFG can benefit content creation and visual simulation. However, as with any advance in generative video quality, there is a risk of misuse for producing deceptive synthetic media. We encourage responsible deployment following the safety guidelines of the underlying generative models.

Acknowledgments

This work was supported by the National Research Foundation of Korea under Grant RS-2024-00336454, the Advanced GPU Utilization Support Program funded by the Government of the Republic of Korea (Ministry of Science and ICT) (02-26-01-0404), the AI Computing Infrastructure Enhancement (GPU Rental Support) User Support Program funded by the Ministry of Science and ICT (MSIT) Republic of Korea (RQT-25-120217), and Artificial intelligence industrial convergence cluster development project funded by the Ministry of Science and ICT(MSIT, Korea) & Gwangju Metropolitan City. This work was also supported by the National Research Foundation of Korea (NRF) grants funded by the Korean government (MSIT) (No. RS-2025-25463760; No. RS-2026-25541956, contribution rate: 15%).

References

[1] N. Amini-Naieni and A. Zisserman. Open-world object counting in videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

[2] Vatsal Baherwani, Yixuan Ren, and Abhinav Shrivastava. Characterizing motion encoding in video diffusion timesteps. *arXiv preprint arXiv:2512.22175*, 2025.

[3] Lital Binyamin, Yoad Tewel, Hilit Segev, Eran Hirsch, Royi Rassin, and Gal Chechik. Make it count: Text-to-image generation with an accurate number of objects. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13242–13251, 2025.

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

[5] Barath Chandran. C, Srinivas Anumasa, and Dianbo Liu. Laplacian score sharpening for mitigating hallucination in diffusion models. *arXiv preprint arXiv:2511.07496*, 2025.

[6] 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.

[7] Alakh Desai and Nuno Vasconcelos. Improving image synthesis with diffusion-negative sampling. In *European Conference on Computer Vision*, pages 199–214. Springer, 2024.

[8] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[9] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 7514–7528, 2021.

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

[11] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[12] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 8633–8646. Curran Associates, Inc., 2022.

[13] Michael F Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18(3):1059–1076, 1989.

[14] Sangwon Jang, Taekyung Ki, Jaehyeong Jo, Jaehong Yoon, Soo Ye Kim, Zhe Lin, and Sung Ju Hwang. Frame guidance: Training-free guidance for frame-level control in video diffusion models. In *The Fourteenth International Conference on Learning Representations*, 2026.

[15] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[16] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[17] Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *Advances in Neural Information Processing Systems*, 37:122458–122483, 2024.

[18] Dohun Lee, Bryan Sangwoo Kim, Geon Yeong Park, and Jong Chul Ye. Videoguide: Improving video diffusion models without training through a teacher's guide. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2599–2608, 2025.

[19] Mingxiang Liao, Hannan Lu, Xinyu Zhang, Fang Wan, Tianyu Wang, Yuzhong Zhao, Wangmeng Zuo, Qixiang Ye, and Jingdong Wang. Evaluation of text-to-video generation models: A dynamics perspective. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[20] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

[21] Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. X-CLIP: end-to-end multi-grained contrastive learning for video-text retrieval. In João Magalhães, Alberto Del Bimbo, Shin'ichi Satoh, Nicu Sebe, Xavier Alameda-Pineda, Qin Jin, Vincent Oria, and Laura Toni, editors, *MM '22: The 30th ACM International Conference on Multimedia, Lisboa, Portugal, October 10 - 14, 2022*, pages 638–647. ACM, 2022.

[22] Liliane Momeni, Mathilde Caron, Arsha Nagrani, Andrew Zisserman, and Cordelia Schmid. Verbs in action: Improving verb understanding in video-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15579–15591, October 2023.

[23] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

[24] Penghui Ruan, Pichao Wang, Divya Saxena, Jiannong Cao, and Yuhui Shi. Enhancing motion in text-to-video generation with decomposed encoding and conditioning. *Advances in Neural Information Processing Systems*, 37:70101–70129, 2024.

[25] Seyedmorteza Sadat, Jakob Buhmann, Derek Bradley, Otmar Hilliges, and Romann M. Weber. CADS: Unleashing the diversity of diffusion models through condition-annealed sampling. In *The Twelfth International Conference on Learning Representations*, 2024.

[26] Ariel Shaulov, Itay Hazan, Lior Wolf, and Hila Chefer. Flowmo: Variance-based flow guidance for coherent motion in video generation. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025.

[27] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

[28] Konstantin Sobolev, Mikhail Zhirnov, Arsen Kuzhamuratov, Denis Dimitrov, Andrey Kuznetsov, and Anton Konushin. State & image guidance: Teaching old text-to-video diffusion models new tricks, 2025.

[29] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

[30] Kaiyue Sun, Kaiyi Huang, Xian Liu, Yue Wu, Zihan Xu, Zhenguo Li, and Xihui Liu. T2v-compbench: A comprehensive benchmark for compositional text-to-video generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 8406–8416, 2025.

[31] 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.

[32] Qwen: An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025.

[33] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025.

A Implementation Details

Hyperparameter Settings. For MotionCFG, we set the motion guidance scale to $\omega = 6.0$, the scheduling ratio to $\tau \in [0.1, 0.2]$, and the perturbation magnitude to $\sigma_c \in [0.1, 0.5]$. All approaches, including standard CFG, CADS, and Interval Guidance, share the same base guidance scale $\omega_{\text{std}} = 6.0$ for fair comparison.

