PhysFlow: Physics-Intrinsic Velocity Regularization for Motion-Intensive Video Generation

Xianglong Guo[1,*] **Chang Yu**[1,*,†] **Haobo Xu**[2]
Junhao Ma[3] **Zhen Lei**[4,5,6]

[1] School of Intelligence Science and Technology, Nanjing University
[2] Tsinghua University
[3] Lingnan College, Sun Yat-sen University
[4] MAIS, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[5] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[6] CAIR, Hong Kong Institute of Science and Innovation, Chinese Academy of Sciences, Hong Kong, China
gxl2955839148@gmail.com

Abstract

Text-to-video models built on flow matching generate compelling videos for common prompts but degrade systematically under motion-intensive scenarios, producing artifacts such as object fragmentation, geometric deformation, and temporal flickering. Existing training-free approaches operate at the attention or guidance-scale level, neither of which directly addresses the velocity field that governs latent motion evolution. We observe that these motion artifacts stem from local misalignment between the predicted velocity and the latent's frame-axis temporal structure, a signal that can be diagnosed from quantities the sampler already computes. Based on this, we propose *PhysFlow*, a training-free velocity regularization method. PhysFlow constructs a per-frame latent flow residual to locate motion-inconsistent regions, steers the velocity to restore frame-axis alignment, and bounds the correction via residual-aware masking and safe clipping. We also introduce *MotionStress-100*, a five-category benchmark with a VLM-based protocol that isolates motion-intensive failures. On Wan2.1, PhysFlow improves the average MotionStress score over the baseline with *no additional inference cost* ($1.00\times$ cost), outperforming CFG-Zero* (-10.0%, $0.98\times$ cost) and FlowMo ($+0.7\%$, $2.12\times$ cost) by at least 4.7%, while improving VBench motion smoothness without loss of visual fidelity. Our code is publicly available at `https://github.com/SID-NJU/PhysFlow-repo`.

1 Introduction

Current text-to-video (T2V) models built on flow matching [22, 24] and diffusion [15, 35] frameworks can synthesize visually compelling videos for a wide range of everyday prompts. However, when the described motion becomes intensive, involving fast rigid-body dynamics, complex articulated movement, fine-grained mechanical interaction, generation quality degrades sharply and systematically [19, 4]. Typical failure modes include *structural fragmentation*, where objects break apart mid-motion; *geometric deformation*, where rigid bodies lose their shape as they move; *limb proliferation*, where articulated figures develop extra or severed limbs during fast actions; and *temporal discontinuity*, where fluids or particles flicker between frames rather than evolving continuously

[*] Xianglong Guo and Chang Yu contributed equally to this work.
[†] Corresponding author: Chang Yu.

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

(Figure 1). They appear consistently across state-of-the-art open-source models [37, 41, 21] whenever the motion complexity or intensity exceeds a model-dependent threshold that is encountered routinely in practice. Existing benchmarks such as VBench [16] aggregate quality scores across a broad prompt distribution and therefore dilute the signal from these motion-specific failures, making it difficult to diagnose *where* and *why* current models break.

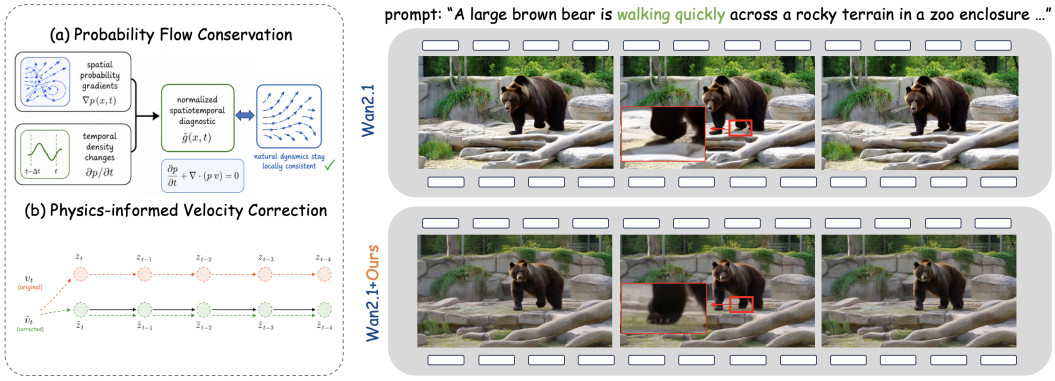

Figure 1: **PhysFlow corrects motion artifacts in T2V generation at inference time. Left**: Motivated by probability-flow conservation **(a)**, our method diagnoses local violations and corrects the sampler velocity $v_t \to \tilde{v}_t$ along the latent trajectory **(b)**. **Right**: under the same prompt and seed, Wan2.1 fragments the bear's leg (top); PhysFlow restores it (bottom) at $1.00\times$ cost.

A growing body of training-free methods attempts to improve T2V quality at inference time without modifying model weights, broadly categorized by the level at which they intervene. *Attention-level* approaches such as Enhance-A-Video [26] and ByTheWay [5] strengthen cross-frame temporal attention via temperature scaling or Fourier-domain energy amplification. While effective at improving frame-level consistency, temporal attention is a statistical coupling mechanism: it encourages frames to agree but carries no information about what direction of agreement is physically correct, so an object that deforms consistently across frames passes such checks undetected. *Guidance-level* approaches, including CFG-Zero* [10], STG [17], and NAG [7] modify the classifier-free guidance computation by adjusting the weighting between conditional and unconditional predictions, but they operate exclusively in the prompt-conditioning dimension of the velocity field. Mostly, they modulate how strongly the model follows the prompt, not whether the predicted motion is locally consistent with the latent's own temporal structure. Our experiments confirm this limitation: CFG-Zero* in fact degrades performance below the unmodified baseline on motion-intensive prompts (Table 1). *External-reward* approaches such as WMReward [42] use a large pretrained world model (VJEPA-2) to score physical plausibility and steer sampling, but require running a separate large-scale model during inference and sampling multiple candidate trajectories, incurring substantial overhead. In summary, no existing training-free method directly intervenes at the velocity-field level (shown in Figure 2), which is the layer that actually governs how the latent video evolves step by step. We use the term *physics-intrinsic* to describe diagnostic signals that arise from the latent video's own trajectory rather than from external physical priors or simulators.

Our work starts from a simple observation. In the flow matching framework, video generation reduces to integrating a learned velocity field via a numerical ODE solver, and every motion artifact in the final video originates from the velocity field pushing the latent in a locally inconsistent direction at some sampling step (see Appendix H). The latent video possesses an inherent inter-frame structure: consecutive frames differ smoothly because the underlying pixel-space motion is continuous and the video autoencoder preserves this continuity. When the predicted velocity for a given frame is well-aligned with this latent trajectory, the sampler transports the frame along a path that maintains structural integrity. When the alignment breaks and the velocity pushes a region in a direction that contradicts how that region relates to its temporal neighbors, the result is the fragmentation, deformation, and flickering described above. Crucially, this misalignment can be detected from quantities the sampler already computes: the per-frame velocity and the frame-to-frame difference together provide a local diagnostic signal without requiring any additional model evaluation.

Building on this insight, we propose PhysFlow, a training-free velocity regularization method that operates at the velocity-field level via a closed-form additive correction. PhysFlow constructs a

per-frame *latent flow residual* from the normalized alignment between the predicted velocity and the inter-frame derivative, converts the residual into a velocity update, and applies it through a residual-aware spatial mask, a magnitude bound, and an active schedule that defers to the original sampler during late refinement. The full method is detailed in Section 4. The entire pipeline involves only element-wise tensor operations on quantities the sampler already produces, adding no model forward passes and negligible wall-clock overhead.

We summarize our contributions as follows:

- We propose the *latent flow residual*, a lightweight, training-free diagnostic that detects local motion inconsistency between the predicted velocity field and the latent video's frame-axis structure. Based on this diagnostic, we design **PhysFlow**, a velocity-level regularization method that corrects motion-intensive generation artifacts at zero additional inference cost.
- We construct **MotionStress-100** (**MS-100**), a 100-prompt benchmark organized into five motion-stress categories (linear motion, articulated human motion, fluid/particle dynamics, camera motion, and fine-grained physics), together with a VLM-based evaluation protocol, to enable targeted assessment of motion-intensive generation quality. We also construct **MotionStress-100-Ext(MS-100-Ext)**, a prompt-extended variant where each MS-100 prompt is expanded by Wan2.1's official prompt-extension utility to test robustness under richer text guidance.
- Systematic experiments on Wan2.1 demonstrate that PhysFlow improves the average MotionStress score by +5.6% on MS-100 and +5.4% on MS-100-Ext without additional inference cost, outperforming FlowMo (+0.7%, $2.12\times$ cost) and CFG-Zero* (-10.0%, $0.98\times$ cost). On VBench, motion smoothness improves from 0.9824 to 0.9883 while preserving visual fidelity. Our method also scales to the larger 14B backbone, yielding a 1.9% improvement in the average MS-100-EXT score.

2 Related Work

Text-to-video generation and motion failures. Diffusion-based text-to-video (T2V) generation has progressed from cascaded UNet pipelines [3, 20, 39] to large-scale Diffusion Transformers [29] trained under the flow matching framework [22, 24]. Modern open-source T2V systems such as Wan2.1 [37], CogVideoX [41], and HunyuanVideo [21] produce visually compelling clips, but remain prone to motion-related failures including object fragmentation, articulated-limb violations, and unstable trajectories under fast or complex motion [19, 4].

Motion-aware control and training-free guidance. Existing methods for improving motion coherence fall into two regimes. **Training-time methods** introduce dedicated motion objectives or architectures (e.g., VideoJAM [6], motion-decoupled tokenization [18], motion-augmented training data [8], or temporal-aware backbones [12, 36]); conditioning-based variants instead inject external motion signals such as flow, trajectories, or pixel priors [11, 27, 23, 9], but require auxiliary inputs that may be unavailable at inference. **Training-free inference-time methods** are most directly comparable to ours: CFG-Zero* [10] modifies only the prompt-conditioning dimension and provides no signal about temporal coherence, while FlowMo [34] minimizes a patch-wise temporal-variance loss via gradient-based optimization on the latent, incurring a $\sim 2.1\times$ slowdown. Earlier methods such as FreeInit [40] and FreeNoise [31] target UNet-based DDPM samplers and transfer poorly to flow-matching DiTs [34]. Our method differs from these by being a closed-form velocity update with negligible overhead, residual-aware and self-localizing through a soft mask, and motivated by probability-flow conservation rather than variance minimization, while requiring no retraining, fine-tuning, auxiliary network, or external conditioning input.

Motion-focused evaluation. General-purpose T2V benchmarks such as VBench [16] and EvalCrafter [25] aggregate dozens of evaluation dimensions over broad prompt distributions, providing comprehensive but coarse signal on motion-stress robustness. We complement these with MotionStress-100, a targeted benchmark organized around five recurring motion-failure families, evaluated through a category-specific VLM-based question-answering protocol [1]; together with paired-seed protocols and human evaluation, this provides a finer-grained view of motion-related improvements that aggregate VBench scores can dilute.

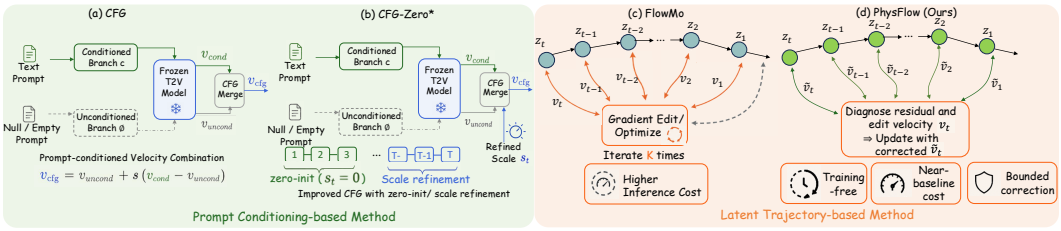

Figure 2: Comparison of training-free inference-time T2V correction methods. CFG and CFG-Zero* [10] act in the prompt-conditioning dimension; FlowMo [34] optimizes the latent at higher cost; PhysFlow (ours) applies a closed-form, bounded velocity correction at near-baseline cost.

