# Householder Motion Embedding: Noise Modulation for Motion-Controllable Video Generation

**Tao Jun Lin**[1,2*] **Jinhong Ni**[1] **Ruyi Zha**[1] **Yifu Wang**[2]
**Xibin Song**[2] **Pan Ji**[2] **Hongdong Li**[1,3]

[1]The Australian National University [2]Vertex Lab
[3]Amazon Inc.
{taojun.lin, hongdong.li}@anu.edu.au

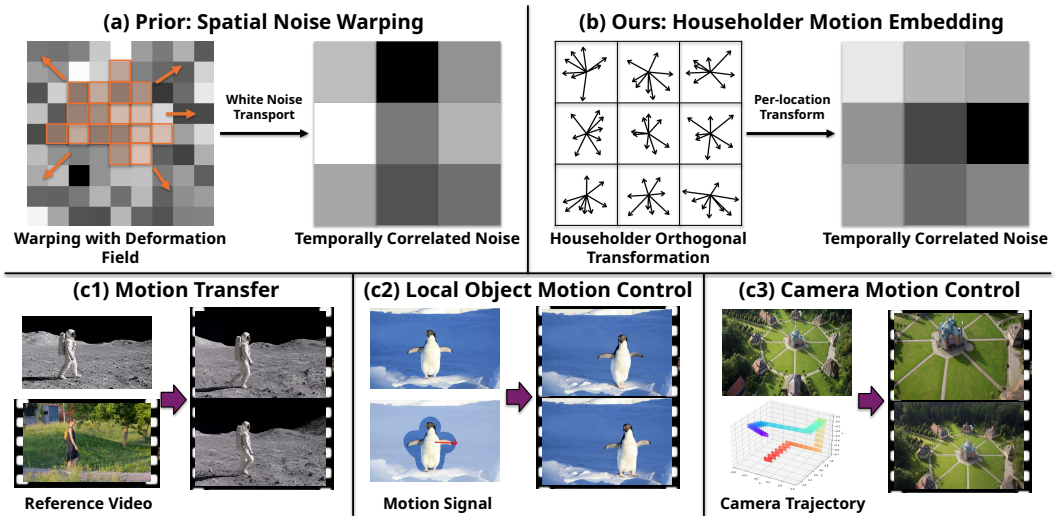



Figure 1: We propose **Householder Motion Embedding (HME)**, a simple noise-modulation method for motion-controllable diffusion video generation. **(a)** Prior methods rely on **spatial noise warping** driven by motion fields. *They preserve spatial Gaussianity only through costly correction steps.* **(b) HME** instead applies **per-location Householder transformations in channel space** to the noise. *It preserves spatial Gaussianity by construction, with a* $2.7\times$ *speed-up and better motion adherence.* **HME** naturally supports **(c1)** motion transfer, **(c2)** local object motion control, and **(c3)** camera motion control.

## Abstract

Diffusion models for video generation have achieved remarkable success, yet precise motion control under explicit user guidance remains challenging. Recent noise-warping methods introduce temporal correlations by warping the noise according to the user-provided motion signal. However, standard spatial warping generally disrupts the noise's i.i.d. Gaussian structure, requiring distribution correction, such as Jacobian-based correction, but at a substantial computational overhead. We propose **Householder Motion Embedding (HME)**, which encodes the motion signal into the noise through a sequence of Householder reflections that together form an orthogonal transformation applied to each location's channel vector. **HME** exactly preserves the spatial Gaussian structure within each frame while introducing cross-frame correlations that promote temporally consistent video. **HME** is

*Corresponding Author

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

agnostic to the network architecture and integrates directly into standard video diffusion fine-tuning pipelines. Experiments on different scenarios show that **HME** matches or outperforms the state of the art, while achieving a 2.7× speedup over noise warping. These results demonstrate that per-location Householder transformations in channel space offer a principled and efficient alternative to noise warping for motion-controllable video generation.

## 1 Introduction

Diffusion based models have recently achieved remarkable success in video generation, demonstrating strong visual fidelity and semantic coherence across temporal dimension [59, 27, 6, 4, 19, 67, 49, 12]. However, generating temporally consistent videos under explicit motion controls (modeling complex camera motion or scene dynamics) remains a fundamental challenge. Existing approaches [64, 45, 24, 35] often suffer from temporal drift, geometric inconsistency, or limited controllability when motion deviates from patterns observed during training.

A common wisdom to inject motion cues into the neural network architecture is through feature modulation [65, 43] or motion conditioning tokens [14, 52, 20, 53], which operate downstream of the stochastic generation process and leave the noise motion-agnostic. An alternative is to modify the noise directly. Building on the observation by PYoCo [13] that shared noise components across frames improve temporal coherence, HIWYN [8], InfRes [10], and GWTF [7] demonstrate that constructing temporally correlated noise by transporting the a priori latent noise via diffeomorphic deformation fields parameterized by optical flow significantly enhances frame-to-frame consistency without modifying model architecture or training objectives. However, spatial resampling during this noise transport inherently destroys the i.i.d. Gaussian structure of the noise, since interpolation induces correlations across neighboring locations and alters the marginal distribution. This matters because the diffusion model is trained to denoise i.i.d. standard Gaussian noise: noise with spatial autocorrelation or altered variance falls outside this training distribution, which prior work has shown degrades generation quality [8, 10, 7]. Consequently, all existing noise-warping methods must introduce additional processing steps to restore spatial Gaussianity at the cost of non-trivial computational overhead.

In this paper, we propose **Householder Motion Embedding (HME)**, a noise-modulation framework that resolves this spatial-Gaussianity bottleneck without sacrificing motion fidelity. Rather than warping noise via spatial deformation fields, we modulate the noise using the intended motion field by per-location orthogonal transformations in channel space, parameterized as the product of Householder reflections [17]. The Householder transformation, composed of a small number of reflections, yields expressive orthogonal operators at low computational cost, while the orthogonality ensures distribution preservation by construction.

Crucially, this distribution preservation guarantee is precisely what spatial noise warping methods lack. Modulating the noise through per-location Householder products therefore eliminates post-hoc correction entirely, requiring no Jacobian adjustments or distribution restoration steps, while inducing structured cross-frame temporal correlations that encode the geometric consistency of the underlying motion. Compared to deformation field based solutions, our method achieves exact spatial Gaussianity by construction, a $2.7\times$ speedup in noise processing time, and improved motion adherence on all three benchmarks.

Similar to existing noise warping methods, **HME** operates entirely at the noise initialization stage and requires no modifications to the denoising architecture, loss function, or training objective. The motion-embedded noise replaces the standard Gaussian noise at the start of the diffusion process, making the method plug-and-play across UNet-based and transformer-based video diffusion frameworks; we demonstrate it on the transformer-based CogVideoX. Our framework accepts any dense optical flow field as conditioning input regardless of its source: whether extracted from a reference video for motion transfer, rendered from a target camera trajectory for camera-controlled generation, or derived from user-specified drag interactions on the initial frame, thereby unifying diverse motion modalities within a single framework without motion-type-specific architectural modifications.

Contributions of this paper are as follows:

1. We propose a novel noise-modulation method for motion-controllable video diffusion that replaces spatial noise transport with per-location orthogonal transformations in latent channel space, parameterized as products of Householder reflections. Unlike existing noise-warping methods, this construction requires no post-hoc Gaussianity correction and operates in a single forward pass with no learnable parameters, making it a drop-in replacement for the noise initialization step in any video diffusion fine-tuning pipeline.

2. We derive the cross-frame covariance of the proposed noise, showing that the dependence between frames is an orthogonal transformation determined by the local motion, and analyse how the number of reflections controls the expressivity of this encoding.

3. We validate through extensive experiments on spatial Gaussianity preservation, motion-transfer and camera-controlled video generation benchmarks that **HME** matches or exceeds the state-of-the-art noise-warping baseline on nearly all motion-adherence and generation-quality metrics, while achieving a $2.7\times$ speedup in noise processing time.