Instruction Template for Motion Token Identification

```
messages = [
    {"role": "system", "content":
        "You are an AI that identifies ACTION VERBS in a sentence.\n"
        "Given indexed words like [i]word, output ONLY the indices of VERBS
            that describe physical actions or movements.\n"
        "DO NOT select: nouns, articles (a, an, the), pronouns (it, its, he,
            she), prepositions (in, on, at), adjectives.\n"
        "If no action verbs exist, return [].\n"
        "Output format: JSON list of indices only, nothing else."
    },
    # Few-shot: Explicit actions
    {"role": "user", "content": "Text: [0]A [1]cat [2]jumps [3]over [4]the
        [5]fence"},
    {"role": "assistant", "content": "[2]"},

    # Few-shot: Multiple actions
    {"role": "user", "content": "Text: [0]Birds [1]fly [2]and [3]sing [4]in
        [5]the [6]morning"},
    {"role": "assistant", "content": "[1, 3]"},

    # Few-shot: No actions (nouns only)
    {"role": "user", "content": "Text: [0]A [1]red [2]ball [3]on [4]the [5]
        table"},
    {"role": "assistant", "content": "[]"},

    # Few-shot: Exclude articles/pronouns
    {"role": "user", "content": "Text: [0]The [1]dog [2]sleeps [3]under [4]a
        [5]tree"},
    {"role": "assistant", "content": "[2]"},

    {"role": "user", "content": f"Text: {indexed_text}"}
]
```

Delayed Scheduling for CogVideoX. Unlike Wan2.1, CogVideoX exhibits severe artifacts when MotionCFG is applied from the very first denoising step. As shown in Figure 7, the Tweedie estimate at $t = T$ is dominated by dataset-specific biases (*e.g.*, POND5 watermarks) rather than meaningful semantic structure. Injecting motion perturbation at this stage may amplify these artifacts instead of enhancing dynamics. To address this, we employ a delayed schedule: the first few steps use only standard conditional prediction ($\omega_{\text{std}} = 1.0$) to establish a clean spatial layout, after which MotionCFG is applied over the subsequent interval. This simple modification successfully enhances motion dynamics while preventing early-stage artifact amplification.



Figure 7: **CogVideoX Tweedie estimate at $t = T$.** The initial prediction is dominated by watermark-like artifacts.

Motion Token Extraction Prompt. We use Qwen2.5-7B-Instruct to automate motion token identification. The input prompt is tokenized with positional indices (*e.g.*, `[0]A [1]cat [2]jumps`), and the model is instructed to return indices of physical action verbs via few-shot in-context learning. Since modern text encoders (*e.g.*, T5) often split a single word into multiple sub-word tokens, we

Table 6: **Examples from the Gemini-generated evaluation prompts.** Motion keywords are in **bold**.

Category	Text Prompt
Animal	A wild horse **galloping** through a dusty canyon
Animal	A great white shark **swimming aggressively** in the deep ocean
Human	A ballet dancer **spinning** on stage under a spotlight
Human	A chef **chopping onions rapidly** on a wooden board
Vehicle	A red sports car **drifting** on a race track with smoke
Vehicle	A space rocket **launching** into the sky with fire
Nature	A volcano **erupting** with lava and ash
Physics	An old building **collapsing** into dust
Abstract	Colorful ink **spreading** in clear water

locate the exact character span of each extracted verb and cross-reference it with the tokenizer's offset mapping. This yields the precise sub-word token indices, which form our active target set $\mathcal{I}_{motion}$ for the subsequent semantic perturbation. The full instruction template is shown in previous page.

Standard Negative Prompts. To encourage high-quality generation and heavily penalize low-quality or motionless artifacts, we employ a comprehensive standard negative prompt across all baselines. The exact text of the negative prompt used in our evaluations is detailed below.

Standard Negative Prompt

```
Bright tones, overexposed, static, blurred details, subtitles, style, works,
    paintings, images, static, overall gray, worst quality, low quality, JPEG
     compression residue, ugly, incomplete, extra fingers, poorly drawn hands
    , poorly drawn faces, deformed, disfigured, misshapen limbs, fused
    fingers, still picture, messy background, three legs, many people in the
    background, walking backwards
```

Gemini-Generated Evaluation Prompts. Existing T2V benchmarks such as T2V-CompBench primarily evaluate spatial composition and attribute binding, often lacking prompts that demand vigorous physical motion. Models can thus achieve high alignment scores with near-static outputs. To address this gap, we curated a motion-intensive prompt set of 100 prompts generated via Gemini, spanning categories including animal behaviors, human actions, vehicle dynamics, and natural phenomena. Each prompt is centered around a distinct action verb describing high-magnitude motion. Representative examples are shown in Table 6.

B Mathematical Derivation and Empirical Analysis

B.1 Proof of Proposition 1

Proposition 1. *Let $\boldsymbol{n}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ be the noise realization in Eq. (2). Assuming $\mathbf{S}_\theta(\boldsymbol{z}_t, \cdot)$ is three-times differentiable with respect to the condition, the expected guided score in Eq. (5) satisfies*

$$\mathbb{E}_{\boldsymbol{n}_t}\left[\tilde{\mathbf{S}}^*_\theta(\boldsymbol{z}_t, \boldsymbol{c}, \boldsymbol{c}_{pert,t})\right] \;=\; \mathbf{S}_\theta(\boldsymbol{z}_t, \boldsymbol{c}) \;-\; \frac{\omega\,\sigma_{\boldsymbol{c}}^2}{2}\,\nabla^2_{\boldsymbol{c}_m}\,\mathbf{S}_\theta(\boldsymbol{z}_t, \boldsymbol{c}) \;+\; O(\sigma_{\boldsymbol{c}}^3), \tag{8}$$

where $\nabla^2_{\boldsymbol{c}_m}$ is the Laplacian with respect to $\boldsymbol{c}_m$.