3 Preliminaries

We work with text-to-video (T2V) models trained under flow matching [22, 24], the standard alternative to denoising diffusion in modern large-scale video generators [37, 30].

Flow matching. Let $z_0 \sim p_{\text{data}}(\cdot \mid c)$ be a clean latent video conditioned on a text prompt c, and $z_1 \sim \mathcal{N}(0, I)$ a Gaussian prior latent. Flow matching trains a network $v_\theta(z_\tau, \tau, c)$ to regress the target velocity $z_1 - z_0$ along the linear interpolation $z_\tau = (1-\tau)\, z_0 + \tau\, z_1$ for $\tau \in [0, 1]$. At inference, generation reduces to integrating the learned velocity field via a numerical ODE solver such as UniPC [44]:

$$\frac{dz_\tau}{d\tau} \;=\; v_\theta(z_\tau, \tau, c), \tag{1}$$

starting from z_1 and producing z_0 at $\tau = 0$. At each step the solver typically uses a classifier-free guided velocity [14], which we denote by v_t following the diffusion-sampling convention of using t for the discretized step index. PhysFlow operates entirely on v_t; the base T2V model, text encoder, VAE, and solver are all frozen.

4 Method

4.1 Overview

PhysFlow regularizes the guided velocity v_t at every sampling step via a closed-form additive correction:

$$\tilde{v}_t \;=\; v_t + \Delta v_t, \qquad \Delta v_t \;=\; \mathbb{1}[i < \rho_{\text{end}} N] \cdot \text{clip}_\tau\big(M \odot (\Delta v_{\text{diag}} + \Delta v_{\text{str}})\big), \tag{2}$$

where i and N denote the current and total sampling steps, where Δv_{diag} is a first-order *diagnostic correction* that restores local trajectory consistency, Δv_{str} is a complementary *straightness prior*, M is a *residual mask* that focuses the correction on inconsistent regions, clip_τ is a *safe clipping* bound, and $\mathbb{1}[i < \rho_{\text{end}} N]$ disables the correction during late refinement. The corrected $\tilde{v}_t$ is passed unchanged to the original solver, so PhysFlow is a strict drop-in replacement at the velocity-field level, and all quantities are computed from z_t and v_t alone — no extra model evaluation is required. The next two subsections expand Δv_{diag} and Δv_{str} (Section 4.2), and the residual-aware bounded application combining M, clip_τ, and the active schedule (Section 4.3). Figure 3 summarizes the pipeline.

4.2 Diagnostic correction and straightness prior

Trajectory consistency criterion. We first need a quantity that flags where v_t violates local trajectory consistency. Since well-formed video latents vary smoothly along the frame index [20, 39], the forward temporal difference $d_f = z_t^{f+1} - z_t^f$ is a meaningful inter-frame derivative. The sampler transports each frame along v_t^f, and local consistency requires that this transport be geometrically tied to how the latent already differs between frames f and $f+1$ — otherwise the sampler is moving frames in directions the latent's own trajectory cannot accommodate, producing the fragmentation, deformation, and flickering described in Section 1. We measure the alignment via a normalized *trajectory-aligned direction*

$$\hat{g}_f \;=\; \frac{v_t^f / D_f}{\| v_t^f / D_f \|_2 + \epsilon}, \qquad D_f \;=\; \begin{cases} -d_f, & |d_f| \geq \epsilon, \\ \epsilon \cdot \text{sgn}(-d_f + \epsilon), & \text{otherwise}, \end{cases} \tag{3}$$

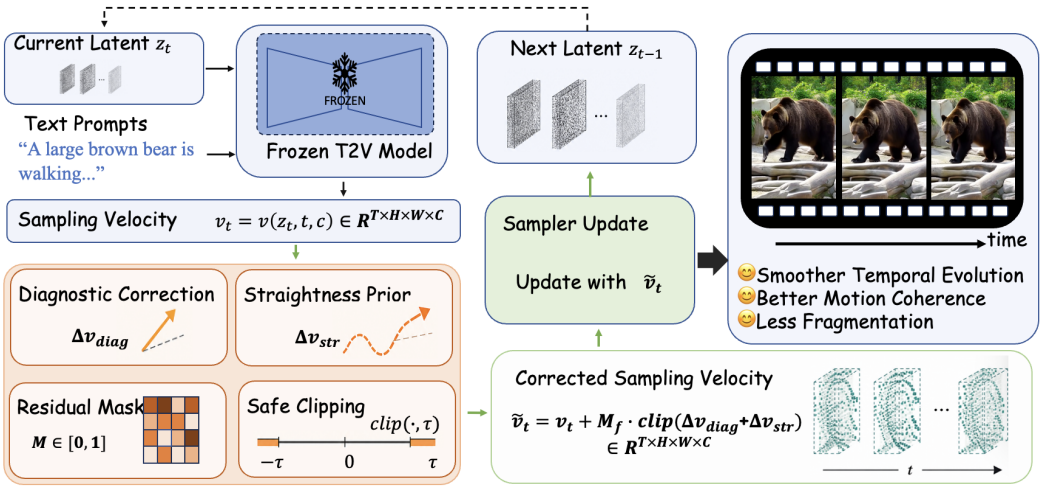

Figure 3: Overview of PhysFlow. A frozen T2V model produces the sampling velocity v_t from latent z_t and prompt c. Four components act on v_t: a *diagnostic correction*, a *straightness prior*, a *residual mask*, and *safe clipping*. The corrected $\tilde{v}_t$ is passed to the original sampler.

with the ℓ_2 normalization along the channel dimension; the piecewise D_f avoids division by near-zero values while preserving sign. Pairing v_t^f with $\hat{g}_f$ and taking unity as a scale-normalized reference gives the *latent flow residual*

$$r_f \;=\; \langle v_t^f,\, \hat{g}_f \rangle - 1 \;\in\; \mathbb{R}^{H\times W}, \tag{4}$$

where the inner product is over the channel dimension; small $|r_f|$ indicates compatible transport, large $|r_f|$ flags a local inconsistency that empirically correlates with the artifacts of Section 1. This formulation adapts the probability-flow view of NSG-VD [43] from post-hoc detection on completed videos to online generation-time regularization.

Diagnostic correction. Pushing v_t^f along $\hat{g}_f$ by the amount needed to drive the residual to zero gives the first-order update

$$\Delta v^f_{\text{diag}} \;=\; -\alpha\, r_f\, \hat{g}_f, \tag{5}$$

with α the regularization strength and the negative sign pulling $\langle v_t^f, \hat{g}_f \rangle$ back toward unity.

Straightness prior. A complementary failure mode is *abrupt curvature* in the latent trajectory itself: a frame whose latent jumps far from the linear interpolation of its neighbors typically decodes into flickering or sudden deformation — the discrete latent-space analog of perceptual straightness in natural video representations [13]. We damp it with a second-order term

$$\Delta v^f_{\text{str}} \;=\; -\beta\, \kappa_f, \qquad \kappa_f \;=\; z_t^{f+1} - 2z_t^f + z_t^{f-1}. \tag{6}$$

Boundary frames are zero-padded so that $\Delta v_{\text{diag}}, \Delta v_{\text{str}} \in \mathbb{R}^{C\times F\times H\times W}$. Ablations (Section 5.3) confirm that removing Δv_{str} costs less than removing Δv_{diag}, so PhysFlow is not merely temporal smoothing.

4.3 Residual-aware bounded application

Applying $\Delta v_{\text{diag}} + \Delta v_{\text{str}}$ as-is creates two practical risks: trajectory inconsistency is typically spatially localized, so a uniform correction over-modifies already-good regions; and large residuals can produce updates that overwhelm the pretrained velocity and damage appearance. We address these with a residual mask, safe clipping, and an active schedule, controlling *where*, *how much*, and *when* respectively. The mask is the residual magnitude itself, normalized to $[0, 1]$:

$$M_f \;=\; \frac{|r_f|}{\max_{f',h,w} |r_{f'}| + \epsilon}, \qquad \Delta v_t \;=\; \mathbb{1}[i < \rho_{\text{end}} N] \cdot \Delta v_{\text{raw}} \cdot \min\Big(1, \tfrac{\tau}{\|\Delta v_{\text{raw}}\|_\infty}\Big), \tag{7}$$

where $\Delta v_{\text{raw}} = M \odot (\Delta v_{\text{diag}} + \Delta v_{\text{str}})$, $M_F := M_{F-1}$ for the last frame and broadcast along the channel dimension, and i is the current step out of N. The mask makes the correction self-localizing: locations with larger residuals receive stronger correction, well-aligned regions are barely touched. The threshold τ caps the per-step magnitude — too small underrepairs motion failures, too large alters appearance and reduces prompt fidelity. The schedule uses $\rho_{\text{end}} = 0.7$ throughout: early-to-middle steps form global layout and coarse motion structure, while later steps refine local appearance [2], so disabling PhysFlow for the final 30% of steps lets the original sampler preserve fine-grained details unaffected. Empirical validation that the residual mask spatially localizes the correction and that the active schedule operates as designed is provided in Appendix H.

5 Experiments

We conduct qualitative and quantitative experiments to demonstrate the effectiveness of PhysFlow. Our experiments evaluate the gain in motion coherence enabled by our method, as well as its ability to maintain or improve other aspects of generation, including visual quality and frame-level consistency.

Implementation details. We employ Wan2.1-T2V-1.3B [37] as the backbone, using its officially provided weights and default sampler. As detailed in Section 4.3, our correction is active during the first 70% of denoising steps to establish global motion structure, then disabled for the late steps to let the model refine fine-grained appearance. We use a single fixed set of hyperparameters across all prompts and benchmarks; full details are in Appendix C.

Benchmarks. We evaluate on MotionStress-100, our newly constructed benchmark of 100 prompts evenly distributed across five motion-failure families that general-purpose benchmarks tend to dilute (linear motion, human/complex motion, fluid and particle dynamics, camera motion, fine-grained physics), and MotionStress-100-Ext, where each prompt is expanded by Wan2.1's official Qwen2.5-7B-Instruct utility [32] to test whether motion-stress failures persist under richer text guidance. For general video quality, we use VBench-1 and VBench-1-Ext [16], reporting four dimensions tied to our intervention: temporal flickering, motion smoothness, subject consistency, and scene. Construction details and protocols are in Appendix A and G.

Baselines. We compare against the original Wan2.1 sampler and two recent training-free inference-time methods. CFG-Zero* [10] introduces an optimized scale and a zero-init step but operates only in the prompt-conditioning dimension. FlowMo [34] minimizes a patch-wise temporal-variance loss via gradient-based optimization on the latent at refinement steps. Hyperparameter listings and a baseline configuration sweep are in Appendix D.

Evaluation metrics. On MS-100, our primary metric is a VLM-based MotionStress QA score: each video is judged by Qwen2.5-VL-7B-Instruct [1] on category-specific yes/no questions, averaged across the five categories. We complement this with two auxiliary frame-level metrics, CLIP-TC [33] and DINO-TC [28], computed as the average cosine similarity between consecutive frame embeddings; we treat them as auxiliary since frame-similarity scores can favor overly static videos. On VBench, we report the four dimensions above using the official pipeline. We further conduct a human preference study on paired-seed MS-100-Ext videos (motion coherence, visual detail, text-prompt alignment) and report per-video inference time on a single A100. Full details are in Appendix A and E.