## 2 Related Work

**Motion-controllable video generation.** Motion-controllable video generation methods broadly fall into three categories. Local object motion control methods [14, 24, 60] condition generation on user-specified trajectories via feature modulation or latent optimization, including Gaussian-filtered trajectory approaches [61, 53], two-stage sparse-to-dense motion field estimation [45, 37], region-wise trajectory and mask conditioning [65], and training-free latent optimization [36]. Global camera control methods [29, 54, 51] embed camera pose sequences into the denoising network through dedicated conditioning modules [20, 3, 52, 57, 55, 63] or learning Video-to-Video generation directly from imperfect 3D/4D renders [42, 18], which require large posed-video datasets for training. Motion transfer methods [1, 15, 28, 16, 40] propagate motion signals from a reference clip without additional supervision, either through temporal attention cloning [32], zero-shot feature guidance [60], inference-time schedule adaptation [46], motion disentanglement [43], or trajectory and optical flow conditioning [14, 30, 50]. Across all three paradigms, motion signals are injected downstream of the stochastic initialization, leaving the noise distribution motion-agnostic. Our method addresses this by modulating the noise through per-location orthogonal transformations, which preserve the i.i.d. Gaussian structure by construction while inducing cross-frame correlations that align per-frame noise distributions along the motion trajectory.

**Noise Warping for Temporally Consistent Video Generation.** Regardless of architectural difference between UNet-based and DiT-based video diffusion models, both are initialized from Gaussian noise whose correlation across frames can be influential to the temporal consistency of the generated output. PYoCo [13] first demonstrates that correlating the noise initialization across frames through a mixed noise prior substantially improves temporal coherence, motivating subsequent work on flow-guided noise warping. HIWYN [8] introduces integral noise, reinterpreting warped noise samples as integrals of a continuous underlying noise field to preserve spatial Gaussianity during deformation field guided transport, at the cost of higher computational complexity. InfRes [10] resolves this by deriving an equivalent infinite-resolution formulation through Brownian bridge increments, reducing overhead by orders of magnitude while matching HIWYN's theoretical guarantees. GWTF [7] further improves efficiency with a linear-time warping algorithm and demonstrates motion-controllable fine-tuning of modern video diffusion models across object motion, camera control, and motion transfer. Despite these advances, all existing methods share a fundamental limitation: spatial resampling during deformation-field-based transport disrupts the i.i.d. Gaussian noise structure, necessitating costly post-hoc correction at higher resolutions. Our method resolves this bottleneck by replacing spatial transport with per-location orthogonal transformations in channel space that preserve spatial Gaussianity exactly by construction.

Our method makes use of Householder transformations, leveraging their distribution-preserving property. Householder transformations have previously been used for other deep learning problems in very different contexts, ranging from position encoding [58, 33] to parameter-efficient fine-tuning [2, 11, 62].

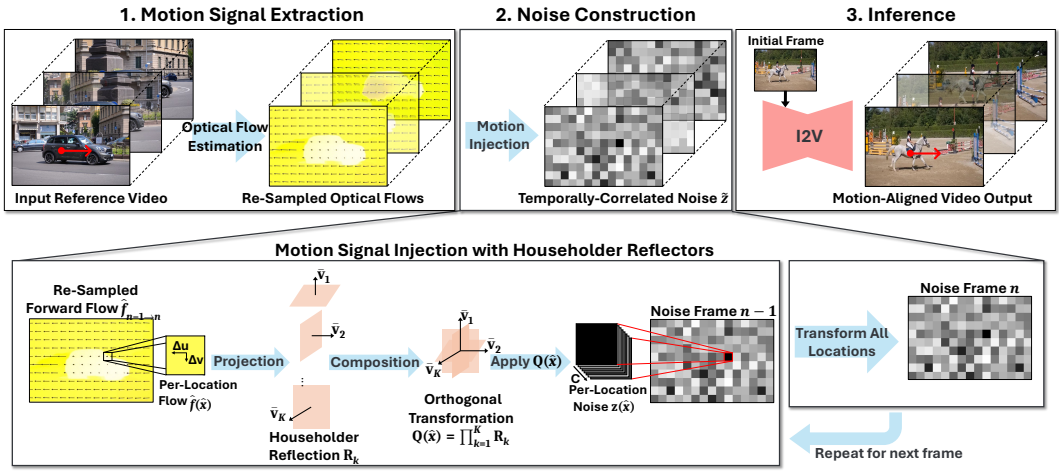



Figure 2: **HME** generates motion controlled videos in 3 stages: 1) optical flow estimation and spatial resampling, 2) temporally correlated noise construction via Householder Reflectors (see Section 3.1 for details), 3) diffusion model fine-tuning and inference. At inference time, our method enables motion- and camera-controlled video generation given a text prompt, an initial frame, and a reference video as the motion signal.

# 3 Method

Our goal is to construct motion-embedded noise that preserves *spatial Gaussianity* while inducing structured temporal correlations across video frames. We achieve this through per-location orthogonal transformations in latent channel space, parameterized as products of Householder reflections.

**Householder Reflector.** Householder reflectors are a standard, efficient numerical tool for constructing orthogonal transformations, requiring only an inner product and a vector update. Let $C$ denote the base model's latent channel size. A Householder reflector is a simple orthogonal operator defined by a single non-zero vector $\mathbf{v} \in \mathbb{R}^C$:

$$\mathbf{R}(\mathbf{v}) = \mathbf{I}_C - 2\bar{\mathbf{v}}\bar{\mathbf{v}}^\top, \qquad \bar{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|_2}. \tag{1}$$

Geometrically, $\mathbf{R}(\mathbf{v})$ reflects any vector across the hyperplane orthogonal to $\mathbf{v}$. As reflections are orthogonal transformations, they preserve Euclidean norms and, crucially, the Gaussian distribution: if $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_C)$, then $\mathbf{R}(\mathbf{v})\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_C)$ for any choice of $\mathbf{v}$. Composing $K$ such reflections yields a more expressive orthogonal operator, and any orthogonal matrix admits an exact decomposition into at most $C$ reflections [17]. We exploit this property to construct orthogonal transformations that preserve spatial Gaussianity exactly.

As shown in Fig. 2, our framework consists of three stages: (1) optical flow estimation, (2) a training-free Householder noise transformation module and (3) video diffusion model fine-tuning. The noise module generates temporally correlated noise during both training and inference, while the denoising architecture and training objective remain unchanged.

## 3.1 Motion-Embedded Noise Construction

### 3.1.1 Problem Formulation and Notation

We distinguish between *image space* and *latent noise space*. In image space, a video $V = \{F_0, \ldots, F_{N-1}\}$ consists of $N$ frames $F_n \in \mathbb{R}^{H\times W\times 3}$ on the $H \times W$ pixel grid, and optical flow assigns each pixel $x$ a displacement $\mathbf{f}(x) = (\Delta u, \Delta v) \in \mathbb{R}^2$. The VAE encodes the video into $\hat{N}$ latent frames on the $\hat{H} \times \hat{W}$ latent grid and each latent location $\hat{x}$ covers a pixel footprint $\Omega(\hat{x})$ (e.g., $8 \times 8$ pixels for CogVideoX). In latent noise space, each $\hat{x}$ carries a noise vector $\mathbf{z}(\hat{x}) \in \mathbb{R}^C$, drawn i.i.d. with $\mathbf{z}(\hat{x}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_C)$. Operations in latent noise space act on the channel dimension only, without modifying spatial coordinates. In our framework, optical flow acts as a conditioning signal that parameterizes per-location orthogonal transformations of latent noise vectors, rather than spatially resampling them.

### 3.1.2 Flow-to-Latent Alignment

The optical flow between consecutive frames, $\mathbf{f}_{n-1\to n} = \Phi(F_{n-1}, F_n)$, can be obtained with any off-the-shelf flow estimator $\Phi$; we use RAFT [47] in our experiments. Since the diffusion model operates on a spatially down-sampled latent grid, we resample the estimated flow field to match the latent resolution:

$$\hat{\mathbf{f}}(\hat{x}) = \mathrm{Resample}(\mathbf{f})(\hat{x}) = \sum_{x\in\Omega(\hat{x})} w(x)\,\mathbf{f}_{\mathrm{norm}}(x), \tag{2}$$

where $\mathrm{Resample}(\cdot)$ maps the flow field from the $H \times W$ pixel grid to the $\hat{H} \times \hat{W}$ latent grid by computing the weighted average displacement over the pixel footprint $\Omega(\hat{x})$ of each latent location, and $\mathbf{f}_{\mathrm{norm}}(x) = (\Delta u/W,\, \Delta v/H) \in [-1, 1]^2$ denotes the flow normalized by the image resolution.