Proof. We first introduce a block decomposition of the text condition. Let $\boldsymbol{c} \in \mathbb{R}^{L\times d}$ denote the text embedding, where L is the number of tokens and d is the embedding dimension. Given the motion-token index set $\mathcal{I}_{\text{motion}}$, we partition the condition into a motion block and a static block:

$$\boldsymbol{c} = (\boldsymbol{c}_s, \boldsymbol{c}_m), \qquad \boldsymbol{c}_s \in \mathbb{R}^{d_s}, \quad \boldsymbol{c}_m \in \mathbb{R}^{d_m}, \tag{9}$$

where $\boldsymbol{c}_m$ collects the vectorized embedding entries corresponding to motion-related tokens and $\boldsymbol{c}_s$ contains the remaining static or contextual components. Under this notation, the perturbed condition in Eq. (2) is written as $\boldsymbol{c}_{\text{pert},t} = (\boldsymbol{c}_s, \boldsymbol{c}_m + \sigma_{\boldsymbol{c}} \boldsymbol{n}_t)$, leaving the static block untouched.

Next, we expand the perturbed score $\mathbf{S}_\theta(\boldsymbol{z}_t, \boldsymbol{c}_{\text{pert},t})$ around $\boldsymbol{c}$ along the motion subspace via a second-order Taylor expansion. For each score coordinate $k \in \{1, \dots, d_z\}$,

$$\begin{aligned} S_\theta^{(k)}(\boldsymbol{z}_t, \boldsymbol{c}_{\text{pert},t}) = S_\theta^{(k)}(\boldsymbol{z}_t, \boldsymbol{c}) + \sigma_{\boldsymbol{c}} \nabla_{\boldsymbol{c}_m} S_\theta^{(k)}(\boldsymbol{z}_t, \boldsymbol{c})^\top \boldsymbol{n}_t \\ + \frac{\sigma_{\boldsymbol{c}}^2}{2} \boldsymbol{n}_t^\top \nabla_{\boldsymbol{c}_m}^2 S_\theta^{(k)}(\boldsymbol{z}_t, \boldsymbol{c})\, \boldsymbol{n}_t + O(\sigma_{\boldsymbol{c}}^3), \end{aligned} \tag{10}$$

where $\nabla_{\boldsymbol{c}_m} S_\theta^{(k)} \in \mathbb{R}^{d_m}$ and $\nabla_{\boldsymbol{c}_m}^2 S_\theta^{(k)} \in \mathbb{R}^{d_m \times d_m}$ are the gradient and Hessian of $S_\theta^{(k)}$ with respect to the motion block. The smoothness assumption (three-times differentiability) ensures the validity of this expansion.

Substituting Eq. (10) into the MotionCFG update Eq. (5) yields

$$\begin{aligned} \tilde{S}_\theta^{*,(k)}(\boldsymbol{z}_t, \boldsymbol{c}, \boldsymbol{c}_{\text{pert},t}) = S_\theta^{(k)}(\boldsymbol{z}_t, \boldsymbol{c}) - \omega\sigma_{\boldsymbol{c}} \nabla_{\boldsymbol{c}_m} S_\theta^{(k)}(\boldsymbol{z}_t, \boldsymbol{c})^\top \boldsymbol{n}_t \\ - \frac{\omega\sigma_{\boldsymbol{c}}^2}{2} \boldsymbol{n}_t^\top \nabla_{\boldsymbol{c}_m}^2 S_\theta^{(k)}(\boldsymbol{z}_t, \boldsymbol{c})\, \boldsymbol{n}_t + O(\sigma_{\boldsymbol{c}}^3). \end{aligned} \tag{11}$$

We now take expectation over $\boldsymbol{n}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{d_m})$. The first-order term vanishes since $\mathbb{E}[\boldsymbol{n}_t] = \mathbf{0}$. For the second-order term, we apply Hutchinson's identity [13]: for any matrix $A \in \mathbb{R}^{d_m \times d_m}$,

$$\mathbb{E}_{\boldsymbol{n}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{d_m})} \left[\boldsymbol{n}_t^\top A \boldsymbol{n}_t \right] = \operatorname{Tr}(A). \tag{12}$$

Applying Eq. (12) with $A = \nabla_{\boldsymbol{c}_m}^2 S_\theta^{(k)}(\boldsymbol{z}_t, \boldsymbol{c})$ gives

$$\begin{aligned} \mathbb{E}_{\boldsymbol{n}_t} \left[\boldsymbol{n}_t^\top \nabla_{\boldsymbol{c}_m}^2 S_\theta^{(k)}(\boldsymbol{z}_t, \boldsymbol{c})\, \boldsymbol{n}_t \right] &= \operatorname{Tr}\Big(\nabla_{\boldsymbol{c}_m}^2 S_\theta^{(k)}(\boldsymbol{z}_t, \boldsymbol{c}) \Big) \\ &= \sum_{i=1}^{d_m} \frac{\partial^2 S_\theta^{(k)}(\boldsymbol{z}_t, \boldsymbol{c})}{\partial (\boldsymbol{c}_m)_i^2} \;=:\; \Delta_{\boldsymbol{c}_m} S_\theta^{(k)}(\boldsymbol{z}_t, \boldsymbol{c}), \end{aligned} \tag{13}$$

where the last equality defines the motion-condition Laplacian, recovering the standard scalar Laplacian operator restricted to the motion subspace.