5.1 Main Results on Motion-Intensive Generation

Qualitative comparison. Figure 4 compares Wan2.1-1.3B with and without PhysFlow under identical prompts and seeds, across all five motion-stress categories. The baseline frequently exhibits object fragmentation, unstable geometry, and temporal flickering; PhysFlow mitigates these artifacts while preserving scene content, with better object structure, more coherent articulated motion, and smoother fluid transitions. These improvements indicate that local motion inconsistency in the latent trajectory is a strong predictor of perceptible artifacts in pixel space.

Quantitative comparison. Table 1 summarizes the MotionStress QA scores across five motion-stress categories. Our method achieves the best average on both settings, improving the Wan2.1 baseline by $+3.73$ on MS-100 ($66.94 \rightarrow 70.67$) and $+3.78$ on MS-100-Ext ($69.61 \rightarrow 73.39$), with the largest gain on fine-grained physics, the category most directly tied to the local motion-consistency failures our diagnostic targets. Compared to recent training-free baselines, CFG-Zero* underperforms

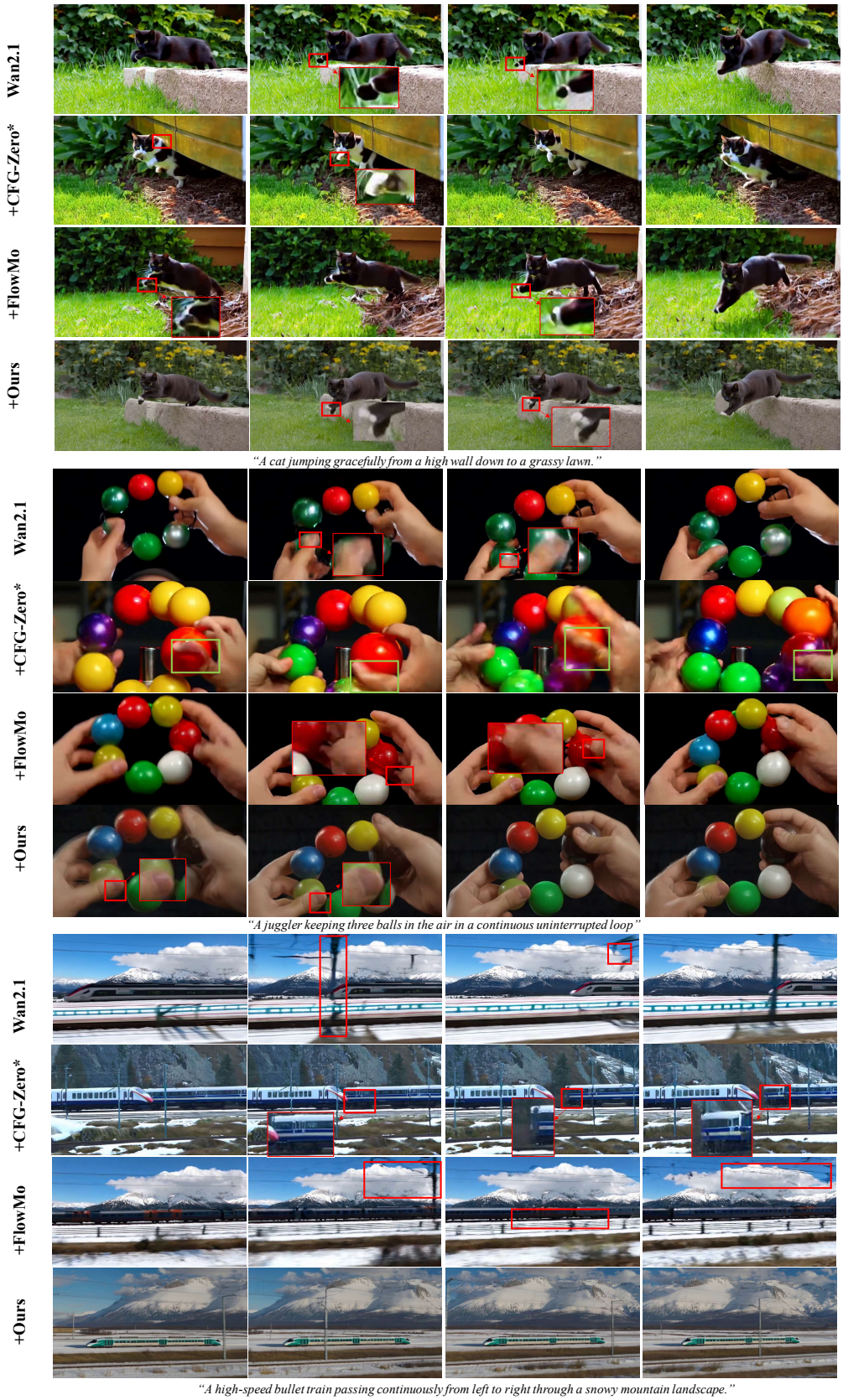

Figure 4: Qualitative comparisons on representative MotionStress prompts. Our method preserves object structure and produces more coherent temporal evolution under identical prompts and seeds. Additional examples are in the Appendix I.

Table 1: MotionStress QA performance and inference cost. **Bold** = best; underline = second best.

Setting	Method	MotionStress QA Performance						Inference Cost	
		Linear ↑	Human ↑	Fluid ↑	Camera ↑	Fine ↑	Avg. ↑	Time (s) ↓	Relative ↓
MS-100	Wan2.1 [37]	65.56	60.00	93.33	58.33	57.50	66.94	78.49	1.00×
	CFG-Zero*	61.67	44.44	83.33	**60.56**	65.00	63.00	77.06	0.98×
	FlowMo	62.78	56.11	90.00	60.00	68.33	67.44	166.37	2.12×
	Ours	**66.11**	**63.33**	**94.16**	58.89	**70.83**	**70.67**	78.53	1.00×
MS-100-Ext	Wan2.1 [37]	63.33	75.00	87.50	62.22	60.00	69.61	78.56	1.00×
	CFG-Zero*	60.56	70.56	89.16	60.56	65.00	63.33	76.89	0.98×
	FlowMo	62.22	75.00	91.66	62.78	61.67	70.67	165.43	2.11×
	Ours	**63.88**	**80.00**	**93.33**	**63.88**	**65.83**	**73.39**	78.64	1.00×

the baseline on average, while FlowMo partially closes the gap but remains below our method on both settings and incurs a 2.12× inference slowdown. In contrast, our method achieves the best MotionStress QA performance while maintaining near-baseline inference cost.

User study. Because VLM-based judges may exhibit systematic biases, we conduct a human preference study on a random subset of paired-seed videos from MS-100-EXT. For each prompt, raters were shown the baseline and our method side-by-side under identical prompts and seeds, and asked to compare them along three axes: motion coherence, visual detail, and text-prompt alignment. The results in Figure 6 show a consistent preference for our method across all three axes. These results are consistent with the VLM-based MotionStress judgments and indicate that our gains are perceptible to human viewers. Full instructions, rater demographics, and example pairs are provided in Appendix E.

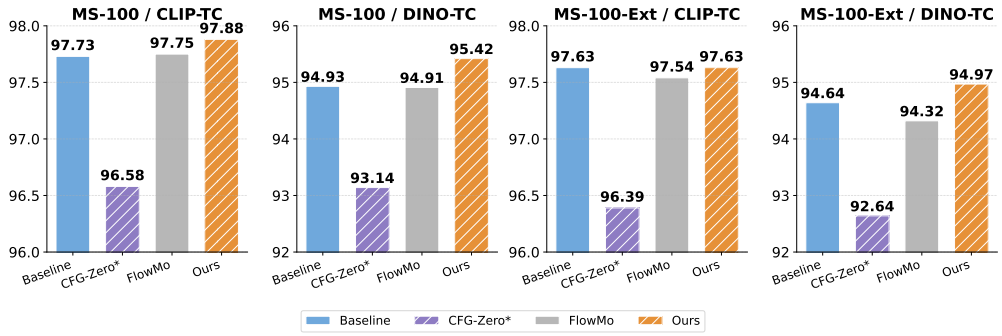

Figure 5: Auxiliary temporal consistency metrics on MotionStress-100 and MotionStress-100-Ext. Higher is better. Our method matches or exceeds the best baseline across all four panels.

Figure 5 reports auxiliary temporal consistency metrics across the four method–benchmark combinations. Our method matches or exceeds all baselines on CLIP-TC and DINO-TC for both MS-100 and MS-100-Ext, while CFG-Zero* drops below the baseline on both metrics, suggesting its prompt-conditioning-only correction can disrupt frame-level temporal structure under motion-stress prompts. We treat these as auxiliary rather than primary, since frame-level consistency scores can favor overly static videos.

Generalization Across Video Backbones. To evaluate the generalizability of PhysFlow, we extend our experiments beyond Wan2.1 to HunyuanVideo-1.5 [38] and Wan2.2-TI2V-5B [37]. We apply the same velocity regularization strategy with fixed hyperparameters, without backbone-specific retuning. As shown in Table 2, PhysFlow consistently improves the MotionStress QA scores across different model architectures, demonstrating its transferability as a training-free inference-time method.

5.2 Evaluation on VBench

For general video quality, we report four VBench dimensions most relevant to our intervention: *temporal flickering* and *motion smoothness* as the two dimensions most directly tied to motion

Table 2: Generalization of PhysFlow across video backbones. All results are MotionStress QA scores (%). Δ denotes the absolute improvement in percentage points.

Backbone / Benchmark	Baseline	PhysFlow	Δ
Wan2.1 (MS-100)	66.94	70.67	+3.73
Wan2.1 (MS-100-Ext)	69.61	73.39	+3.78
HunyuanVideo-1.5	70.44	72.28	+1.84
Wan2.2-TI2V-5B	64.61	69.06	+4.45

coherence, *subject consistency* to check identity preservation, and *scene* as a content-preservation check. The full sixteen-dimensional VBench-1 evaluation under a paired-seed protocol is reported in Appendix N; see Appendix G for a discussion of dimension selection.

Table 3: Selected VBench dimensions. Higher is better. **Bold** = best per setting. The paired-seed protocol fixes seeds $\{42, 43, 44, 45, 46\}$ between baseline and ours.

Setting	Method	Temp. Flicker ↑	Motion Smooth. ↑	Subject Cons. ↑	Scene ↑
VBench-1	Baseline	0.9913	0.9824	**0.9557**	0.2192
	Ours	**0.9914**	**0.9883**	0.9530	**0.2404**
VBench-1-Ext	Baseline	0.9916	0.9807	0.9309	0.5301
	Ours	**0.9924**	**0.9864**	**0.9347**	**0.5648**
VBench-1-Ext (+paired-seed)	Baseline	0.9909	0.9798	0.9333	0.5434
	Ours	**0.9918**	**0.9855**	**0.9351**	**0.5725**

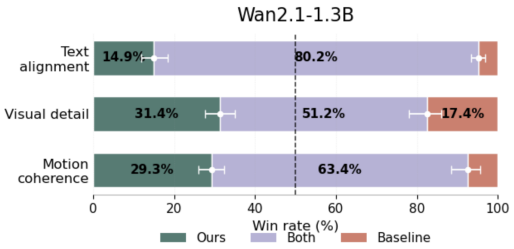

Figure 6: User study. Win rates are computed over paired-seed comparisons. Error bars indicate 95% confidence intervals.