### 3.1.3 Optical Flow Conditioned Parameterization

We then convert each flow vector into $K$ transformations by applying fixed random linear projections:

$$\mathbf{v}_k^{(n)}(\hat{x}) = \mathbf{A}_k\,\hat{\mathbf{f}}_{n-1\to n}(\hat{x}) + \mathbf{b}_k, \qquad k = 1, \ldots, K. \tag{3}$$

The flow at each latent location is two-dimensional, whereas a Householder reflector acting on the latent noise requires a non-zero vector in $\mathbb{R}^C$; the projections therefore lift each flow vector into $K$ distinct reflection directions in channel space, whose composition forms a $C \times C$ orthogonal transformation. The projections $\mathbf{A}_k \in \mathbb{R}^{C\times 2}$ and $\mathbf{b}_k \in \mathbb{R}^C$ are randomly initialized once and frozen throughout training and inference, introducing no learnable parameters and yielding a deterministic, reproducible mapping from flow to transformation. The only requirement on these parameters is $\mathbf{v}_k^{(n)}(\hat{x}) \neq \mathbf{0}$, since the Householder normalization is undefined for the zero vector; we use a small non-zero bias $\mathbf{b}_k$ to avoid such degenerate projections. No orthogonality constraint on $\mathbf{A}_k$ is needed, as orthogonality of the resulting operator is guaranteed by the Householder construction itself. Learning $\mathbf{A}_k$ and $\mathbf{b}_k$ is possible but unnecessary: their role is only to provide distinct deterministic embeddings of the flow into channel space. Because the projections are shared across spatial locations, the orthogonal transformation constructed below depends on $\hat{x}$ only through the local flow $\hat{\mathbf{f}}_{n-1\to n}(\hat{x})$. The projection parameters total only $K(2C + C) = 3KC$ values, independent of spatial resolution.

### 3.1.4 Per-Location Orthogonal Noise Transformation

The motion-embedded noise is constructed sequentially: the noise at each frame is obtained by applying a flow-conditioned orthogonal transformation to the noise of the preceding frame. Formally, for each frame $n \geq 1$ and latent location $\hat{x}$:

$$\tilde{\mathbf{z}}_n(\hat{x}) = \mathbf{Q}_n(\hat{x})\,\tilde{\mathbf{z}}_{n-1}(\hat{x}), \qquad \mathbf{Q}_n(\hat{x}) = \prod_{k=1}^{K} \mathbf{R}\Big(\mathbf{v}_k^{(n)}(\hat{x})\Big), \tag{4}$$

where frame 0 retains the unmodified Gaussian noise $\tilde{\mathbf{z}}_0(\hat{x}) = \mathbf{z}_0(\hat{x}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_C)$, and the reflection vectors $\mathbf{v}_k^{(n)}(\hat{x})$ are obtained from the resampled forward optical flow between frames $n-1$ and $n$ via Eq. (3).

Since each $\mathbf{Q}_n(\hat{x})$ is a product of Householder reflections and therefore orthogonal, and the product of orthogonal matrices is orthogonal, the noise at frame $n$ is equivalently a single orthogonal transformation of the initial noise:

$$\tilde{\mathbf{z}}_n(\hat{x}) = \mathbf{P}_n(\hat{x})\,\mathbf{z}_0(\hat{x}), \qquad \mathbf{P}_n(\hat{x}) = \mathbf{Q}_n(\hat{x}) \cdots \mathbf{Q}_1(\hat{x}). \tag{5}$$

This accumulated operator $\mathbf{P}_n$ is orthogonal by construction, which is the key property exploited in Section 3.2. The complete noise construction procedure is given in Algorithm 1.

**Number of Reflections and Expressivity.** Each Householder reflection $\mathbf{R}(\mathbf{v}) = \mathbf{I}_C - 2\bar{\mathbf{v}}\bar{\mathbf{v}}^\top$ is a rank-one perturbation of the identity, and the composed transformation $\mathbf{Q}_n$ is a full-rank orthogonal matrix for every $K \geq 1$. Expanding the product in Eq. (4) yields a series in the number of interacting reflections,

$$\mathbf{Q}_n(\hat{x}) = \mathbf{I}_C + \sum_{m=1}^{K} (-2)^m \sum_{k_1<\cdots<k_m} \Big(\prod_{j=1}^{m-1} \bar{\mathbf{v}}_{k_j}^\top \bar{\mathbf{v}}_{k_{j+1}}\Big)\, \bar{\mathbf{v}}_{k_1} \bar{\mathbf{v}}_{k_m}^\top, \tag{6}$$

**Algorithm 1** Motion-Embedded Noise via Householder Transformations

**Require:** Reference video $V = \{F_0, \ldots, F_{N-1}\}$, degradation level $\gamma \in [0, 1]$, number of reflections $K$, fixed random projections $\{\mathbf{A}_k, \mathbf{b}_k\}_{k=1}^{K}$

**Ensure:** Motion-embedded noise tensor $\tilde{Z}_\gamma$

$\mathbf{z}_0(\hat{x}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_C)$ for all $\hat{x}$; $\tilde{\mathbf{z}}_0(\hat{x}) \leftarrow \mathbf{z}_0(\hat{x})$

**Step 1: construct per frame noise**

**for** $n = 1, \ldots, N-1$ **do**

  **a) Estimate flows and resample**

  $\hat{\mathbf{f}}_{n-1\to n}(\hat{x}) \leftarrow \mathrm{Resample}(\Phi(F_{n-1}, F_n))$ ▷ estimate and align flow to latent grid

  **b) apply orthogonal transformation to noise of previous frame**

  $\mathbf{v}_k^{(n)}(\hat{x}) \leftarrow \mathbf{A}_k\, \hat{\mathbf{f}}_{n-1\to n}(\hat{x}) + \mathbf{b}_k, \quad k = 1, \ldots, K$

  $\mathbf{Q}_n(\hat{x}) \leftarrow \prod_{k=1}^{K} \mathbf{R}\Big(\mathbf{v}_k^{(n)}(\hat{x})\Big)$ ▷ per-location flow to orthogonal transformation

  $\tilde{\mathbf{z}}_n(\hat{x}) \leftarrow \mathbf{Q}_n(\hat{x})\, \tilde{\mathbf{z}}_{n-1}(\hat{x})$

**end for**

**Step 2: Stack and apply noise degradation**

$\tilde{Z} \leftarrow \big[\tilde{\mathbf{z}}_0, \ldots, \tilde{\mathbf{z}}_{N-1}\big]$; $\boldsymbol{\zeta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, same shape as $\tilde{Z}$

$$\tilde{Z}_\gamma \leftarrow \frac{(1-\gamma)\, \tilde{Z} + \gamma\, \boldsymbol{\zeta}}{\sqrt{(1-\gamma)^2 + \gamma^2}}$$

**return** $\tilde{Z}_\gamma$

where we omit the arguments $(n, \hat{x})$ of $\bar{\mathbf{v}}_k$. Analogous to a Taylor series, the order-$m$ term captures interactions among $m$ flow-dependent reflection directions and, through Eq. (3), is a rational function of the flow $\hat{\mathbf{f}}_{n-1\to n}(\hat{x})$ of degree $2m$; $K$ determines the highest order retained. Since each order-$m$ term is bounded in norm by $2^m \prod_j |\bar{\mathbf{v}}_{k_j}^\top \bar{\mathbf{v}}_{k_{j+1}}|$ and the random projections yield reflection directions with small pairwise inner products, higher-order terms are increasingly attenuated and the flow influences $\mathbf{Q}_n$ predominantly through the low-order terms. Adding reflections therefore yields diminishing gains in expressivity, while the cost grows linearly with $K$, consistent with the saturation observed in Table 5.

### 3.2 Theoretical Properties

The key theoretical guarantee of our noise construction is that the transformed noise remains exactly spatially Gaussian, regardless of the motion condition; a detailed proof and empirical results are given in Section E.1.