Combining the above and applying the result component-wise to all d_z score coordinates,

$$\mathbb{E}_{\boldsymbol{n}_t} \left[\tilde{\mathbf{S}}_\theta^*(\boldsymbol{z}_t, \boldsymbol{c}, \boldsymbol{c}_{\text{pert},t}) \right] = \mathbf{S}_\theta(\boldsymbol{z}_t, \boldsymbol{c}) - \frac{\omega\sigma_{\boldsymbol{c}}^2}{2} \Delta_{\boldsymbol{c}_m} \mathbf{S}_\theta(\boldsymbol{z}_t, \boldsymbol{c}) + O(\sigma_{\boldsymbol{c}}^3). \tag{14}$$

Identifying $\nabla_{\boldsymbol{c}_m}^2 \mathbf{S}_\theta = \Delta_{\boldsymbol{c}_m} \mathbf{S}_\theta$ (the trace of the motion-block Hessian, as defined in Eq. (13)) completes the proof. □

B.2 Empirical Alignment with the Motion-Coordinate Laplacian

We complement Proposition 1 with two empirical analyses on Wan2.1-T2V-1.3B. First, we compare the MotionCFG correction with an independently sampled symmetric finite-difference estimate along the motion-token coordinates. Second, we examine the local motion-coordinate curvature encountered along CFG and MotionCFG sampling trajectories.

Setup and estimators. For a fixed latent, timestep, and condition $(\mathbf{z}_t, t, \mathbf{c})$, let $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, \mathbf{c})$ denote the model prediction used in this analysis. We define the Monte Carlo MotionCFG correction, before multiplication by the guidance weight ω, as

$$\mathbf{g}_{\text{MC}}(K) = \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, \mathbf{c}) - \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, \mathbf{c} + \sigma_{\mathbf{c}} \mathbf{v}_k). \tag{15}$$

Each direction $\mathbf{v}_k$ is standard Gaussian on the selected motion-token coordinates and zero elsewhere. Thus, $K = 1$ corresponds to the single-sample correction used by MotionCFG.

Table 7: **Alignment with the motion-coordinate Laplacian.** Agreement between the Monte Carlo MotionCFG correction $\mathbf{g}_{\mathrm{MC}}(K)$ and the independently sampled symmetric finite-difference estimate $\mathbf{g}_{\mathrm{Lap}}$. A norm ratio of one indicates matching magnitudes.

K	Cosine sim. ↑	Relative error ↓	Norm ratio
1	0.732	0.916	1.361
4	0.901	0.475	1.129
8	0.948	0.342	1.096
128	**0.992**	**0.122**	**1.004**

Table 8: **Motion-coordinate curvature along sampling.** Comparison of motion-coordinate finite-difference estimates along CFG and MotionCFG trajectories at matched denoising steps. Both branches share the initial latent, text condition, and probe directions.

Step	$\frac{\|\mathbf{g}^{\mathbf{M}}_{\mathbf{Lap}}\|_2}{\|\mathbf{g}^{\mathbf{CFG}}_{\mathbf{Lap}}\|_2}$	$\cos\left(\mathbf{g}^{\mathbf{M}}_{\mathbf{Lap}}, \mathbf{g}^{\mathbf{CFG}}_{\mathbf{Lap}}\right)$
0	1.000	1.000
5	**2.211**	0.079
10	2.034	0.014
35	1.236	0.014

Using an independent set of Gaussian directions $\{\mathbf{u}_j\}_{j=1}^{J}$ with the same support, we compute

$$\mathbf{g}_{\mathrm{Lap}} = -\frac{1}{2J}\sum_{j=1}^{J}\Big[\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, \mathbf{c} + \sigma_{\mathbf{c}}\mathbf{u}_j) - 2\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, \mathbf{c}) + \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, \mathbf{c} - \sigma_{\mathbf{c}}\mathbf{u}_j)\Big]. \tag{16}$$

The symmetric difference cancels the first-order term in each probe direction. Under the small-perturbation expansion used in Proposition 1,

$$\mathbb{E}[\mathbf{g}_{\mathrm{Lap}}] = -\frac{\sigma_{\mathbf{c}}^2}{2}\Delta_{\mathbf{c}_m}\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, \mathbf{c}) + O(\sigma_{\mathbf{c}}^3). \tag{17}$$

Thus, $\mathbf{g}_{\mathrm{Lap}}$ estimates the scaled negative motion-coordinate Laplacian appearing in the correction.

Gaussian symmetry gives $\mathbb{E}[\mathbf{g}_{\mathrm{MC}}(K)] = \mathbb{E}[\mathbf{g}_{\mathrm{Lap}}]$. This equality holds at finite perturbation scale, while the interpretation of their shared expectation as a scaled Laplacian relies on the small-perturbation expansion. Our comparison therefore evaluates agreement between two independently sampled estimators of the expected correction.

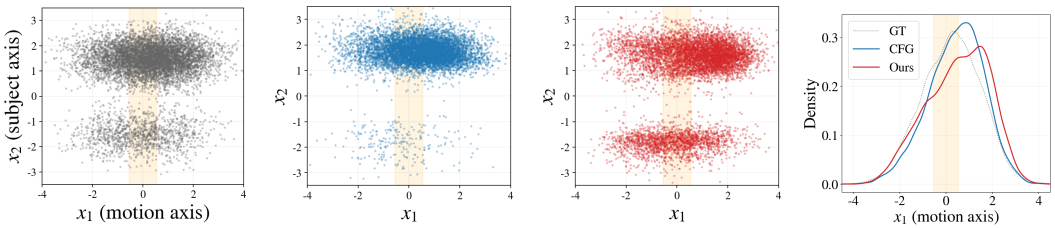
Results. Table 7 reports cosine similarity, relative error, and norm ratio between the two estimates. The single-sample correction is positively aligned with $\mathbf{g}_{\mathrm{Lap}}$, with a cosine similarity of 0.732, although its relative error remains 0.916. Agreement improves as K increases: at $K = 128$, cosine similarity reaches 0.992, relative error decreases to 0.122, and the norm ratio approaches unity at 1.004. These results demonstrate close agreement in both direction and magnitude after Monte Carlo averaging, while distinguishing this expected behavior from the variability of an individual correction.