Table 3 reports VBench results under standard stochastic generation (VBench-1 and VBench-1-Ext) and a paired-seed protocol on VBench-1-Ext. Our method improves temporal flickering and motion smoothness across all three settings; on VBench-1, the small subject-consistency change (-0.0027) is more than offset by a $+0.0212$ increase in scene, while on VBench-1-Ext all four dimensions improve. The paired-seed setting confirms these gains reflect our correction rather than seed-level randomness, and overall the VLM-based MotionStress, human preference, and VBench evaluations consistently show that PhysFlow improves motion coherence while preserving general video quality.

5.3 Ablation study

We ablate the main design choices of our method on MotionStress-100 without prompt expansion, isolating the effect of each correction term and the safe-clipping threshold τ.

Table 4: Component ablation on MotionStress-100. "small/large clip" refers to smaller/larger values of the safe threshold τ. Higher is better for all metrics. **Bold** = best; underline = second best.

Method	MotionStress QA Performance						Auxiliary Temporal Consistency	
	Linear ↑	Human ↑	Fluid ↑	Camera ↑	Fine ↑	Avg. ↑	CLIP-TC ↑	DINO-TC ↑
Baseline	65.56	60.00	93.33	58.33	57.50	66.94	97.73	94.93
Full (default)	66.11	**63.33**	**94.16**	58.89	**70.83**	**70.67**	**97.88**	95.42
Full w/ small clip	63.89	57.22	93.33	60.56	67.50	68.50	97.65	94.91
Full w/ large clip	**67.22**	57.22	90.83	**66.67**	70.00	70.39	97.78	**95.47**
w/o straightness	**67.22**	55.55	90.00	58.89	65.00	67.33	97.62	94.68
w/o diagnostic	65.00	56.11	86.67	58.89	60.83	65.50	97.63	95.41

Results in Table 4 show that the full method achieves the best average MotionStress score ($66.94 \rightarrow 70.67$) while also improving auxiliary temporal consistency over the baseline. Removing the diagnostic term drops performance *below* the baseline (65.50), while removing only the straightness prior gives a much smaller drop (67.33), confirming that the residual-driven steering is the main driver and the straightness prior plays a stabilizing role. Smaller and larger values of τ both work but underperform the default, indicating a meaningful trade-off between correction strength and visual preservation.

Validity of the residual signal. The contrast between the two single-term ablations also indicates that r_f carries motion-specific information rather than being a heuristic quantity: if it were uninformative,

the diagnostic-only variant ($\beta = 0$) would behave as additive noise and degrade the baseline; instead, it *improves* it (67.33), whereas the straightness-only variant ($\alpha = 0$, no residual signal) drops below baseline (65.50).

6 Discussion and Conclusion

We presented PhysFlow, a training-free velocity regularization method for motion-intensive video generation. PhysFlow diagnoses local trajectory inconsistency from a normalized residual between the sampler velocity and the latent's inter-frame derivative, and converts it into a closed-form additive correction bounded by a self-localizing mask, a magnitude clip, and an active schedule. Paired with MotionStress, our targeted benchmark and VLM-based protocol for motion-intensive failures, PhysFlow consistently improves motion coherence over the Wan2.1 baseline at near-baseline inference cost, while existing training-free methods either underperform the baseline or incur substantial overhead. More broadly, our results suggest that the velocity field is an underused intervention surface for training-free video generation, complementing existing attention- and guidance-level methods.

Limitations. PhysFlow assumes the latent video is sufficiently smooth along the frame axis for the inter-frame difference to be a meaningful derivative; it may degrade on heavily-tokenized latent representations or in extremely few-step distilled models, where the active schedule offers too few correction opportunities. In our additional 10- and 30-step evaluations, gains remain positive (Appendix O). PhysFlow also corrects sampler-induced trajectory inconsistency, not prompt ambiguity: failures rooted in under-specified text descriptions, such as unstated motion direction or unrealistic dynamics, lie outside its scope.

Acknowledgments and Disclosure of Funding

Funding. This work was supported in part by the New Generation Artificial Intelligence-National Science and Technology Major Project (No. 2025ZD0123501), the Open Research Fund of the State Key Laboratory of Multimodal Artificial Intelligence Systems (No. MAIS2025064), and the InnoHK Program.

Competing Interests. The authors declare no competing interests.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL `https://arxiv.org/abs/2502.13923`.

[2] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, et al. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

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

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

[5] Jiazi Bu, Pengyang Ling, Pan Zhang, Tong Wu, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. Bytheway: Boost your text-to-video generation model to higher quality in a training-free way. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12999–13008, 2025.

[6] Hila Chefer, Uriel Singer, Amit Zohar, Yuval Kirstain, Adam Polyak, Yaniv Taigman, Lior Wolf, and Shelly Sheynin. Videojam: Joint appearance-motion representations for enhanced motion generation in video models. *arXiv preprint arXiv:2502.02492*, 2025.

[7] Dar-Yen Chen, Hmrishav Bandyopadhyay, Kai Zou, and Yi-Zhe Song. Normalized attention guidance: Universal negative guidance for diffusion models. *arXiv preprint arXiv:2505.21179*, 2025.

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

[9] Yuren Cong, Mengmeng Xu, Christian Simon, Shoufa Chen, Jiawei Ren, Yanping Xie, Juan-Manuel Perez-Rua, Bodo Rosenhahn, Tao Xiang, and Sen He. Flatten: optical flow-guided attention for consistent text-to-video editing. *arXiv preprint arXiv:2310.05922*, 2023.

[10] Weichen Fan, Amber Yijia Zheng, Raymond A Yeh, and Ziwei Liu. Cfg-zero*: Improved classifier-free guidance for flow matching models. *arXiv preprint arXiv:2503.18886*, 2025.

[11] Daniel Geng, Charles Herrmann, Junhwa Hur, Forrester Cole, Serena Zhang, Tobias Pfaff, Tatiana Lopez-Guevara, Yusuf Aytar, Michael Rubinstein, Chen Sun, et al. Motion prompting: Controlling video generation with motion trajectories. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1–12, 2025.

[12] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation. *arXiv preprint arXiv:2211.13221*, 2022.

[13] Olivier J Hénaff, Robbe LT Goris, and Eero P Simoncelli. Perceptual straightening of natural videos. *Nature neuroscience*, 22(6):984–991, 2019.

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

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

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

[17] Junha Hyung, Kinam Kim, Susung Hong, Min-Jung Kim, and Jaegul Choo. Spatiotemporal skip guidance for enhanced video diffusion sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11006–11015, 2025.

[18] Yang Jin, Zhicheng Sun, Kun Xu, Liwei Chen, Hao Jiang, Quzhe Huang, Chengru Song, Yuliang Liu, Di Zhang, Yang Song, et al. Video-lavit: Unified video-language pre-training with decoupled visual-motional tokenization. *arXiv preprint arXiv:2402.03161*, 2024.

[19] Bingyi Kang, Yang Yue, Rui Lu, Zhijie Lin, Yang Zhao, Kaixin Wang, Gao Huang, and Jiashi Feng. How far is video generation from world model: A physical law perspective. *arXiv preprint arXiv:2411.02385*, 2024.

[20] Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15954–15964, 2023.

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

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

[23] Shaowei Liu, Zhongzheng Ren, Saurabh Gupta, and Shenlong Wang. Physgen: Rigid-body physics-grounded image-to-video generation. In *European Conference on Computer Vision*, pages 360–378. Springer, 2024.

[24] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[25] Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tieyong Zeng, Raymond Chan, and Ying Shan. Evalcrafter: Benchmarking and evaluating large video generation models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22139–22149, 2024.

[26] Yang Luo, Xuanlei Zhao, Mengzhao Chen, Kaipeng Zhang, Wenqi Shao, Kai Wang, Zhangyang Wang, and Yang You. Enhance-a-video: Better generated video for free. *arXiv preprint arXiv:2502.07508*, 2025.

[27] Wan-Duo Kurt Ma, John P Lewis, and W Bastiaan Kleijn. Trailblazer: Trajectory control for diffusion-based video generation. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024.

[28] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

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

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

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

[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. URL `https://arxiv.org/abs/2412.15115`.

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

[34] Ariel Shaulov, Itay Hazan, Lior Wolf, and Hila Chefer. Flowmo: Variance-based flow guidance for coherent motion in video generation. *arXiv preprint arXiv:2506.01144*, 2025.

[35] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

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

[37] 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.

[38] Bing Wu, Chang Zou, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Jack Peng, Jianbing Wu, Jiangfeng Xiong, Jie Jiang, et al. Hunyuanvideo 1.5 technical report. *arXiv preprint arXiv:2511.18870*, 2025.

[39] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7623–7633, 2023.

[40] Tianxing Wu, Chenyang Si, Yuming Jiang, Ziqi Huang, and Ziwei Liu. Freeinit: Bridging initialization gap in video diffusion models. In *European conference on computer vision*, pages 378–394. Springer, 2024.

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

[42] Jianhao Yuan, Xiaofeng Zhang, Felix Friedrich, Nicolas Beltran-Velez, Melissa Hall, Reyhane Askari-Hemmat, Xiaochuang Han, Nicolas Ballas, Michal Drozdzal, and Adriana Romero-Soriano. Inference-time physics alignment of video generative models with latent world models. *arXiv preprint arXiv:2601.10553*, 2026.

[43] Shuhai Zhang, ZiHao Lian, Jiahao Yang, Daiyuan Li, Guoxuan Pang, Feng Liu, Bo Han, Shutao Li, and Mingkui Tan. Physics-driven spatiotemporal modeling for ai-generated video detection. *arXiv preprint arXiv:2510.08073*, 2025.

[44] Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36:49842–49869, 2023.

A MotionStress-100 Benchmark

A.1 Motivation

While general-purpose text-to-video benchmarks such as VBench [16] cover a broad range of evaluation dimensions, they aggregate performance across a wide prompt distribution and often dilute the signal from motion-specific failures. In particular, our preliminary inspection of state-of-the-art T2V samplers reveals five recurring families of motion-related artifacts that current benchmarks do not isolate cleanly:

- **Trajectory degradation under sustained linear motion** — high-speed objects (vehicles, animals) drift, deform, or curve unnaturally during long straight motion.
- **Anatomical violation under articulated motion** — humans and animals develop extra limbs, limb teleportation, or frozen frames during complex actions.
- **Discontinuity in fluid and particle dynamics** — fluids, smoke, and fire flicker between frames or appear/disappear locally rather than evolving continuously.
- **Geometry warping under camera motion** — backgrounds and rigid structures distort during pans, tracking shots, or zooms.
- **Rigid-body deformation in fine-grained physical interaction** — mechanical objects (gears, hourglasses, scissors) lose their geometry or violate physical constraints during interaction.

To support targeted evaluation of these failure modes, we construct **MotionStress-100**, a 100-prompt motion-stress benchmark organized into five categories of 20 prompts each, mapping directly to the failure families above: linear motion, human/complex motion, fluid and particle dynamics, camera motion, and fine-grained physics.

A.2 Prompt construction

Prompts were generated with the assistance of Gemini-3 Pro under explicit category-level constraints: each prompt must (i) clearly belong to exactly one of the five failure-mode categories, (ii) describe a clean, recognizable motion that human raters can verify, and (iii) avoid mentioning any specific generation method or hyperparameter. We then manually filtered and edited the generated set to remove duplicates, ambiguous descriptions, and prompts that crossed category boundaries. Representative examples from each category are shown in Table 5.

Table 5: Representative prompts from each MotionStress-100 category.

Category	Example prompt
Linear motion	"A high-speed bullet train passing continuously from left to right through a snowy mountain landscape."
Human/complex motion	"A gymnast performing a backflip on a balance beam, maintaining human anatomy perfectly."
Fluid and particle	"Blue and red ink dropping into a glass of clear water, swirling and expanding naturally."
Camera motion	"A slow, perfectly smooth drone shot panning across a sprawling futuristic city."
Fine-grained physics	"A pair of gears meshing and turning smoothly without their teeth deforming."