**Proposition 1** (Exact Spatial Gaussianity)**.** *Let $\tilde{\mathbf{z}}_0(\hat{x}) = \mathbf{z}_0(\hat{x}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_C)$ be i.i.d. across spatial locations, and let $\tilde{\mathbf{z}}_n(\hat{x}) = \mathbf{Q}_n(\hat{x})\, \tilde{\mathbf{z}}_{n-1}(\hat{x})$ for $n \geq 1$, where each $\mathbf{Q}_n(\hat{x})$ is a product of Householder reflections parameterized by the resampled optical flow $\hat{\mathbf{f}}_{n-1\to n}(\hat{x})$. Then $\tilde{\mathbf{z}}_n(\hat{x}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_C)$ for all $n$ and all $\hat{x}$, and the vectors $\tilde{\mathbf{z}}_n(\hat{x})$ are mutually independent across latent locations $\hat{x}$.*

**Temporal Correlation Structure.** While Proposition 1 concerns each frame in isolation, the sequential construction couples the frames. Using Eq. (5), the cross-frame covariance of the noise at a latent location $\hat{x}$ is, for $n > m$,

$$\mathrm{Cov}[\tilde{\mathbf{z}}_n(\hat{x}),\, \tilde{\mathbf{z}}_m(\hat{x})] = \mathbf{P}_n(\hat{x})\, \mathbf{P}_m(\hat{x})^\top = \mathbf{Q}_n(\hat{x}) \cdots \mathbf{Q}_{m+1}(\hat{x}), \tag{7}$$

while noise at different locations remains uncorrelated (Section E.2). Equation (7) shows that the dependence between two frames is an orthogonal $C \times C$ matrix determined entirely by the optical flow at $\hat{x}$ between frames $m$ and $n$: the noise vector at frame $n$ is a flow-dependent orthogonal transformation of the one at frame $m$. Because the projections $\{\mathbf{A}_k, \mathbf{b}_k\}$ are shared across locations, latent locations undergoing the same motion receive the same cross-frame transformation, whereas locations with different motion receive different ones, so the spatial pattern of these transformations mirrors the motion field. The denoising model (the CogVideoX DiT) is trained with this structured noise in place of i.i.d. noise and learns to exploit it as an implicit motion-conditioning signal. A geometric comparison with spatial noise warping is given in Section 5.

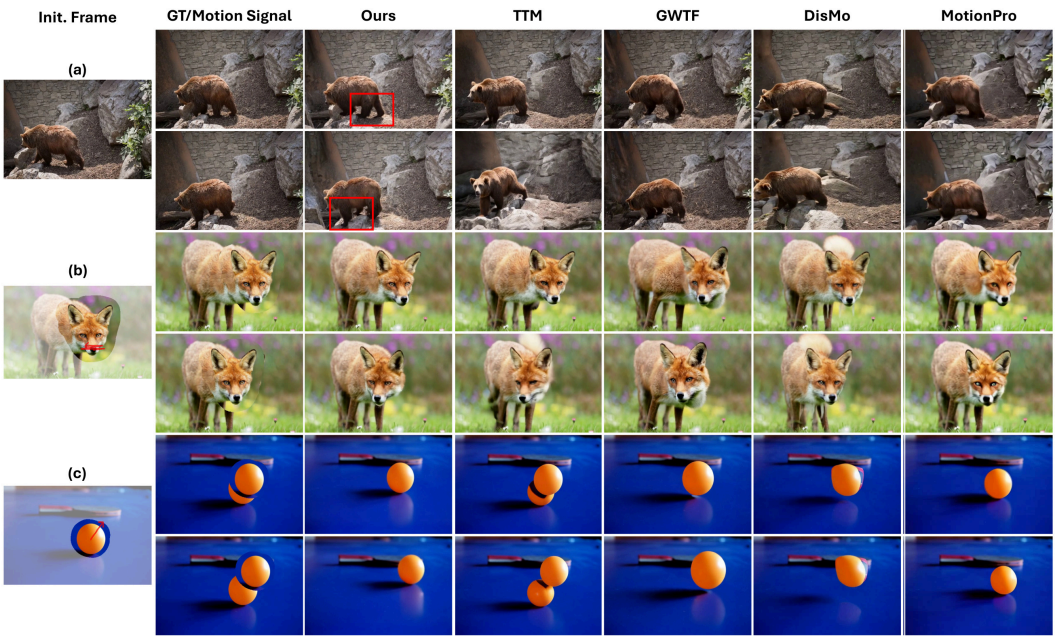



Figure 3: **Qualitative comparison of (a) motion-transfer, (b) fine-grained motion control, and (c) local object motion control I2V generation, evaluated on DAVIS and MC-Bench.** We present the intermediate frame and last frame of generated videos by each method. Our method demonstrates strong appearance consistency and high generation fidelity while faithfully adhering to the provided motion control signals. **Best viewed zoomed in; videos are provided in the Supplementary Material.**

### 3.3 Fine-tuning with Motion-Embedded Noise

The key insight is that the motion conditioning is carried entirely by the cross-frame covariance structure of $\tilde{Z}$, not by its marginal distribution. Since each frame's noise is marginally standard Gaussian (Proposition 1), the model trains with the standard diffusion objective without modification. The structured inter-frame dependencies, encoded by $\mathbf{Q}_n(\hat{x})$, act as an implicit motion signal that the denoiser learns to exploit during fine-tuning. More training details are provided in Section C.

## 4 Experiments

### 4.1 Implementation Details

We use the pretrained CogVideoX-5B-I2V as our base I2V model [59]. We fine-tune the base video diffusion model on 100K videos from RealCam-Vid [66], curated from multiple sources including RealEstate10K [68], MiraData [25], and DL3DV-10K [31], using optical flows estimated by RAFT [47] for training. All experiments are conducted at a spatial resolution of $480 \times 720$ with 49 frames per video; the latent channel dimension is $C = 16$, determined by the CogVideoX VAE. We follow GWTF [7] to fine-tune the model for 30,000 iterations using a rank-2048 LoRA setup (learning rate $1 \times 10^{-5}$, batch size 8), running on 4 NVIDIA A100 80GB GPUs for a total of 40 GPU-days.

### 4.2 Video Generation with Motion Control

**Baselines and Evaluation Metrics.** We evaluate **HME** on motion transfer, fine-grained motion control, object-level motion control and camera motion control for I2V generation. The performances on DAVIS [39], MC-Bench [65] are displayed in Table 1 and Table 2, respectively. Following GWTF, we adopt VBench [23] to assess the perceptual quality of generated video with related metrics such as Subject Consistency, Background Consistency, Temporal Flickering, and Motion Smoothness. To

Table 1: Quantitative comparison of motion controlled I2V generation on DAVIS. Best performance in **Bold**, best baseline performance in Underline.

| Method | Subject Consist. ↑ | BG Consist. ↑ | Temporal Flicker. ↑ | Motion Smooth. ↑ | FID ↓ | FVD ($\times 10^3$) ↓ | Optical Flow (EPE) ↓ | CoTracker (mIoU) ↑ | PSNR (dB) ↑ | CLIP Image ↑ |
|---|---|---|---|---|---|---|---|---|---|---|
| MotionClone [32] | 0.754 | 0.862 | 0.874 | 0.928 | 133.852 | 1.771 | 26.451 | 0.373 | 10.032 | 0.814 |
| ImageConductor [30] | 0.791 | 0.891 | 0.913 | 0.939 | 128.087 | 1.830 | 20.382 | 0.372 | 12.117 | 0.811 |
| GWTF$_{\gamma=0.5}$ [7] | 0.888 | 0.920 | 0.942 | 0.978 | 68.385 | 0.987 | 5.644 | 0.464 | 14.748 | 0.912 |
| MotionPro [65] | 0.873 | 0.913 | 0.932 | 0.964 | 103.859 | 1.253 | 19.692 | 0.473 | 14.257 | 0.879 |
| DisMo [43] | 0.851 | 0.903 | 0.920 | 0.956 | 85.201 | 1.156 | 11.189 | 0.428 | 11.475 | 0.882 |
| TTM [46] | 0.869 | 0.916 | 0.935 | 0.970 | 70.658 | 1.093 | 13.871 | 0.412 | 13.942 | 0.901 |
| Ours$_{\gamma=0.1,K=2}$ | 0.896 | 0.924 | **0.951** | **0.979** | 65.343 | **0.830** | 4.816 | 0.540 | **15.662** | 0.920 |
| Ours$_{\gamma=0.3,K=2}$ | **0.898** | 0.925 | 0.946 | **0.979** | **61.780** | 0.862 | **4.433** | 0.546 | 15.399 | **0.923** |
| Ours$_{\gamma=0.5,K=2}$ | **0.898** | **0.926** | 0.945 | **0.979** | 62.799 | 0.855 | 4.486 | **0.548** | 15.162 | **0.923** |