B.3 Motion-Coordinate Curvature along Sampling Trajectories

We next investigate whether MotionCFG leads sampling through states with different local motion-coordinate curvature. Starting from the same initial latent and text condition, we run standard CFG and MotionCFG and evaluate Eq. (16) at matched denoising steps. We denote the resulting estimates by $\mathbf{g}^{\mathrm{CFG}}_{\mathrm{Lap}}$ and $\mathbf{g}^{\mathrm{M}}_{\mathrm{Lap}}$, respectively. At each matched step, we use the same Gaussian probe directions for both branches to control variation due to probe sampling.

As shown in Table 8, the estimates coincide at initialization and differ substantially after the trajectories diverge. At step 5, the MotionCFG estimate has 2.211 times the norm of the CFG estimate, with a cosine similarity of 0.079. The difference remains at step 35, beyond the MotionCFG guidance window, where the norm ratio is 1.236 and cosine similarity is 0.014.

These measurements show that MotionCFG changes the local motion-coordinate curvature encountered along the sampling trajectory. The model parameters remain fixed; the differences arise from evaluating the conditional prediction at different latent states. Together with the fixed-state analysis above, these results provide empirical evidence for the correction's finite-difference interpretation and its effect on the subsequent sampling trajectory.



(a) GT (b) CFG (c) MotionCFG (d) x_1 marginal

Figure 9: **Flow-matching toy experiment.** (a–c) 2D scatter plots of sampled distributions. The orange band marks the motion-ambiguous region around $x_1 = 0$. (b) Standard CFG concentrates samples toward one dominant mode but fails to reduce mass in the ambiguous region. (c) MotionCFG suppresses the ambiguous region while maintaining broader support along the motion axis. (d) Marginal density along the motion axis confirms that CFG (blue) leaves the ambiguous-region density comparable to GT (gray), while MotionCFG (red) clearly reduces it.

C Complementary Analyses and Discussions

C.1 Toy Experiment

We conduct a two-dimensional flow-matching toy experiment to clarify how MotionCFG differs from standard CFG in the perspective of Semantic Laplacian Sharpening.

Setup. We construct a conditional Gaussian mixture in $\mathbb{R}^2$, where x_1 is the motion axis and x_2 is the subject axis. A key design choice is the asymmetric mode separation: the motion modes are placed close together ($\mu_m = 0.8$) while the subject modes are placed farther apart ($\mu_s = 1.5$), with covariance $\mathbf{\Sigma} = \mathrm{diag}(1.0^2, 0.5^2)$. This reflects the structure of real T2V generation, where different motion interpretations (*e.g.*, a strong jump vs. a subtle hop) are closer in feature space than different object identities (*e.g.*, a cat vs. a dog), making the motion axis inherently harder to resolve. We further verify this asymmetry in the T5 encoder used by Wan2.1: as shown in Figure 8, subject-related tokens exhibit larger average cosine distances than motion-related tokens, confirming that motion semantics occupy a more compact region in the text-embedding space. The resulting toy distribution has two motion modes that strongly overlap around $x_1 = 0$, creating a pronounced motion-ambiguous region, while the subject modes remain well separated.

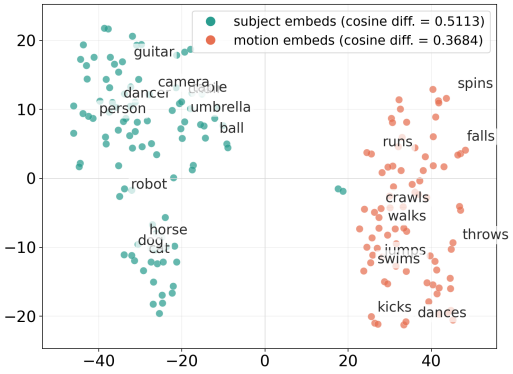


Figure 8: **T5 embedding geometry.** Subject tokens are more dispersed (avg. cosine distance 0.51) than motion tokens (0.37), confirming that motion semantics are more compact in the condition space.

For a condition (c_m, c_s), c_m controls the left/right motion preference and c_s controls the lower/upper subject preference. The four mode scores are defined as $q_1 = -c_m - c_s + 2c_m c_s$, $q_2 = -c_m + c_s - 2c_m c_s$, $q_3 = c_m - c_s - 2c_m c_s$, and $q_4 = c_m + c_s + 2c_m c_s$. The mixture weights are obtained by a softmax over these scores, $\pi_j = \exp(q_j)/\sum_{k=1}^{4} \exp(q_k)$, and the conditional distribution is $p(\mathbf{x} \mid c_m, c_s) = \sum_{j=1}^{4} \pi_j \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_j, \mathbf{\Sigma})$. In evaluation, we use a weak motion condition and a stronger subject condition, reflecting the case where subject identity is easier to resolve than motion.