A.3 Prompt extension (MotionStress-100-Ext)

To evaluate whether the motion-stress challenge persists when text guidance is richer, we additionally construct **MotionStress-100-Ext** by expanding each MotionStress-100 prompt into a longer, more descriptive scene specification. We use the official prompt-extension utility from the Wan2.1 release [37], which wraps Qwen2.5-7B-Instruct [32] and is the default prompt-expansion module shipped with the Wan2.1 inference pipeline. A representative pair is shown in Table 6. The extension introduces additional appearance, lighting, and stylistic details, but does not alter the core motion described in the original prompt.

Table 6: Example of prompt extension via Qwen2.5-7B-Instruct.

Original (MS-100)	"A galloping white horse running in a straight line across a vast green meadow."
Extended (MS-100-Ext)	"A majestic white horse galloping in a straight line across a vast, lush green meadow. The horse's coat shines under the sunlight, and its mane flows freely behind it. Its hooves kick up small clouds of dust as it runs at full speed. The horse's powerful muscles ripple beneath its sleek skin, and its tail swishes behind it. The meadow is dotted with wildflowers and tall grasses swaying gently in the breeze. In the distance, rolling hills and a clear blue sky can be seen. The camera follows the horse from a medium shot, capturing its elegant movement and the vibrant greenery of the meadow. Natural motion and fluid camera movements enhance the dynamic scene."

A.4 Evaluation protocol

We use the generation configuration described in Section B. The remainder of this subsection details the VLM-based scoring and auxiliary metrics.

VLM-based question answering. Following recent practice in T2V evaluation, we use a vision-language model (VLM) as an automatic judge. For each generated video, we sample uniformly spaced frames and present them to **Qwen2.5-VL-7B-Instruct** [1] (in bf16 precision) along with a category-specific question set. Each category is associated with two or three yes/no questions targeting its dominant failure mode; questions are designed so that the expected answer aligns with motion-coherence success rather than absence of motion. Table 7 lists the full question set with the expected answers.

Table 7: Question sets used by the VLM judge for MotionStress evaluation. Each category has two or three yes/no questions targeting its main failure mode. The "Expected" column gives the answer that corresponds to motion-coherent generation.

Category	Question	Expected
Linear motion	Does the object maintain a consistent speed throughout the video?	yes
	Does the object's trajectory show any sudden jumps or teleportation?	no
	Does the moving object maintain its shape consistently across frames?	yes
Human / complex	Do the person's limbs maintain natural proportions throughout the movement?	yes
	Is the motion fluid without any frame where the body appears frozen or distorted?	yes
	Does the action complete naturally without abrupt interruption?	yes
Fluid / particle	Does the fluid motion appear continuous without sudden appearance or disappearance?	yes
	Does the fluid maintain physically plausible behavior throughout?	yes
Camera motion	Does the background remain geometrically consistent during camera movement?	yes
	Is the camera motion smooth without sudden jerks?	yes
	Do objects maintain their 3D structure as the viewpoint changes?	yes
Fine-grained physics	Does the object maintain its physical properties (shape, texture) throughout?	yes
	Does the mechanical motion follow a physically plausible trajectory?	yes

Scoring. Let K_c denote the number of yes/no questions in category c (so $K_c \in \{2, 3\}$ in our setup). For each prompt p in category c, we generate 3 videos under different seeds. For each (video, question) pair, the VLM judge produces a yes/no answer; we assign a score of 1 if the answer matches the expected answer in Table 7, and 0 otherwise. The score for a single video is the average over the K_c questions, and the score for prompt p is the average over its 3 videos:

$$S_p = \frac{1}{3K_c} \sum_{v=1}^{3} \sum_{q=1}^{K_c} \mathbb{1}\big[\hat{a}_{v,q} = a_q^\star\big], \tag{8}$$

where $\hat{a}_{v,q}$ is the VLM's answer for video v on question q, and $a_q^\star$ is the expected answer. The category score is then the average over all 20 prompts in the category:

$$S_c = \frac{1}{20} \sum_{p \in \mathcal{P}_c} S_p, \tag{9}$$

and the final MotionStress score reported in the main paper is the unweighted average across all five categories:

$$S = \frac{1}{5} \sum_{c} S_c. \tag{10}$$

We average across categories rather than across all prompts so that each failure family contributes equally, regardless of how many questions it contains. All scores are reported as percentages.

Auxiliary metrics. In addition to the VLM-based MotionStress score, we report three auxiliary metrics that are common in the T2V literature.

CLIP-TC (CLIP temporal consistency) measures frame-level feature stability over time using the CLIP image encoder. For each video, we encode all generated frames with the CLIP image encoder and compute the average cosine similarity between every pair of consecutive frame embeddings; the reported score is the average over all videos, scaled by 100.

DINO-TC is computed analogously to CLIP-TC but using the DINOv2 [28] image encoder instead of CLIP, providing a complementary perspective on temporal feature consistency that is less biased toward semantic content and more sensitive to structural appearance.

CLIP-SIM measures text-video semantic alignment. For each video, we encode the text prompt with the CLIP [33] text encoder and each sampled frame with the CLIP image encoder, then compute the cosine similarity between the prompt embedding and each frame embedding; the per-video score is the average cosine similarity across frames, and the reported number is the average over all videos in the benchmark, scaled by 100. We restrict CLIP-SIM to the non-extended setting because (i) the CLIP text encoder has a 77-token cap, which can truncate the richer scene descriptions in MS-100-Ext, and (ii) CLIP is known to be relatively insensitive to fine-grained motion semantics, an effect that compounds under longer prompts.

We treat all three as auxiliary evidence rather than primary metrics. CLIP-SIM is reported as a supplementary alignment check; CLIP-TC and DINO-TC, like all frame-similarity-based scores, can favor overly static videos and are therefore reported only alongside the MotionStress QA score, which directly probes category-specific motion failures. Table 8 reports the CLIP-SIM scores on MS-100; CLIP-TC and DINO-TC are reported in the main paper ((Figure 5)).

Table 8: CLIP-SIM on MotionStress-100 (text-video semantic alignment, higher is better). Reported on MS-100 only; see text for the rationale.

Method	CLIP-SIM
Baseline (Wan2.1)	31.46
CFG-Zero*	31.32
FlowMo	<u>31.59</u>
Ours	**31.64**

Our method achieves the highest CLIP-SIM on MS-100, while the spread across all four methods is small ($\sim$0.3 points), consistent with CLIP-SIM's known low sensitivity to motion-related differences. This reinforces our use of MotionStress QA as the primary metric and CLIP-SIM as a supplementary check on text-video alignment.

Frame sampling for temporal consistency metrics. For computational tractability and consistency with prior practice [16, 25], the VLM-based MotionStress QA score uses 8 uniformly sampled frames per video, matching the visual context capacity of Qwen2.5-VL-7B-Instruct. For the CLIP-TC and DINO-TC metrics in the main paper, however, we report *frame-level* values computed on all generated frames (41 frames per video for MotionStress evaluation), which align more closely with the metric's intent of capturing per-frame temporal consistency. For completeness, Table 9 reports both the keyframe-level (8-frame) and frame-level (41-frame) versions of CLIP-TC and DINO-TC. The relative ranking between methods is preserved across both protocols, indicating that our improvements in temporal consistency are robust to the choice of frame sampling strategy. The frame-level scores are uniformly higher in absolute value because consecutive frames are closer in feature space than uniformly sampled keyframes, but the gap between methods remains consistent.

Table 9: CLIP-TC and DINO-TC on MotionStress benchmarks under two protocols.see Appendix D.2 for the full 4-method comparison under both protocols. **Keyframe-level** computes consecutive cosine similarities on 8 uniformly sampled frames (the standard 8-frame VBench/EvalCrafter protocol). **Frame-level** computes them on all 41 generated frames. Higher is better; **bold** = best per column.

		Keyframe-level (8 frames)		Frame-level (all 41 frames)	
Setting	**Method**	CLIP-TC ↑	DINO-TC ↑	CLIP-TC ↑	DINO-TC ↑
MS-100	Baseline	97.73	94.93	98.94	98.01
	Ours	**97.88**	**95.42**	**99.07**	**98.34**
MS-100-Ext	Baseline	97.63	94.64	98.93	98.06
	Ours	**97.63**	**94.97**	**99.01**	**98.27**

B Generation settings

We use Wan2.1-T2V-1.3B as the backbone for all main experiments, with 50 UniPC sampling steps, classifier-free guidance scale 5.0, and flow shift 5.0.

MotionStress-100 and MotionStress-100-Ext. We generate videos at 832×480 resolution with 41 frames at 16 fps (approximately 2.5 seconds), producing 3 videos per prompt under different random seeds. The shorter clip length is chosen so that motion artifacts are not diluted by long static segments, and to enable paired-seed evaluation within our compute budget.

VBench-1 and VBench-1-Ext. We follow the official VBench evaluation protocol: videos are generated at 832×480 resolution with 81 frames at 16 fps (approximately 5 seconds), and we sample 5 videos per prompt as required by the VBench scoring procedure. For VBench-1 we use the original prompt set; for VBench-1-Ext we use the official Wan2.1-T2V-1.3B augmented prompts released alongside the VBench codebase.[1] This ensures that our VBench numbers are directly comparable to results reported by the VBench authors and by other papers using the same augmented set.

Hardware. All experiments are conducted on a single NVIDIA A100 GPU (80GB).

Fairness across methods. All baselines (FlowMo, CFG-Zero*) and our method share identical generation settings within each benchmark, including the same prompts, random seeds (for paired-seed evaluation), backbone, sampling steps, resolution, and frame count. The only difference is the inference-time intervention applied at each sampling step.

C Hyperparameters of our method

For all experiments in the main paper we use a single fixed set of hyperparameters across all prompts and benchmarks:

- **Diagnostic steering strength**: $\alpha = 5.0$ (Eq. (5)).
- **Straightness prior strength**: $\beta = 1.0$ (Eq. (6)).
- **Safe clipping threshold**: $\tau = 20$ (ℓ_∞-norm bound on the combined correction in Eq. (7)).
- **Schedule**: $\rho_{\text{start}} = 0$, $\rho_{\text{end}} \approx 0.7$; the correction is enabled from the first sampling step until approximately 70% of the trajectory, after which the original sampler runs unmodified to refine fine-grained appearance.
- **Numerical stability constant**: $\epsilon = 10^{-4}$.
- All diagnostic and correction computations are performed in float32; the base T2V model retains its original inference precision.

These values were chosen on a small held-out set of motion-stress prompts not included in MotionStress-100 and were *not* tuned per prompt, per category, or per benchmark. The same set is used for MotionStress-100, MotionStress-100-Ext, and all VBench evaluations.

[1] `https://github.com/Vchitect/VBench/tree/master/prompts/augmented_prompts/Wan2.1-T2V-1.3B`

Ablation variants. The clipping ablations reported in Table 4 use $\tau = 1$ ("small clip") and $\tau = 100$ ("large clip"), respectively two orders of magnitude below and five times above the default $\tau = 20$. The *w/o straightness* variant sets $\beta = 0$ while keeping $\alpha = 5.0$; the *w/o diagnostic* variant sets $\alpha = 0$ while keeping $\beta = 1.0$. All other hyperparameters are unchanged.

D Baseline configurations and hyperparameter sweep

D.1 Default configurations used in the main paper

For each external baseline, we use the official implementation released by the original authors with their default configurations.

FlowMo [34]. We use the official implementation at `https://github.com/arielshaulov/FlowMo` with all default hyperparameters from the released code: learning rate $\eta = 0.001$, one refinement iteration per active step, and refinement applied at the first 12 of 50 sampling steps, following the schedule shipped with the repository. We do not modify any aspect of the FlowMo pipeline.