Table 2: Quantitative comparison of Fine-grained and Object-level motion control on MC-Bench.

| **Method** | **Subject Cons. ↑** | **Background Cons. ↑** | **Motion Smooth. ↑** | **Aesthetic Qual. ↑** | **Imaging Qual. ↑** | **CTD ↓** | **Bg-Obj CTD ↑** |
|---|---|---|---|---|---|---|---|
| MotionPro [65] | 0.980 | 0.976 | **0.995** | 0.586 | 0.610 | **8.019** | 23.30 |
| TTM [46] | 0.984 | **0.981** | **0.995** | 0.601 | **0.645** | 8.432 | 36.32 |
| DisMo [43] | **0.985** | 0.973 | 0.991 | **0.608** | 0.601 | 8.954 | 36.16 |
| *Noise-based methods* | | | | | | | |
| GWTF$_{\gamma=0.5}$ [7] | 0.975 | 0.974 | 0.993 | 0.589 | **0.645** | 15.747 | **42.44** |
| Ours$_{\gamma=0.3,K=2}$ | 0.976 | 0.977 | **0.995** | 0.594 | **0.645** | 13.879 | 39.53 |

assess the motion adherence, we evaluate 1) CoTracker Distance and Background CoTracker distance for MC-Bench samples; 2) Optical Flow EPE, CoTracker mIoU, and PSNR against the reference videos for DAVIS samples. The input for all evaluations is a single reference image paired with a motion signal obtained from either the optical flow estimated from a reference video or derived from a user-dragged mask on the initial frame. We use the resulting forward optical flows between frames to construct the motion-embedded noise $\tilde{Z}_\gamma$ via Algorithm 1 (details in Section D).

**Results on Motion Transfer.** We evaluate motion transfer I2V on 45 DAVIS videos, using the motion of each original video as reference. As shown in Table 1, we compare against six baselines: training-free methods (MotionClone [32], TTM [46]), noise-warping (GWTF), and supervised controllers (ImageConductor [30], MotionPro [65], DisMo [43]). Each of our three variants ($\gamma \in \{0.1, 0.3, 0.5\}$) outperforms all six baselines on all ten metrics, despite being fine-tuned on 100K videos, compared with approximately 4M for GWTF. It attains the highest scores on all four VBench metrics, as well as the best FID and FVD. At $\gamma = 0.5$, Optical Flow EPE decreases from 5.644 to 4.486, CoTracker mIoU increases from 0.464 to 0.548, and PSNR improves from 14.748 to 15.162 dB. These gains require no post-hoc Gaussianity correction, indicating that motion embedding via orthogonal transformations is a more efficient alternative to spatial noise warping. We further evaluate across varying noise degradation levels, where all variants consistently outperform GWTF and other baselines.

**Results on Fine-grained and Object-level Control.** We evaluate object-level and fine-grained motion control on 965 MC-Bench samples against MotionPro, TTM, DisMo, and GWTF. Our approach outperforms GWTF on five metrics, achieving lower CTD and better Background Consistency and Motion Smoothness, indicating that Householder motion embedding captures motion priors more faithfully than spatial warping. The CTD gap with supervised methods reflects a trade-off between controllability and generative freedom: TTM, MotionPro, and DisMo enforce strict trajectory adherence via explicit supervision but degrade subject-background disentanglement, evidenced by lower Bg-Obj CTD (23.30, 36.32, 36.16) relative to our method (39.53) and GWTF (42.44). Our method also matches the best Motion Smoothness (0.995) without architectural changes, showing that HME is competitive with supervised motion control while better preserving subject–background disentanglement.

**Results on Camera Motion Control.** We evaluate **HME** on camera controlled I2V generation under pre-defined camera trajectories using 200 randomly sampled scenes from the DL3DV subset of the RealCam-Vid validation set. Frame-level alignment is assessed via PSNR, LPIPS, and SSIM, and camera motion consistency via Normalized Absolute Trajectory Error (ATE). As shown in Table 3, our method achieves the best performance across all seven metrics, outperforming noise-warping and other baselines. The leading ATE score confirms accurate replication of the target camera trajectory, while the superior PSNR, LPIPS, and SSIM scores demonstrate that our method maintains high visual




Figure 4: **Qualitative Comparison of camera motion controlled I2V generation, evaluated on DL3DV.** We present the intermediate frames and last frames of generated videos by each method. **Best viewed zoomed in; videos are provided in the Supplementary Material.**

Table 3: Quantitative comparison of DL3DV camera motion control.

| Method | FID ↓ | FVD ↓ | PSNR (dB) ↑ | CLIP Image ↑ | LPIPS ↓ | SSIM ↑ | ATE ↓ |
|---|---|---|---|---|---|---|---|
| MotionPro [65] | 37.195 | 291.510 | 12.701 | 0.814 | 0.545 | 0.470 | 0.825 |
| DisMo [43] | 22.458 | <u>187.134</u> | 12.255 | 0.876 | 0.506 | 0.446 | 1.155 |
| TTM [46] | <u>21.367</u> | 196.533 | <u>12.764</u> | <u>0.881</u> | 0.443 | 0.467 | <u>0.464</u> |
| GWTF$_{\gamma=0.5}$ [7] | 23.394 | 235.260 | 12.626 | 0.880 | <u>0.431</u> | <u>0.469</u> | 0.473 |
| Ours$_{\gamma=0.3,K=2}$ | **20.338** | **175.103** | **13.656** | **0.885** | **0.402** | **0.496** | **0.421** |

fidelity alongside strong motion adherence, reflecting a great balance between appearance quality and temporal consistency under explicit camera control.

**Qualitative Results.** In Fig. 3, we present a qualitative comparison between our method and all competing baselines on (a) motion transfer, (b) fine-grained, and (c) object-level motion controlled video generation. For (a), the red box highlights that our method more faithfully replicates the detailed motion of the bear across frames, accurately capturing the displacement of the subject in accordance with the reference motion signal, whereas competing methods fail to preserve fine motions. For (b), our method not only follows the prescribed motion signal faithfully but also maintains high visual fidelity, producing coherent details throughout the sequence, while other baselines exhibit visible artifacts. For (c), our method demonstrates precise spatial awareness by correctly animating the target ball along the intended trajectory while preserving the relative positioning between objects, whereas competing methods either lose track of the motion signal or produce spatially inconsistent object placements across frames.

We further show camera-motion-controlled I2V results in Fig. 4. Compared with the strongest baseline, TTM, our generated videos yield estimated camera trajectories closer to the target trajectory, indicating stronger camera-motion adherence while better preserving viewpoint changes, scene structure, and appearance consistency. Additional results are provided in Section F.

**Robustness to the Flow Estimator.** To evaluate robustness to realistic optical-flow inputs, we replace RAFT with UniMatch at evaluation time only, while our model remains trained on RAFT flows; GWTF and our method receive identical UniMatch flows. Despite this mismatch, our method remains close to its RAFT results and still outperforms GWTF on all ten metrics (Table 4), especially in Flow EPE (4.707 against 5.857).

### 4.3 Ablation and Analysis

**Choice of the Number of Householder Reflections.** We ablate the number of Householder reflections $K \in \{1, 2, 4, 8, 12\}$ in Table 5. Motion adherence is weakest at $K = 1$ (Flow EPE $4.868$) and improves with diminishing returns as $K$ grows (Flow EPE $4.433 \rightarrow 4.144$ from $K = 2$ to $K = 8$; CoTracker mIoU $0.558$ and PSNR $15.581$ dB at $K = 12$), consistent with the attenuation of higher-order terms in Eq. (6), while runtime grows approximately linearly, from $2.27$ ms/latent at $K = 1$ to $14.78$ ms/latent at $K = 12$ (Table 6). At $K = 2$, our method runs at $\mathbf{2.80}$ ms/latent, a $2.7\times$ speedup over GWTF and $13.6\times$ over HIWYN. It is also memory-light: only $3KC$ projection values are stored, shared across frames and locations, and neither dense per-location $C \times C$ matrices nor noise upsampling are needed, so the peak additional memory is $0.5$–$6$ MB, versus $76$ MB for GWTF, $285$ MB for InfRes and $49{,}384$ MB for HIWYN. We use $K = 2$ as the default: it achieves the best FID with only a small latency increase over $K = 1$, while larger $K$ offers slightly stronger motion adherence at higher cost.