We train a small conditional flow-matching model on this distribution. We sample $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\mathbf{x}_1$ from the target mixture, train the velocity field along the straight path $\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\mathbf{x}_1$, and generate samples by Euler integration. At sampling time, standard CFG extrapolates away from the global null condition, while MotionCFG perturbs only the motion condition.

Qualitative behavior. Figure 9 visualizes the sampled distributions. The GT conditional distribution has substantial overlap along the motion axis, while the subject axis remains clearly separated.

Table 10: **Effect of σ_c (fixed $\tau = 0.1$).** Increasing the scale provides scalable control over motion intensity.

σ_c	X-CLIP ↑	FLOW ↑	DEVIL ↑
0.01	23.0244	4.0177	0.3165
0.03	22.8113	4.7649	0.3291
0.05	**23.3881**	4.7320	0.3374
0.10	22.6914	**6.3951**	**0.3729**

Table 11: **Effect of Guidance Schedule (τ).** Moderate intervals improve dynamics, while excessive τ degrades fidelity and motion.

τ	X-CLIP ↑	FLOW ↑	DEVIL ↑
0.05	**22.9430**	**1.8005**	0.2779
0.10	22.8258	1.7918	0.3069
0.20	22.4572	1.3929	**0.3300**
0.30	21.9369	1.1102	0.3268

Standard CFG reduces some middle-region mass, but it does so by amplifying the entire condition against the global null branch. As a result, the samples become concentrated toward one dominant joint mode. This behavior is expected: CFG does not distinguish whether the amplified signal comes from motion or subject semantics, so a stronger subject signal can dominate the guidance direction.

MotionCFG behaves differently. Since the perturbed branch modifies only the motion condition, the resulting contrastive direction is localized to the under-resolved motion axis. The scatter plot shows that MotionCFG moves samples away from the ambiguous band around $x_1 = 0$ without collapsing as sharply as standard CFG. The histogram in Figure 9(d) shows the same effect after marginalizing out the subject axis. The density inside the orange band corresponds to motion-ambiguous samples; MotionCFG lowers this density more clearly than standard CFG, while maintaining broader support along the motion axis.

Quantitative trend. Table 9 reports M_{amb}, the fraction of samples in the motion-ambiguous band. Standard CFG only slightly reduces this value relative to the GT conditional distribution. MotionCFG provides a much stronger reduction, decreasing M_{amb} from 0.306 to 0.233 compared with standard CFG. Together with the scatter and histogram plots, this suggests that motion-local perturbation more directly suppresses uncertainty along the compact motion axis, whereas standard CFG performs a global sharpening that is more prone to single-mode concentration.

Table 9: **Ambiguous-motion mass.** M_{amb} is the fraction of samples inside the orange band around $x_1 = 0$. Lower is better.

Method	$\mathbf{M}_{\text{amb}} \downarrow$
GT	0.317
Conditional	0.322
Standard CFG	0.306
MotionCFG	**0.233**

C.2 Additional Ablations

Impact of Perturbation Strength (σ_c). We analyze the impact of the perturbation scale, denoted as σ_c, on the resulting motion dynamics. As summarized in Table 10, increasing σ_c significantly enhances the magnitude of generated motions, as indicated by the rise in FLOW and DEVIL Score. While higher σ_c values introduce stronger perturbations to the motion-related embeddings, the X-CLIP score remains remarkably stable. This suggests that σ_c serves as a reliable hyperparameter for scaling motion intensity while preserving the core semantic integrity.

Effect of Guidance Schedule (τ). We investigate the influence of the guidance schedule, controlled by the timestep ratio τ, which determines the interval over which the semantic perturbation is applied. As shown in Table 11, increasing τ from 0.05 to 0.20 notably enhances high-level semantic dynamics, improving the DEVIL Score from 0.2779 to 0.3300 and DINO Segm. from 0.3660 to 0.4236. However, extending the perturbation interval too far ($\tau = 0.30$) leads to a performance drop across most metrics. Notably, excessive τ causes a significant degradation in text alignment (X-CLIP drops to 21.9369) and reduces raw pixel-level motion (FLOW decreases from 1.8005 to 1.1102). This indicates that while an appropriate application window is essential for developing structural motion, injecting perturbations over an excessively long schedule corrupts the spatial semantics and ultimately limits overall dynamic fidelity.

Influence of LLM Size. A potential concern is MotionCFG's dependency on the accuracy of the LLM used for motion token extraction. To evaluate this, we ablate across the Qwen2.5 family (0.5B–7B) on our full evaluation set. As shown in Table 12, smaller models occasionally misidentify tokens, *e.g.*, Qwen2.5-0.5B with non-verbs such as "a" and "nearby". Despite such errors, MotionCFG still improves dynamics across all model sizes, with even the 0.5B variant outperforming the baseline (DEVIL: 0.319 vs. 0.279). Performance improves monotonically with LLM accuracy, and Qwen2.5-

Table 12: **Robustness to LLM size for motion token extraction.** The example column shows each model's extraction for the prompt: *"An elephant sprays water with its trunk, a lion sitting nearby."*

Method	Extraction Example	X-CLIP↑	DINO Segm.↑	Info DINO↑	DEVIL↑
Standard CFG	-	22.2369	0.3359	0.0568	0.2785
Qwen2.5-0.5B	"sprays", "a", "nearby"	22.5443	0.4114	0.0751	0.3186
Qwen2.5-3B	"sprays"	22.8157	0.4342	0.0751	0.3334
Qwen2.5-7B	**"sprays"**	**22.9234**	**0.4385**	**0.0765**	**0.3391**

Table 13: **Comparison with LLM-based prompt engineering on motion-related T2V-CompBench prompts.** Prompt-level rewriting can improve some motion metrics, but often degrades text-video alignment. MotionCFG achieves a stronger fidelity–dynamics trade-off by perturbing motion-related text embeddings directly.