CFG-Zero* [10]. We use the official implementation at `https://github.com/WeichenFan/CFG-Zero-star` with the recommended configuration for 832×480 generation: zero-init applied at the first sampling step (`zero_steps=1`) and optimized scale enabled. The CFG-Zero* repository recommends flow shift 3.0 for 832×480 resolution, which we use for the main paper results. We additionally evaluate flow shift 5.0 below to verify that our reported numbers are not an artifact of this configuration choice.

D.2 Hyperparameter sweep

To verify that the gap between our method and the two training-free baselines is not an artifact of suboptimal hyperparameters, we additionally evaluate each baseline under an alternative configuration suggested by the original papers or by their natural extension to our setting:

- **FlowMo with** $\eta = 0.005$, the learning rate value reported in the FlowMo paper text (the released code defaults to $\eta = 0.001$).
- **CFG-Zero* with flow shift 5.0**, matching the flow shift used by the Wan2.1 baseline and FlowMo, to control for any effect of the shift parameter on the comparison.

We rerun both alternative configurations on MS-100 and MS-100-Ext. Results are summarized in Table 10 and Table 11.

Table 10: Hyperparameter sweep on **MotionStress-100**. Higher is better. "Default" rows are reproduced from Table 1 for reference; "Alt." rows are the sweep configurations. CLIP-TC and DINO-TC are reported under both **keyframe-level** (8 uniformly sampled frames) and **frame-level** (all 41 frames) protocols; see Appendix A.4.

								8 frames		41 frames	
Method	Config	Linear	Human	Fluid	Camera	Fine	Avg.	CLIP-TC	DINO-TC	CLIP-TC	DINO-TC
CFG-Zero*	Default (shift 3.0)	61.67	44.44	83.33	**60.56**	65.00	63.00	96.58	93.14	<u>98.89</u>	<u>97.85</u>
CFG-Zero*	Alt. (shift 5.0)	<u>64.44</u>	54.44	85.00	55.00	67.50	65.28	96.66	93.04	98.20	97.21
FlowMo	Default ($\eta = 0.001$)	62.78	<u>56.11</u>	<u>90.00</u>	<u>60.00</u>	<u>68.33</u>	<u>67.44</u>	<u>97.75</u>	<u>94.91</u>	98.15	97.14
FlowMo	Alt. ($\eta = 0.005$)	63.33	53.33	89.17	58.89	67.50	66.44	97.68	94.56	98.86	97.79
Ours	—	**66.11**	**63.33**	**94.16**	58.89	**70.83**	**70.67**	**97.88**	**95.42**	**99.07**	**98.34**

Discussion. The sweep yields three observations. First, **CFG-Zero* benefits substantially from flow shift 5.0** on both MS-100 ($63.00 \rightarrow 65.28$) and MS-100-Ext ($63.33 \rightarrow 69.11$), confirming that the default 3.0 shift is suboptimal under motion-stress prompts. Even at its best configuration, however, CFG-Zero* remains well below our method. Second, **FlowMo at** $\eta = 0.005$ **does not consistently outperform** $\eta = 0.001$: on MS-100 the higher learning rate slightly underperforms the released

Table 11: Hyperparameter sweep on **MotionStress-100-Ext**. Higher is better. CLIP-TC and DINO-TC are reported under both keyframe-level (8 sampled frames) and frame-level (all 41 frames) protocols.

								8 frames		41 frames	
Method	Config	Linear	Human	Fluid	Camera	Fine	Avg.	CLIP-TC	DINO-TC	CLIP-TC	DINO-TC
CFG-Zero*	Default (shift 3.0)	60.56	70.56	89.16	60.56	65.00	63.33	96.39	92.64	98.08	97.06
CFG-Zero*	Alt. (shift 5.0)	63.33	75.00	85.83	60.56	60.83	69.11	96.63	93.20	98.28	97.48
FlowMo	Default ($\eta = 0.001$)	62.22	75.00	91.66	62.78	61.67	70.67	97.54	94.32	98.82	97.76
FlowMo	Alt. ($\eta = 0.005$)	63.33	73.39	90.83	61.67	60.83	70.11	97.26	93.20	98.64	97.38
Ours	—	**63.88**	**80.00**	**93.33**	**63.88**	**65.83**	**73.39**	**97.63**	**94.97**	**99.01**	**98.27**

default, and on MS-100-Ext the two configurations are within 0.5 points of each other. This suggests that the FlowMo released default is a reasonable choice and our main-paper numbers are not an artifact of using a too-conservative learning rate. Third, **across all four baseline configurations and both benchmarks, our method retains a clear margin**: the gap to the best baseline configuration is at least $+2.7$ on MS-100 (vs. CFG-Zero* at shift 5.0) and $+2.7$ on MS-100-Ext (vs. FlowMo at $\eta = 0.001$). **The auxiliary CLIP-TC and DINO-TC scores show the same pattern under both keyframe-level (8-frame) and frame-level (41-frame) protocols: our method achieves the highest temporal consistency in every setting.**

We therefore conclude that the comparisons in the main paper reflect the inherent capability of each method on motion-stress prompts, rather than any specific choice of hyperparameter.

E Human evaluation details

To complement the VLM-based MotionStress evaluation, we conduct a small-scale human preference study. This appendix details the protocol used to produce the results in Figure 6.

Sampling protocol. We randomly sample 20 video pairs from the paired-seed MotionStress-100-Ext generations described in Section A.4, ensuring coverage across the five motion-stress categories. Each pair consists of one video generated by the Wan2.1 baseline and one by our method, under identical prompt and random seed; the only difference is whether the proposed velocity correction is enabled. The order of presentation (left/right) within each pair is randomized to avoid position bias.

Rating procedure. Raters were presented with each video pair side-by-side via a Tencent Docs questionnaire, together with the original text prompt. For each pair, raters were asked to compare the two videos along three independent axes:

- **Motion coherence** — which video exhibits more temporally consistent and physically plausible motion (no object fragmentation, no abrupt deformation, no flickering)?
- **Visual detail** — which video has higher per-frame visual fidelity (sharper structure, fewer geometric artifacts, more realistic appearance)?
- **Text-prompt alignment** — which video better reflects the content described by the input prompt?

For each axis the rater chose one of three options: *Ours better*, *Baseline better*, or *Both comparable / tie*. Raters were not informed which video was generated by which method. The full instructions and an example pair are shown to each rater at the start of the questionnaire.

Aggregation. We collected 200 total responses across the 20 pairs and three axes. For each axis, the win rate of a method is computed as the fraction of responses choosing that method as better; the tie rate is the fraction of responses choosing *Both comparable*. The bar chart in Figure 6 reports these three quantities (Ours, Both, Baseline) per axis. Error bars indicate 95% confidence intervals computed by bootstrap resampling over responses.

Notes on interpretation. The "Both comparable" option absorbs pairs where the difference is small or imperceptible, which is common for paired-seed generations under prompts that the baseline already handles well. Our method's gains therefore appear primarily as a shift from the *Baseline better* bucket into the *Ours better* and *Both comparable* buckets, rather than as a uniform reversal of preferences. This is consistent with the design of our method: the correction is most active where the latent flow residual is large, leaving prompts with already-coherent motion largely unchanged.

F Generalization to a stronger backbone (Wan2.1-T2V-14B)

The main paper reports all results on Wan2.1-T2V-1.3B for compute reasons. To verify that the proposed correction is not specific to the smaller backbone, we additionally evaluate it on Wan2.1-T2V-14B [37] under MotionStress-100-Ext. We use the official 14B checkpoint with default sampler settings, the same prompts and random seeds as the 1.3B paired-seed evaluation, and the same hyperparameters as in Section C without any per-model tuning.

Table 12: MotionStress-100-Ext results with the Wan2.1-T2V-14B backbone. Higher is better.

Method	Linear	Human	Fluid	Camera	Fine	Avg.	CLIP-TC	DINO-TC
Baseline (14B)	**63.33**	**84.44**	90.83	62.22	61.67	72.50	97.78	94.71
Ours (14B)	**63.33**	**84.44**	**95.83**	**63.33**	**62.50**	**73.89**	**97.87**	**94.95**

Discussion. As shown in Table 12, On Wan2.1-T2V-14B, our method improves the average MotionStress score from 72.50 to 73.89 ($+1.39$), with the largest gains on fluid/particle dynamics ($+5.00$) and camera motion ($+1.11$). The improvement is smaller than on the 1.3B backbone ($+3.78$ on the same MS-100-Ext benchmark), which is consistent with expectation: a stronger backbone already has better internal motion modeling, leaving less room for inference-time correction to recover. We nonetheless observe two encouraging properties. First, the correction *never harms* performance on any category — the 14B baseline already saturates linear motion and human motion at high scores, and our method matches the baseline exactly on those categories while improving the others. Second, the same hyperparameters that work well for the 1.3B model transfer directly to 14B without any retuning, which suggests the diagnostic targets a backbone-agnostic property of the latent trajectory rather than a specific failure mode of the smaller model.

This experiment confirms that the proposed velocity correction is a backbone-agnostic, training-free intervention whose value persists — albeit with diminishing returns — as the underlying model scales up.

G Choice of VBench dimensions

VBench [16] comprises 16 evaluation dimensions covering aspects such as subject consistency, background consistency, temporal flickering, motion smoothness, dynamic degree, aesthetic quality, imaging quality, object class, multiple objects, human action, color, spatial relationship, scene, appearance style, temporal style, and overall consistency. We report four of these dimensions in the main paper, selected on the basis of methodological relevance to the proposed velocity correction:

- **Temporal flickering** and **motion smoothness** — the two dimensions most directly tied to motion coherence, which our method explicitly targets.
- **Subject consistency** — to verify that motion stabilization does not damage object identity across frames.
- **Scene** — as a content-preservation check verifying that the correction does not substantially disrupt the generated scene semantics.

Other VBench dimensions, such as object class accuracy or appearance style, are largely determined by the base T2V model's text-to-content mapping, which our inference-time correction does not modify and is therefore unlikely to affect in either direction. We do not optimize for any VBench dimension during method development; the four reported dimensions are chosen on methodological grounds, not on where our method performs best.

H Diagnostic behavior analysis

To verify that the proposed latent flow residual r_f provides a meaningful diagnostic signal in practice, and that the active schedule (Eq. (7)) operates as intended, we conduct two diagnostic experiments using the same Wan2.1-T2V-1.3B backbone and hyperparameters as the main paper.

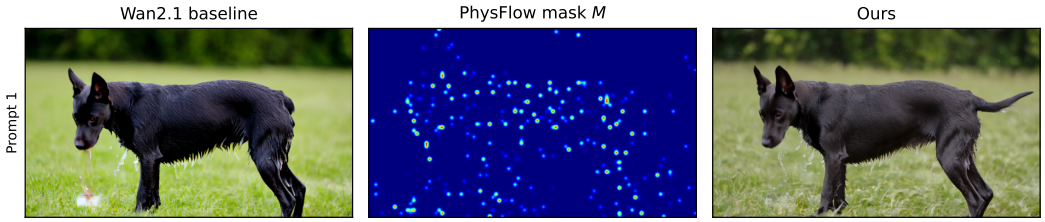

Figure 7: Spatial localization of the PhysFlow residual mask M_f on a representative motion-stress prompt. **Left**: Wan2.1 baseline. **Middle**: per-pixel mask M_f averaged across all active sampling steps and latent frames; brighter regions receive stronger correction. **Right**: PhysFlow output under the same prompt and seed.