Table 4: Robustness to the flow estimator: DAVIS results with UniMatch [56] flows at evaluation time.

| Method | Subject Consist. ↑ | BG Consist. ↑ | Temporal Flicker. ↑ | Motion Smooth. ↑ | FID ↓ | FVD ($\times 10^3$) ↓ | Optical Flow (EPE) ↓ | CoTracker (mIoU) ↑ | PSNR (dB) ↑ | CLIP Image ↑ |
|---|---|---|---|---|---|---|---|---|---|---|
| GWTF$_{\gamma=0.5}$ [7] | 0.882 | 0.918 | 0.945 | **0.979** | 73.348 | 1.180 | 5.857 | 0.472 | 14.437 | 0.901 |
| Ours$_{\gamma=0.3,K=2}$ | **0.895** | **0.926** | **0.948** | **0.979** | **63.619** | **0.837** | **4.707** | **0.541** | **15.418** | **0.915** |

Table 5: Performance comparison across different numbers of Householder reflections $K$ (all with $\gamma = 0.3$), on DAVIS.

| **Method** | **Subject ↑** | **Background ↑** | **Temporal ↑** | **Motion ↑** | **FID ↓** | **FVD ↓** | **Flow EPE ↓** | **CoTracker mIoU ↑** | **PSNR ↑** | **CLIP Image ↑** |
|---|---|---|---|---|---|---|---|---|---|---|
| Ours$_{K=1}$ | **0.898** | 0.925 | 0.946 | 0.978 | 63.992 | 861.121 | 4.868 | 0.543 | 15.300 | 0.918 |
| Ours$_{K=2}$ | **0.898** | 0.925 | 0.946 | **0.979** | **61.780** | 861.740 | 4.433 | 0.546 | 15.399 | **0.923** |
| Ours$_{K=4}$ | **0.898** | 0.926 | 0.948 | **0.979** | 64.579 | **821.749** | 4.358 | 0.555 | 15.424 | 0.922 |
| Ours$_{K=8}$ | **0.898** | **0.927** | 0.948 | **0.979** | 63.314 | 827.664 | **4.144** | 0.550 | 15.512 | **0.923** |
| Ours$_{K=12}$ | **0.898** | **0.927** | **0.950** | **0.979** | 64.699 | 831.586 | 4.197 | **0.558** | **15.581** | **0.923** |

Table 6: Computational overhead of noise-based methods: GPU time and approximate peak additional GPU memory on an NVIDIA RTX PRO 6000 Blackwell GPU (1024×1024 resolution, $C = 16$; HIWYN with $L = 3$).

| **Method** | HIWYN [8] | GWTF [7] | InfRes [10] | Ours$_{K=1}$ | Ours$_{K=2}$ | Ours$_{K=4}$ | Ours$_{K=6}$ | Ours$_{K=8}$ | Ours$_{K=12}$ |
|---|---|---|---|---|---|---|---|---|---|
| **GPU Time (ms/Latent) ↓** | 38.00 | 7.63 | 23.94 | **2.27** | 2.80 | 5.18 | 7.59 | 10.01 | 14.78 |
| **Memory Overhead (MB) ↓** | 49,384 | 76 | 285 | **0.5** | 1 | 2 | 3 | 4 | 6 |

**Choice of Noise Degradation Level.** The degradation parameter $\gamma$ trades motion adherence against generative freedom. All three variants, $\gamma \in \{0.1, 0.3, 0.5\}$ at $K = 2$, outperform GWTF on all ten metrics (Table 1); lower $\gamma$ strengthens motion adherence, while higher $\gamma$ favours perceptual quality and subject consistency, with small differences overall. We adopt $\gamma = 0.3$ for all other experiments; qualitative comparisons are in Section G.

## 5 Discussion

**Geometric Analysis of Temporal Noise Correlation.** PYoCo [13] shows that correlating noise across frames promotes temporal coherence. GWTF [7] realises this by transporting noise along flow trajectories *across* locations, whereas our noise stays tied to its location: $\tilde{\mathbf{z}}_n(\hat{x}) = \mathbf{P}_n(\hat{x})\,\mathbf{z}_0(\hat{x})$ preserves its norm, and its change between frames, $(\mathbf{Q}_n(\hat{x}) - \mathbf{I}_C)\,\tilde{\mathbf{z}}_{n-1}(\hat{x})$, depends only on the local flow. Since the projections are shared across locations, regions with the same motion receive the same transformation; this includes static regions, since $\mathbf{Q}_n \neq \mathbf{I}_C$ for zero flow when $\mathbf{b}_k \neq \mathbf{0}$. We hypothesise that this per-location encoding of motion in channel space provides a more direct conditioning signal than spatial transport, contributing to the improved motion adherence over GWTF.

**Limitations.** Despite its effectiveness, our method has several limitations. The choice of the number of reflections $K$ introduces an expressivity-efficiency trade-off and may require tuning for different settings. The method also relies on accurate optical flow for training, so noisy motion estimates can weaken motion adherence. Relatedly, since the motion signal is carried by per-location flow, subjects undergoing complex non-rigid geometric transformations (e.g., the rapidly rotating snowboarder in DAVIS) remain challenging, which we leave for future work. Moreover, while the noise construction is training-free and parameter-free, the video diffusion model still requires fine-tuning to fully exploit the motion-embedded noise. Finally, the baselines are evaluated with their released checkpoints rather than retrained on our training data and RAFT flows.

## 6 Conclusion

We presented **Householder Motion Embedding (HME)**, a noise-modulation framework for motion-controllable image-to-video generation. By replacing spatial noise transport with per-location Householder orthogonal transformations in latent channel space, our method preserves spatial Gaussianity by construction while encoding motion-aligned temporal correlations. Experiments on motion transfer, object-level control, and camera motion control show that **HME** improves motion adherence, visual fidelity, and noise-processing efficiency over existing noise-warping methods. Beyond noise construction, we believe that geometric transformation-based conditioning offers a promising way for incorporating structured motion priors into video diffusion models.

## Acknowledgments and Disclosure of Funding

This work was conducted in collaboration with Vertex Lab Technology Limited. This research is funded in part by an Australian Research Council (ARC) Discovery Grant DP220100800. Hongdong Li holds concurrent appointments with The Australian National University (full-time) and with Amazon as an Amazon Scholar (part-time).

## References

[1] Luca Savant Aira, Antonio Montanaro, Emanuele Aiello, Diego Valsesia, and Enrico Magli. Motioncraft: Physics-based zero-shot video generation. In *NeurIPS*, 2024.

[2] Alejandro Moreno Arcas, Albert Sanchis, Jorge Civera, and Alfons Juan. Hoft: Householder orthogonal fine-tuning. *arXiv preprint arXiv:2505.16531*, 2025.

[3] Sherwin Bahmani, Ivan Skorokhodov, Aliaksandr Siarohin, Willi Menapace, Guocheng Qian, Michael Vasilkovsky, Hsin-Ying Lee, Chaoyang Wang, Jiaxu Zou, Andrea Tagliasacchi, David B. Lindell, and Sergey Tulyakov. VD3D: Taming large video diffusion transformers for 3D camera control. In *ICLR*, 2025. URL https://openreview.net/forum?id=0n4bS0R5MM.

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

[5] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your Latents: High-Resolution Video Synthesis with Latent Diffusion Models. In *CVPR*, 2023.

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

[7] Ryan Burgert, Yuancheng Xu, Wenqi Xian, Oliver Pilarski, Pascal Clausen, Mingming He, Li Ma, Yitong Deng, Lingxiao Li, Mohsen Mousavi, et al. Go-with-the-flow: Motion-controllable video diffusion models using real-time warped noise. In *CVPR*, pages 13–23, 2025.

[8] Pascal Chang, Jingwei Tang, Markus Gross, and Vinicius C Azevedo. How i warped your noise: a temporally-correlated noise prior for diffusion models. In *ICLR*. OpenReview, 2024.

[9] Weifeng Chen, Yatai Ji, Jie Wu, Hefeng Wu, Pan Xie, Jiashi Li, Xin Xia, Xuefeng Xiao, and Liang Lin. Control-A-Video: Controllable Text-to-Video Diffusion Models with Motion Prior and Reward Feedback Learning. *arXiv preprint arXiv:2305.13840*, 2023.