Method	X-CLIP ↑	FLOW ↑	D-Segm. ↑	I-DINO ↑	DEVIL ↑
LLM-refined $\mathcal{P}_{\text{pos}}$	20.2686	1.9578	0.3664	0.0586	0.2547
LLM-generated $\mathcal{P}_{\text{neg}}$	17.4975	**2.5841**	0.4312	**0.0686**	**0.2847**
MotionCFG (Qwen2.5-7B)	<u>20.3305</u>	<u>2.3076</u>	<u>0.4802</u>	0.0595	0.2690
MotionCFG (Manual)	**21.1557**	2.2613	**0.4909**	<u>0.0624</u>	<u>0.2804</u>

7B achieves the best overall trade-off, confirming that MotionCFG is robust to imprecise extraction and does not require a large-scale LLM to be effective.

Comparison with LLM-based Prompt Engineering. We further compare MotionCFG with LLM-based prompt engineering on motion-related attribute prompts from T2V-CompBench. Specifically, we consider two prompt-level baselines: $\mathcal{P}_{\text{pos}}$, where an LLM rewrites the original prompt into a more motion-explicit positive prompt, and $\mathcal{P}_{\text{neg}}$, where an LLM generates an explicit negative prompt to discourage static or weak-motion outputs. As shown in Table 13, prompt-level rewriting can improve certain motion metrics, but often sacrifices text-video alignment. In contrast, MotionCFG keeps the original prompt fixed and perturbs only motion-related text embeddings, yielding a more favorable fidelity–dynamics trade-off.

Effect of Perturbation Locations. We compare two intervention locations: the input token embeddings before the T5 Transformer and the final contextual embeddings supplied to the video denoiser. We evaluate 50 prompts using paired initial seeds and identical settings ($\omega = 6.0$, $\tau = 0.2$, and $\sigma_{\mathbf{c}} = 0.1$); only the perturbation location changes.

As shown in Table 14, perturbing the final contextual embeddings yields higher FLOW, DINO Segm. Dist., Info-DINO, and DEVIL scores, together with lower SSIM. Perturbing the final embeddings acts directly on the contextual rows received by the denoiser and allows the original text representation to be cached and reused. In contrast, applying fresh perturbations to the initial input embeddings requires rerunning the text encoder at every active MotionCFG step. These results support the final contextual embeddings as the intervention location under the evaluated settings, with the additional computational benefit of avoiding repeated text-encoder evaluation.

C.3 Computational Overhead

Table 15 reports end-to-end sampling time per video on a single NVIDIA B200 GPU using Wan2.1-T2V-1.3B. Note that MotionCFG adds only $\sim$1 second over the baseline, as it requires no architectural modification, *i.e.*, introducing only a lightweight perturbation of text embeddings during the early denoising steps. The LLM inference for motion token extraction is negligible relative to the video diffusion process.

In contrast, FlowMo [26] performs step-wise gradient-based optimization in the latent space, resulting in signifi-

Table 15: **Complexity analysis.** Average end-to-end sampling time (sec/video). MotionCFG achieves superior temporal dynamics with negligible computational overhead.

Method	Time↓	X-CLIP↑	DEVIL↑
Baseline	76.339	22.2128	0.3069
CADS	76.710	22.7687	0.1973
IG	**73.305**	23.0677	0.3147
FlowMo	106.490	22.4004	0.3032
Ours	77.310	**23.3881**	**0.3374**

Table 14: **Ablation on perturbation location.** We compare perturbations applied to initial T5 input embeddings and final contextual T5 embeddings using 50 prompts with paired initial seeds and identical settings.

Perturbation location	X-CLIP ↑	SSIM ↓	FLOW ↑	D-Segm. ↑	I-DINO ↑	DEVIL ↑
Initial T5 input	**0.2354**	0.8003	2.2231	0.4259	0.0566	0.2646
Final T5 output	0.2314	**0.7518**	**6.8930**	**0.4570**	**0.0668**	**0.3192**

Table 16: **User study results.** Relative preference (%) and absolute success rates (%) of MotionCFG compared with baseline methods. Preference scores are obtained from 32 participants evaluating visual quality and text alignment (C1) and task-specific performance (C2/C3). Absolute correctness is evaluated by 36 additional participants on physical plausibility and action correctness. The both-fail columns report cases where neither method succeeds.

		Preference (%)		Physical Success (%)		Action Success (%)		Both Fail (%)	
Task	Comparison	C1	C2/C3	Other	Ours	Other	Ours	Physical	Action
Motion	vs. Baseline	**77.42**	**93.55**	31.11	**82.23**	35.55	**86.11**	13.33	9.44
	vs. CADS	**79.35**	**89.68**	22.22	**67.78**	28.33	**77.78**	20.00	10.00
	vs. IG	**85.16**	**81.94**	28.89	**69.45**	34.45	**73.89**	25.56	18.33
Counting	vs. Baseline	**77.42**	**69.68**	–	–	–	–	–	–

cantly higher inference time (e.g., $1.48\times$ slower than the baseline). Despite this minimal overhead, MotionCFG achieves the highest X-CLIP and DEVIL scores among all methods.