H.1 Residual as a correction direction

The latent flow residual is not used as an independent detection score that is heuristically transferred to control: the residual and the correction direction are coupled by construction. At each frame location, recall

$$r_f = \langle v_t^f, \hat{g}_f \rangle - 1,$$

where $\hat{g}_f$ is the normalized latent diagnostic direction (Eq. (3)). We compute $\hat{g}_f$ from v_t^f at each step but treat it as a fixed (stop-gradient) reference when forming the correction; in our implementation this corresponds to a `detach()` on $\hat{g}_f$ before the residual product. Under this convention, the gradient of the local mismatch loss $\mathcal{L}_f = \frac{1}{2} r_f^2$ with respect to v_t^f simplifies to

$$\nabla_{v_t^f} \mathcal{L}_f \;=\; r_f \, \hat{g}_f,$$

so the diagnostic correction

$$\Delta v_{\text{diag}}^f \;=\; -\alpha \, r_f \, \hat{g}_f$$

is exactly the first-order descent step that pulls the velocity–diagnostic pairing back toward the unit reference. In this sense, r_f measures how far the pairing has drifted, while $\hat{g}_f$ specifies the local axis along which the velocity should be corrected.

This construction adapts the NSG-VD [43] probability-flow view from post-hoc detection to online control: NSG-VD interprets deviation from the unit reference as evidence of a generated artifact, while we interpret the same deviation as a target for the descent direction prescribed by $\hat{g}_f$. We do not claim a calibrated physical conservation law in pixel space; instead, we use the latent residual as an online control signal whose effectiveness is supported by the ablations and the residual-curve analysis in this appendix.

H.2 Spatial localization of the residual mask

Figure 7 visualizes the residual-aware mask M_f used by PhysFlow during sampling. For each prompt, we compare the Wan2.1 baseline output (left) against the per-pixel mask aggregated over all active sampling steps (middle) and the corresponding PhysFlow output (right). The mask concentrates in high-frequency, motion-relevant regions — the dog's wet fur, leg articulation, and grass texture — while leaving low-residual background regions largely unmasked. This confirms that PhysFlow does not apply correction uniformly: the mask self-localizes to regions where the sampler is violating frame-axis consistency, consistent with the design rationale in Section 4.3.

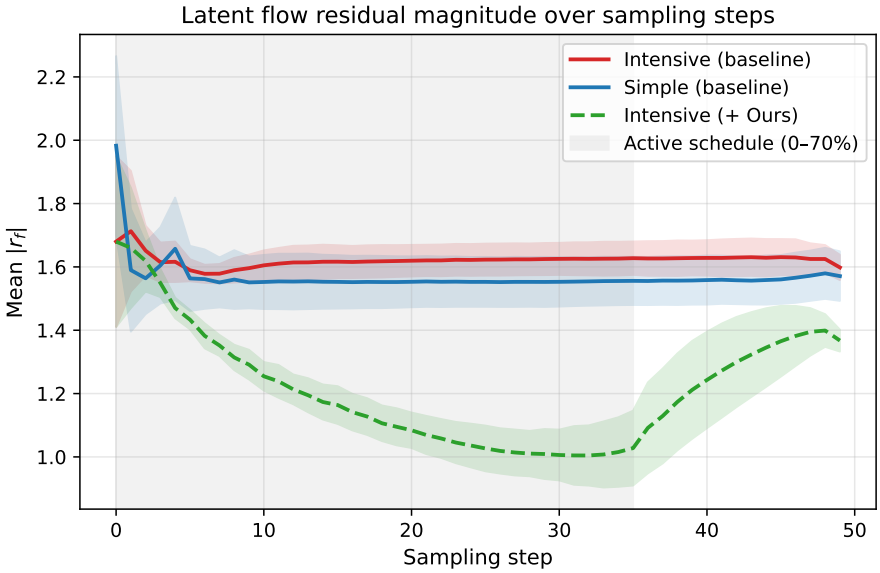

Figure 8: Mean latent flow residual $|r_f|$ over sampling steps. Motion-intensive prompts (red) show systematically higher residual than simple prompts (blue) under the baseline. PhysFlow (green dashed) drives the residual on intensive prompts down toward unity within the active schedule (shaded region, steps 0 to $\rho_{\text{end}} \cdot 50 \approx 35$); the residual rebounds after the schedule ends, confirming that the active window is operating as designed.

H.3 Residual evolution across sampling steps

Figure 8 reports the mean latent flow residual $|r_f|$ at each sampling step under three configurations: motion-intensive prompts under the baseline (red), simple prompts under the baseline (blue), and motion-intensive prompts with PhysFlow applied (green). Each curve is averaged across multiple prompts, with shaded bands indicating one standard deviation.

Three observations emerge: **(i)** Motion-intensive prompts consistently exhibit higher residual magnitude than simple prompts under the baseline (red vs. blue, gap ≈ 0.05–0.10), confirming that $|r_f|$ correlates with motion difficulty rather than being a generic noise indicator. **(ii)** Applying PhysFlow drives the residual on intensive prompts steadily downward toward unity (green dashed), reaching $|r_f| \approx 1.0$ near step 30 — direct evidence that the correction actively reduces the very quantity it diagnoses. **(iii)** After the active schedule ends at step $\rho_{\text{end}} \cdot 50 \approx 35$, the residual rebounds because the correction is intentionally disabled to let the sampler refine fine-grained appearance; this matches the schedule design and visually confirms that the active window in Eq. (7) operates as intended.

Together, these two analyses provide direct empirical support for two core claims of our method: that r_f identifies motion-inconsistent regions in latent space, and that the active schedule meaningfully delineates when the correction should and should not act.

H.4 Effect of schedule endpoint ρ_{end}

The active schedule in Eq. (7) is controlled by a single hyperparameter $\rho_{\text{end}} \in (0, 1)$ that determines how early PhysFlow disengages, letting the original sampler take over for the late refinement steps. The main paper uses $\rho_{\text{end}} = 0.7$ throughout (correction active for the first $\sim 70\%$ of sampling steps). To verify this choice, we additionally evaluate two alternatives: $\rho_{\text{end}} = 0.3$ (correction disengages early, leaving most of the trajectory uncorrected) and $\rho_{\text{end}} = 0.9$ (correction stays active almost until the end, intruding into the late detail-refinement stage).

Qualitative results in Figure 9 reveal a clear failure mode at each extreme. With $\rho_{\text{end}} = 0.3$, the correction stops too early to fully resolve motion artifacts: residual fragmentation and structural drift remain visible in the final video, indicating that early-stage stabilization alone is insufficient. With $\rho_{\text{end}} = 0.9$, the correction remains active during steps that the pretrained sampler uses for

fine-grained appearance generation; the resulting videos exhibit visible residual noise and texture degradation, reflecting interference with the sampler's late-stage denoising dynamics. The default $\rho_{\text{end}} = 0.7$ balances these two regimes, providing sufficient motion-stabilization coverage while leaving the final $\sim 30\%$ of sampling steps untouched for appearance refinement. This is consistent with the residual evolution observed in Figure 8, where the latent flow residual is meaningfully reduced before step $\rho_{\text{end}} \cdot 50 \approx 35$ and naturally rebounds afterward.

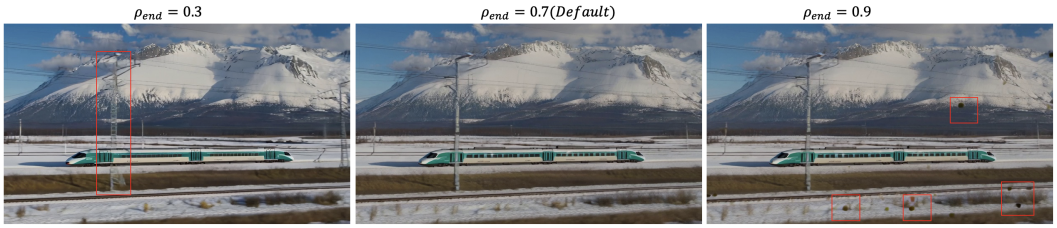

Figure 9: Effect of the schedule endpoint ρ_{end} under the same prompt and seed. $\rho_{\text{end}} = 0.3$ (correction disengages early): residual motion artifacts remain visible. $\rho_{\text{end}} = 0.7$ (default): coherent motion with clean appearance. $\rho_{\text{end}} = 0.9$ (correction extends into late refinement steps): visible residual noise and texture degradation. The default value provides sufficient motion-stabilization coverage while preserving the late detail-refinement stage of the original sampler.

I Additional Qualitative Results

We provide additional qualitative comparisons in Figure 10 to complement the main-paper examples (Figure 4). The selected prompts span all five MotionStress categories and exhibit the same patterns observed in the main paper: PhysFlow consistently mitigates object fragmentation, geometric deformation, and temporal flickering while preserving overall scene content under identical prompts and seeds.

J Transfer to Other Flow-Matching Backbones

We evaluate two additional flow-matching video-generation backbones without backbone-specific tuning, using the same correction implementation and hyperparameters as for Wan2.1-T2V-1.3B. Wan2.2 is evaluated with its default 121-frame generation setting. Table 13 shows positive average MotionStress gains on both architectures. These comparisons support cross-family transfer but do not establish universal efficacy on all samplers.

Table 13: Cross-backbone MotionStress QA scores (higher is better). All comparisons use fixed PhysFlow hyperparameters without backbone-specific retuning.

Backbone	Method	Linear	Human	Fluid	Camera	Fine	Avg.
HunyuanVideo-1.5	Baseline	**62.22**	65.56	91.67	59.44	73.33	70.44
	PhysFlow	61.67	**68.89**	**94.17**	**60.00**	**76.67**	**72.28**
Wan2.2 (121 frames)	Baseline	65.56	51.11	88.33	58.89	59.17	64.61
	PhysFlow	**71.67**	**52.78**	**92.50**	**63.33**	**65.00**	**69.06**

K Paired Uncertainty Analysis

We use a category-stratified paired bootstrap with 100,000 resamples to assess MotionStress improvements. Category-specific intervals are wider than overall intervals, especially for categories with smaller gains. We report intervals for the paired method difference, not independent confidence intervals for each method's absolute score.