[10] Yitong Deng, Winnie Lin, Lingxiao Li, Dmitriy Smirnov, Ryan D Burgert, Ning Yu, Vincent Dedun, and Mohammad H. Taghavi. Infinite-resolution integral noise warping for diffusion models. In *ICLR*, 2025. URL https://openreview.net/forum?id=Y6LPWBo2HP.

[11] Wei Dong, Yuan Sun, Yiting Yang, Xing Zhang, Zhijun Lin, Qingsen Yan, Haokui Zhang, Peng Wang, Yang Yang, and Hengtao Shen. Efficient adaptation of pre-trained vision transformer via householder transformation. *NeurIPS*, 37:102056–102077, 2024.

[12] Yu Gao, Haoyuan Guo, Tuyen Hoang, Weilin Huang, Lu Jiang, Fangyuan Kong, Huixia Li, Jiashi Li, Liang Li, Xiaojie Li, et al. Seedance 1.0: Exploring the boundaries of video generation models. *arXiv preprint arXiv:2506.09113*, 2025.

[13] Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve Your Own Correlation: A Noise Prior for Video Diffusion Models. In *ICCV*, 2023.

[14] 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 *CVPR*, pages 1–12, 2025.

[15] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. In *ICLR*, 2024. URL `https://arxiv.org/abs/2307.10373`.

[16] Ahmet Berke Gökmen, Yiğit Ekin, Bahri Batuhan Bilecen, and Aysegul Dundar. Ropecraft: Training-free motion transfer with trajectory-guided rope optimization on diffusion transformers. In *NeurIPS*, 2025.

[17] Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.

[18] Zekai Gu et al. Diffusion as shader. In *SIGGRAPH*, 2025.

[19] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In *ICLR*, 2024.

[20] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. CameraCtrl: Enabling camera control for video diffusion models. In *ICLR*, 2025.

[21] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video Diffusion Models. In *NeurIPS*, 2022.

[22] Emiel Hoogeboom, Rianne van den Berg, and Max Welling. Emerging convolutions for generative normalizing flows. In *ICML*, pages 2771–2780. PMLR, 2019.

[23] 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 *CVPR*, 2024.

[24] Yash Jain, Anshul Nasery, Vibhav Vineet, and Harkirat Behl. Peekaboo: Interactive video generation via masked-diffusion. In *CVPR*, 2024.

[25] Xuan Ju, Yiming Gao, Zhaoyang Zhang, Ziyang Yuan, Xintao Wang, Ailing Zeng, Yu Xiong, Qiang Xu, and Ying Shan. Miradata: A large-scale video dataset with long durations and structured captions. *NeurIPS*, 37:48955–48970, 2024.

[26] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *ECCV*, 2024. URL `https://arxiv.org/abs/2307.07635`.

[27] 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.

[28] Max Ku, Cong Wei, Weiming Ren, Huan Yang, and Wenhu Chen. AnyV2V: A tuning-free framework for any video-to-video editing tasks. *TMLR*, 2024.

[29] Zhengfei Kuang, Shengqu Cai, Hao He, Yinghao Xu, Hongsheng Li, Leonidas Guibas, and Gordon Wetzstein. Collaborative video diffusion: Consistent multi-video generation with camera control. In *NeurIPS*, 2024.

[30] Yaowei Li, Xintao Wang, Zhaoyang Zhang, Zhouxia Wang, Ziyang Yuan, Liangbin Xie, Ying Shan, and Yuexian Zou. Image conductor: Precision control for interactive video synthesis. In *AAAI*, volume 39, pages 5031–5038, 2025.

[31] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *CVPR*, pages 22160–22169, 2024.

[32] Pengyang Ling, Jiazi Bu, Pan Zhang, Xiaoyi Dong, Yuhang Zang, Tong Wu, Huaian Chen, Jiaqi Wang, and Yi Jin. MotionClone: Training-free motion cloning for controllable video generation. In *ICLR*, 2025. URL https://openreview.net/forum?id=aY3L65HgHJ.

[33] Haiping Liu, Lijing Lin, Jingyuan Sun, Zhegong Shangguan, Mauricio A Alvarez, and Hongpeng Zhou. Rethinking rope: A mathematical blueprint for n-dimensional positional embedding. *arXiv preprint arXiv:2504.06308*, 2025.

[34] Zakaria Mhammedi, Andrew Hellicar, Ashfaqur Rahman, and James Bailey. Efficient orthogonal parametrisation of recurrent neural networks using householder reflections. In *ICML*, pages 2401–2409. PMLR, 2017.

[35] Chong Mou, Mingdeng Cao, Xintao Wang, Zhaoyang Zhang, Ying Shan, and Jian Zhang. Revideo: Remake a video with motion and content control. *arXiv*, 2024.

[36] Koichi Namekata, Sherwin Bahmani, Ziyi Wu, Yash Kant, Igor Gilitschenski, and David B. Lindell. SG-I2V: Self-guided trajectory control in image-to-video generation. In *ICLR*, 2025. URL https://openreview.net/forum?id=uQjySppU9x.

[37] Muyao Niu, Xiaodong Cun, Xintao Wang, Yong Zhang, Ying Shan, and Yinqiang Zheng. MOFA-Video: Controllable Image Animation via Generative Motion Field Adaptions in Frozen Image-to-Video Diffusion Model. In *ECCV*, 2024.

[38] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

[39] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016.

[40] Alexander Pondaven, Aliaksandr Siarohin, Sergey Tulyakov, Philip Torr, and Fabio Pizzati. Video motion transfer with diffusion transformers. In *CVPR*, 2025.

[41] Zeju Qiu, Weiyang Liu, Haiwen Feng, Yuxuan Xue, Yao Feng, Zhen Liu, Dan Zhang, Adrian Weller, and Bernhard Schölkopf. Controlling text-to-image diffusion by orthogonal finetuning. *NeurIPS*, 36:79320–79362, 2023.

[42] Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *CVPR*, pages 6121–6132, 2025.

[43] Thomas Ressler-Antal, Frank Fundel, Malek Ben Alaya, Stefan Andreas Baumann, Felix Krause, Ming Gui, and Björn Ommer. Dismo: Disentangled motion representations for open-world motion transfer. In *NeurIPS*, 2025.

[44] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

[45] Xiaoyu Shi, Zhaoyang Huang, Fu-Yun Wang, Weikang Bian, Dasong Li, Yi Zhang, Manyuan Meng, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling. In *SIGGRAPH*, 2024. URL https://arxiv.org/abs/2401.15977.

[46] Assaf Singer, Noam Rotstein, Amir Mann, Ron Kimmel, and Or Litany. Time-to-Move: Training-free motion-controlled video generation via dual-clock denoising. In *ICLR*, 2026.

[47] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pages 402–419, 2020. URL https://arxiv.org/abs/2003.12039.

[48] Jakub M Tomczak and Max Welling. Improving variational auto-encoders using householder flow. *arXiv preprint arXiv:1611.09630*, 2016.

[49] 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.

[50] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion controllability. In *NeurIPS*, 2024.

[51] Yiming Wang, Qihang Zhang, Shengqu Cai, Tong Wu, Jan Ackermann, Zhengfei Kuang, Yang Zheng, Frano Rajič, Siyu Tang, and Gordon Wetzstein. BulletTime: Decoupled control of time and camera pose for video generation. In *CVPR*, 2026.

[52] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. MotionCtrl: A unified and flexible motion controller for video generation. In *SIGGRAPH*, pages 1–11, 2024.

[53] Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, David Junhao Zhang, Mike Zheng Shou, Yan Li, Tingting Gao, and Di Zhang. DragAnything: Motion control for anything using entity representation. In *ECCV*, 2024.

[54] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Xintao Wang, Tien-Tsin Wong, and Ying Shan. DynamiCrafter: Animating Open-domain Images with Video Diffusion Priors. In *ECCV*, 2024.

[55] Dejia Xu, Weili Nie, Chao Liu, Sifei Liu, Jan Kautz, Zhangyang Wang, and Arash Vahdat. Camco: Camera-controllable 3d-consistent image-to-video generation. *arXiv preprint arXiv:2406.02509*, 2024.

[56] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *TPAMI*, 45(11):13941–13958, 2023.