C.4 Failure Cases

While MotionCFG performs reliably under the demonstrated settings ($\sigma_c < 1.0$), extreme perturbation scales can cause generation failures. As shown in Figure 10, setting $\sigma_c = 3.0$ pushes $\boldsymbol{c}_{\text{pert},t}$ far from the valid semantic space, and extrapolating away from this out-of-distribution anchor drives the latent trajectory into degenerate regions. This manifests as extreme contrast, loss of fine-grained detail, and large black silhouetted patches. These results confirm that σ_c should be kept moderate (*e.g.*, $\sigma_c < 1.0$) to ensure the perturbed condition remains a meaningful contrastive signal rather than uninformative noise.



Figure 10: **Failure cases under extreme perturbation** ($\sigma_c = 3.0$). The perturbed condition falls far outside the valid semantic space, causing severe artifacts including oversaturation, loss of spatial detail, and black silhouetting.

D User Study

We conducted two complementary user studies to evaluate MotionCFG in terms of relative preference and absolute correctness. In the preference study, 32 participants evaluated 20 pairs of generated videos based on visual quality and text alignment (C1), and task-specific performance: motion dynamics (C2) or counting accuracy (C3). The presentation order was randomized to reduce ordering bias.

As shown in Table 16, MotionCFG is consistently preferred across all comparisons. For motion generation, MotionCFG achieves a 93.55% preference rate over the baseline in motion dynamics, while also outperforming CADS and IG. For counting, MotionCFG is preferred over the baseline in both visual quality and counting accuracy.

To further evaluate absolute correctness beyond relative preference, we conducted an additional study with 36 participants. Each comparison was classified into four mutually exclusive outcomes: both methods fail, only the compared method succeeds, only MotionCFG succeeds, or both methods succeed. Physical plausibility and action correctness were assessed separately, with absolute success defined as the sum of exclusive success and joint success.

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 state that MotionCFG is a training-free, inference-time guidance method for improving motion dynamics in text-to-video generation. The theoretical claim (Proposition 1 in Section 3.3) is supported by a formal proof (Section B) and empirical validation (Section C.1). The empirical claims are supported by quantitative comparisons across multiple models (Section 4.1), ablation studies (Section 4.2), and qualitative examples, while the extension to object counting (Section 4.3) is presented as an additional demonstration rather than the primary claim.

   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: Limitations and future work are discussed at the end of Section 5. Additional failure cases and sensitivity analyses are provided in Section C.

   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: Proposition 1 states its assumptions explicitly (three-times differentiability of the score function with respect to the condition). The complete proof is provided in Section B with detailed derivation steps. We also discuss the practical limitations of the asymptotic result, including single-sample Monte Carlo variance and higher-order residuals, and provide empirical validation on synthetic data (Section C.1).

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes].

   Justification: The paper specifies all hyperparameters (ω, σ_c, τ), base models (Wan2.1-1.3B/14B, CogVideoX-5B), prompt sets, baselines, and evaluation metrics. Implementation details including the LLM-based motion token extraction prompt, negative prompt templates, and computational overhead analysis are provided in Section 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: [No].

   Justification: The code and data are not publicly released at submission time to preserve anonymity. However, the method is training-free and requires no additional learned parameters. The paper provides full algorithmic details (Algorithm 1), hyperparameters, prompt construction protocol, and evaluation metrics sufficient to reproduce the main results. We plan to release the code upon acceptance.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes].

   Justification: The paper specifies the evaluated text-to-video models, prompt sets, baselines, metrics, guidance scales, perturbation strengths, and schedule ratios in Sections 4 and A. Since MotionCFG is training-free, no optimizer or training split is required for the proposed method.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No].

   Justification: The current version reports aggregate metrics over benchmark prompt sets but does not include confidence intervals, error bars, or statistical significance tests. We omit them due to the high computational cost of repeated large-scale video generation, and report multiple metrics, ablations, and model-scale comparisons as complementary evidence.

   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: Section C.3 reports average end-to-end sampling time per video and the GPU used for the complexity analysis. It also compares the computational overhead of MotionCFG against standard CFG and other inference-time baselines.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes].

   Justification: The work is an inference-time method evaluated on existing text-to-video models and benchmark or curated prompt sets. We do not collect private data or release a new high-risk pretrained model.

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: In Section 5, the paper discusses the positive impact of improving controllability and motion fidelity in text-to-video generation, and also acknowledges limitations including reliance on an external LLM and the need for adaptive scheduling across semantic axes.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A].

    Justification: The paper does not introduce or release a new pretrained image/video generator, scraped dataset, or large-scale model. Experiments are conducted using existing models, and responsible access is governed by the usage policies and safeguards of those underlying model providers.

    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 use publicly available pretrained models and benchmarks under their respective licenses and terms of use, including Apache 2.0-licensed Wan2.1 and CogVideoX code, and research-use licensed CogVideoX-5B weights, T2V-CompBench, CountVid, and Qwen2.5. We cite the original sources for all assets and do not redistribute proprietary datasets or model weights.

    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 paper introduces a curated motion-intensive prompt set generated to evaluate high-magnitude motion dynamics. Section A documents the prompt construction protocol, categories, and representative examples; no private or human-subject data are included.

    Guidelines:

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

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

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

Answer: [Yes].

Justification: We conducted a user study with 32 participants who evaluated 20 pairs of generated videos on visual quality and task-specific performance (Section D). Participation was voluntary and no personally identifiable information was collected. The evaluation details and settings are described in Section D.

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: [N/A].

Justification: The user study involved only preference judgments on generated videos, with no collection of personally identifiable information and no risk to participants. This type of evaluation is exempt from IRB review at our institution.

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: [Yes].

Justification: The paper explicitly describes the use of a lightweight LLM for motion token identification in the proposed pipeline. The supplementary material further evaluates robustness to different LLM sizes and extraction quality.

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.