Across the original three paired random seeds, the gains are $+4.17, +2.00, +5.00$ on MS-100 and $+6.00, +2.33, +3.00$ on MS-100-Ext. At the prompt level, PhysFlow improves, ties, and degrades

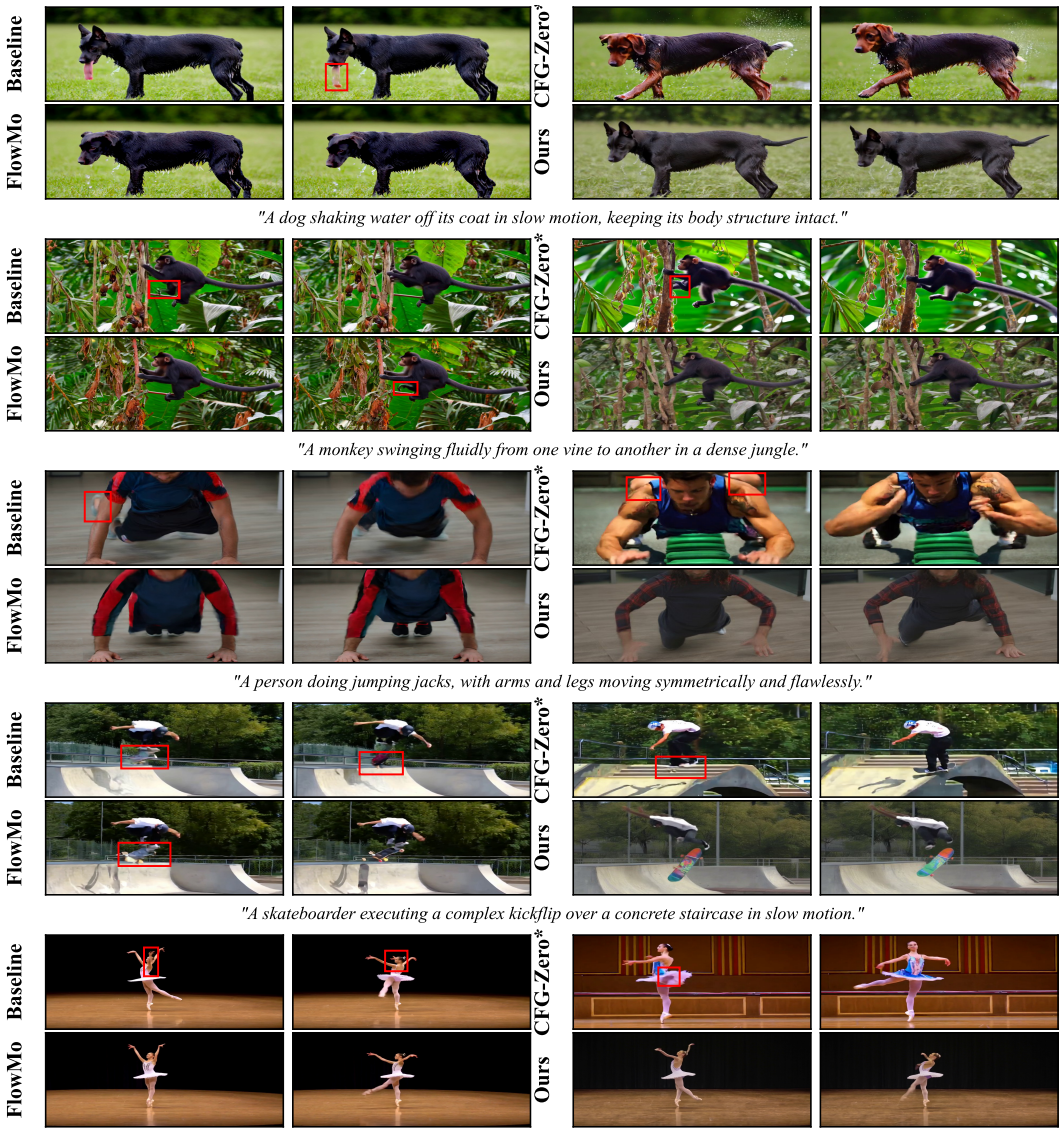

Figure 10: Additional qualitative comparisons on MotionStress prompts.

Table 14: MotionStress-100: score difference and 95% paired bootstrap confidence interval (percentage points).

Category	Baseline	PhysFlow	Δ	95% CI of Δ
Linear	65.56	66.11	+0.56	$[-6.67, 8.33]$
Human/complex	60.00	63.33	+3.33	$[-7.78, 13.33]$
Fluid/particle	93.33	94.17	+0.83	$[-5.00, 6.67]$
Camera	58.33	58.89	+0.56	$[-6.67, 7.22]$
Fine-grained physics	57.50	70.83	+13.33	$[6.67, 21.67]$
Average	**66.94**	**70.67**	**+3.73**	$[\mathbf{0.22}, \mathbf{7.22}]$

Table 15: MotionStress-100-Ext: score difference and 95% paired bootstrap confidence interval (percentage points).

Category	Baseline	PhysFlow	Δ	95% CI of Δ
Linear	63.33	63.89	+0.56	$[-3.89, 5.56]$
Human/complex	75.00	80.00	+5.00	$[-4.44, 15.56]$
Fluid/particle	87.50	93.33	+5.83	$[-1.67, 15.00]$
Camera	62.22	63.89	+1.67	$[0.00, 4.44]$
Fine-grained physics	60.00	65.83	+5.83	$[1.67, 10.00]$
Average	**69.61**	**73.39**	**+3.78**	$[\mathbf{0.89}, \mathbf{6.83}]$

32, 50, and 18 cases on MS-100, respectively, and 23, 67, and 10 cases on MS-100-Ext. Thus the gain is not uniform across individual prompts.

L Residual Magnitude and Motion Failures

On 45 baseline-only generations, we record the maximum absolute latent-flow residual at the last (50th) sampling step, before applying any correction. We sort the videos into three equal-sized residual groups. Their VLM MotionStress scores decrease as the recorded residual magnitude increases, providing a video-level association rather than a causal or calibrated physical interpretation.

Table 16: Residual magnitude versus MotionStress QA on 45 baseline videos. Failure rate is $1-$QA score.

Group	Residual range	Mean max.	QA (%)	Failure (%)
High	7.7989–9.1079	8.2475	62.22	37.78
Medium	7.2509–7.7605	7.4667	64.44	35.56
Low	6.7535–7.2363	7.0665	73.33	26.67

M Wan2.1-14B Residual Trajectory

Within each Wan2.1 backbone, we compare the residual decrease while PhysFlow is active (steps 5–35), its rebound after correction turns off (steps 35–50), and the final gap relative to the backbone's own intensive-motion baseline. Absolute residual magnitudes are not directly comparable between model sizes and the trajectory differences should not be interpreted as perceptual improvements accumulated across independent sampling steps.

Table 17: Within-backbone residual-trajectory statistics, using the same sampling-step indices.

Backbone	Active reduction	Late rebound	Final baseline–PhysFlow gap
Wan2.1-1.3B	0.46	0.36	0.23
Wan2.1-14B	0.50	0.43	0.18

The similar active-stage reductions suggest that the smaller MotionStress gain on 14B is not simply explained by a weaker active-stage correction. The 14B trajectory instead shows a somewhat larger rebound and smaller final gap within its own reference trajectory; neither statistic is a calibrated video-quality score.

N Full VBench-1 Evaluation

Table 18 reports all sixteen VBench-1 dimensions under a paired-seed evaluation protocol, using the same five random seeds $\{42, 43, 44, 45, 46\}$ for the baseline and PhysFlow.

Table 18: Full VBench-1 evaluation across all sixteen dimensions. Higher is better.

Metric	Baseline	PhysFlow
Temporal flickering	0.9895	**0.9903**
Motion smoothness	0.9835	**0.9886**
Subject consistency	**0.9571**	0.9542
Background consistency	**0.9753**	0.9697
Dynamic degree	**0.6388**	0.6166
Scene	0.2423	**0.2735**
Human action	**0.7500**	0.7460
Overall consistency	0.2365	**0.2379**
Spatial relation	0.7342	**0.7425**
Color	**0.8810**	0.8799
Multiple objects	0.6145	**0.6424**
Object class	0.7979	**0.8076**
Imaging quality	0.6872	**0.6953**
Aesthetic quality	**0.6482**	0.6417
Temporal style	0.2333	**0.2360**
Appearance style	0.1977	**0.1988**

O Reduced-Step Sampling

With fixed sampler and hyperparameters, PhysFlow raises the average MotionStress score from 57.06 to 63.00 at 10 steps (+5.94 points), and from 64.33 to 68.67 at 30 steps (+4.34 points), improving nine of the ten category–step comparisons. The active schedule spans about 70% of sampling steps, leaving approximately seven active corrections even at 10 steps. These experiments address 10- and 30-step operation but do not establish performance under one-step or other highly distilled generation.

Table 19: MotionStress QA results under reduced sampling-step budgets. Higher is better.

Category	10-step Base	10-step PhysFlow	30-step Base	30-step PhysFlow
Linear	57.78	61.11	65.00	68.89
Human/complex	38.89	47.22	50.56	55.00
Fluid/particle	79.17	85.83	90.00	89.17
Camera	49.44	56.67	54.44	62.78
Fine-grained physics	60.00	64.17	61.67	67.50
Average	**57.06**	**63.00**	**64.33**	**68.67**

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 the method, benchmark, and empirical claims, and the experiments directly evaluate these claims on MotionStress-100, MotionStress-100-Ext, VBench, human preference evaluation, inference cost, and a stronger Wan2.1-14B backbone. We report follow-up transfer results on Wan2.2 and HunyuanVideo-1.5, and avoid claiming universal generalization.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The Discussion and Conclusion section discusses limitations, including the fact that the residual is a latent-space diagnostic rather than a physical simulator, that the original core evaluation is based on Wan2.1 and the additional transfer tests are limited to two other backbones, that MotionStress-100 is a targeted diagnostic benchmark, and that VLM-based evaluation may contain judge bias.

   Guidelines:

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

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper does not present formal theoretical theorems or proofs. The probability-flow discussion is used as motivation for a latent diagnostic and is empirically evaluated through ablations and benchmark results.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Sections 4–5 and Appendices A–D describe the method, sampling settings, prompts, evaluation protocol, seeds, baselines, hyperparameters, and scoring procedure. The supplemental material includes code and prompt lists for reproducing the main MotionStress, VBench, ablation, and timing experiments.

   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: We provide all experimental details needed to reproduce the main results in the paper and appendix, including the complete method specification, hyperparameters, generation settings, baseline configurations, prompt construction procedure, VLM-based scoring protocol, auxiliary metrics, and qualitative generation protocol. We will release the code, MotionStress-100 prompts, evaluation scripts, and configuration files 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 is training-free and therefore has no optimizer or training split. Section 5 and the appendices specify the Wan2.1 checkpoints, sampling steps, resolution, frame count, CFG scale, prompt sets, random seeds, correction hyperparameters, baseline settings, and evaluation metrics.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The human preference study reports 95% confidence intervals computed by bootstrap resampling over responses. For automatic metrics, we use paired seeds where applicable and report category-level averages over multiple prompts and seeds; the exact aggregation protocol is provided in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The paper reports per-video inference time on a single A100 GPU and gives the generation configuration used for timing. Appendix D provides the main generation settings and compute configuration for reproducing the reported experiments.

   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 uses publicly available generative models and evaluation benchmarks, introduces a synthetic prompt benchmark, and conducts a low-risk human preference study over generated videos. We do not collect sensitive personal data or deploy the system in a real-world decision-making setting.

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: Section 6 (Discussion and Conclusion) notes that improving motion coherence in text-to-video generation can benefit creative tools, simulation, and model evaluation. It also acknowledges that higher-quality video synthesis may increase the realism of synthetic media, which raises misuse risks such as deceptive video generation. We regard these risks as a foreseeable negative societal impact of the broader class of T2V improvements to which this work belongs.

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: We do not release a new pretrained generative model or a scraped dataset. The released assets consist of code, synthetic prompts, evaluation scripts, and generated examples; we encourage use with existing model licenses, safety policies, and provenance mechanisms.

Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We cite all existing models, benchmarks, and tools used in the experiments, including Wan2.1, VBench, Qwen2.5-VL, CLIP, DINOv2, CFG-Zero*, and FlowMo. We use the official implementations and checkpoints according to their released terms and include asset references in the paper and supplemental material.

    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 MotionStress-100 and documents its motivation, five-category structure, prompt construction procedure, prompt-extension protocol, VLM question sets, and scoring formula in Appendix A. The prompt list and evaluation scripts will be released with documentation upon acceptance.

    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: Appendix E provides the human evaluation protocol, including the comparison axes, instructions shown to raters, response options, randomization/blinding procedure, aggregation method, and confidence interval calculation. The study was voluntary and uncompensated.

    Guidelines:

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

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

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

    Answer: [No]

    Justification: The human study consists of low-risk anonymous preference judgments over generated videos, does not collect sensitive personal information, and is not used to make decisions about participants. We disclose the task to participants and report the evaluation protocol in Appendix E.

    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 describes the use of Gemini-3 Pro for assisting MotionStress-100 prompt construction and Qwen2.5-7B-Instruct for prompt extension in Appendix A. These uses affect benchmark construction rather than the core PhysFlow algorithm, and the resulting prompts were manually filtered and documented.

    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.