[57] Shiyuan Yang, Liang Hou, Haibin Huang, Chongyang Ma, Pengfei Wan, Di Zhang, Xiaodong Chen, and Jing Liao. Direct-a-video: Customized video generation with user-directed camera movement and object motion. In *SIGGRAPH*, 2024.

[58] Songlin Yang, Yikang Shen, Kaiyue Wen, Shawn Tan, Mayank Mishra, Liliang Ren, Rameswar Panda, and Yoon Kim. Path attention: Position encoding via accumulating householder transformations. In *NeurIPS*, 2025.

[59] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Dong, et al. CogVideoX: Text-to-video diffusion models with an expert transformer. In *ICLR*, 2025. URL `https://arxiv.org/abs/2408.06072`.

[60] Danah Yatim, Rafail Fridman, Omer Bar-Tal, Yoni Kasten, and Tali Dekel. Space-time diffusion features for zero-shot text-driven motion transfer. In *CVPR*, 2024.

[61] Shengming Yin, Chenfei Wu, Jian Liang, Jie Shi, Houqiang Li, Gong Ming, and Nan Duan. DragNUWA: Fine-Grained Control in Video Generation by Integrating Text, Image, and Trajectory. *arXiv preprint arXiv:2308.08089*, 2023.

[62] Shen Yuan, Haotian Liu, and Hongteng Xu. Bridging the gap between low-rank and orthogonal adaptation via householder reflection adaptation. *NeurIPS*, 37:113484–113518, 2024.

[63] David Junhao Zhang, Roni Paiss, Shiran Zada, Nikhil Karnad, David E. Jacobs, Yael Pritch, Inbar Mosseri, Mike Zheng Shou, Neal Wadhwa, and Nataniel Ruiz. ReCapture: Generative video camera controls for user-provided videos using masked video fine-tuning. In *CVPR*, 2025.

[64] Yabo Zhang, Yuxiang Wei, Dongsheng Jiang, Xiaopeng Zhang, Wangmeng Zuo, and Qi Tian. ControlVideo: Training-Free Controllable Text-to-Video Generation. In *ICLR*, 2024.

[65] Zhongwei Zhang, Fuchen Long, Zhaofan Qiu, Yingwei Pan, Wu Liu, Ting Yao, and Tao Mei. Motionpro: A precise motion controller for image-to-video generation. In *CVPR*, pages 27957–27967, 2025.

[66] Guangcong Zheng, Teng Li, Xianpan Zhou, and Xi Li. Realcam-vid: High-resolution video dataset with dynamic scenes and metric-scale camera movements. *arXiv preprint arXiv:2504.08212*, 2025.

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

[68] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM TOG (Proc. SIGGRAPH)*, 37 (4), 2018.

# A Other Related Work

**Image-to-Video Diffusion Models.** Diffusion-based video generation has progressed rapidly from early UNet-based architectures [21, 5, 4, 19] that append temporal attention layers to pretrained image diffusion backbones, to large-scale Diffusion Transformer (DiT) models [38] that jointly model spatial and temporal dependencies through full spatio-temporal attention [59, 27, 49, 6]. Image-to-video (I2V) models such as Stable Video Diffusion [4] and CogVideoX [59] extend this capability by conditioning generation on a reference frame, enabling content-consistent animation from a single image. Despite their architectural differences, both UNet-based and DiT-based video diffusion models are initialized from Gaussian noise whose correlation across frames can be influential to the temporal consistency of the generated output [13]. Our method exploits this property by modulating the noise with optical flow to induce structured temporal correlations before denoising begins, without modifying the denoising architecture.

**Application of Householder Reflections in Deep Learning.** Householder reflections provide a numerically stable and efficient parameterization of orthogonal operators, with a growing range of applications in deep learning. Early work uses them to enforce orthogonality in RNN transition matrices as a remedy for vanishing and exploding gradients [34]. In generative modeling, products of Householder reflections serve as volume-preserving transformations in normalizing flows for VAEs [48, 22]. More recently, they have been adopted for parameter-efficient fine-tuning, where orthogonal weight updates preserve hyper-spherical energy during adaptation [41, 62, 2, 11], and for positional encoding, where data-dependent Householder products replace static RoPE rotations to improve positional expressiveness [58, 33]. In all of these settings, Householder products operate on weight matrices, transition operators, or attention mechanisms. Our work applies them to the diffusion noise, where orthogonality guarantees that each frame's noise remains exactly i.i.d. Gaussian after noise modulation.

# B Preliminaries

**Latent Video Diffusion.** We build on CogVideoX [59], which encodes a video into a latent $X_0 \in \mathbb{R}^{\hat{N}\times C\times \hat{H}\times \hat{W}}$ with a 3D VAE and learns to reverse the forward process $X_t = \alpha_t X_0 + \sigma_t \boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, where $\alpha_t^2 + \sigma_t^2 = 1$. A DiT denoiser $D_\theta(X_t, t, c)$, conditioned on the first frame and the text prompt $c$, is trained with the $v$-prediction objective

$$\mathcal{L} = \mathbb{E}_{X_0,\boldsymbol{\epsilon},t} \left\| D_\theta(X_t, t, c) - (\alpha_t \boldsymbol{\epsilon} - \sigma_t X_0) \right\|^2, \tag{8}$$

and sampling starts from $X_T = \boldsymbol{\epsilon}$. Noise-based motion-control methods, including noise warping and ours, replace $\boldsymbol{\epsilon}$ by structured noise both in the training objective and as the initial noise at inference, leaving the architecture and loss unchanged.

# C Fine-tuning with Motion-Embedded Noise

**Training Procedure.** For each training video clip, we estimate pairwise forward optical flows with RAFT [47], construct $\tilde{Z}_\gamma$ via Algorithm 1, and replace the standard i.i.d. initialization with $\tilde{Z}_\gamma$. The denoising objective and all other aspects of the training pipeline remain unchanged. At inference, the same noise construction is applied from a reference motion source, which can be obtained from a reference video, a predefined camera trajectory, or user-specified drag interactions on the initial frame.

**Noise Degradation.** Following GWTF [7], we interpolate between fully structured and fully unstructured noise via a degradation parameter $\gamma \in [0, 1]$:

$$\tilde{Z}_\gamma = \frac{(1-\gamma)\,\tilde{Z} + \gamma\,\boldsymbol{\zeta}}{\sqrt{(1-\gamma)^2 + \gamma^2}}, \qquad \boldsymbol{\zeta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{9}$$

where $\boldsymbol{\zeta}$ has the same shape as $\tilde{Z}$, $\gamma \to 0$ retains full motion conditioning and $\gamma \to 1$ approaches standard Gaussian noise. Since $\boldsymbol{\zeta}$ is drawn independently of $\tilde{Z}$ and the denominator normalizes the variance, every frame of $\tilde{Z}_\gamma$ remains exactly i.i.d. $\mathcal{N}(\mathbf{0}, \mathbf{I})$ across spatial locations and channels for

any $\gamma$, so Proposition 1 carries over to the noise the model actually receives. During training, $\gamma$ is sampled uniformly from $[0, 1]$; at inference it is a user-controlled parameter governing the strength of motion adherence.

# D Evaluation Protocol

**DAVIS (motion transfer).** We estimate forward optical flows on each reference video with RAFT [47] and use them to construct $\tilde{Z}_\gamma$. Motion adherence is measured by comparing RAFT flows of the generated video against those of the reference video (Flow EPE), together with CoTracker [26] mIoU and PSNR.

**MC-Bench (fine-grained and object-level control).** For each sample, we render a motion video from the provided mask and drag trajectory, estimate its flow with RAFT, and restrict the flow to the masked region.

**DL3DV (camera control).** Following TTM [46], we convert each camera trajectory into the corresponding optical flow. For ATE, camera poses are estimated with COLMAP [44] on both the generated and the reference video, and the normalized error is computed between the two trajectories.

**Baseline inputs.** All flow-conditioned methods (GWTF and ours) receive identical input flows, while the remaining baselines are conditioned on the same reference videos.

# E Extended Details of Spatial Gaussianity Preservation

## E.1 Proof of Proposition 1

To prove Proposition 1, we establish its two claims using the unrolled form $\tilde{\mathbf{z}}_n(\hat{x}) = \mathbf{P}_n(\hat{x})\,\mathbf{z}_0(\hat{x})$, where $\mathbf{P}_n(\hat{x}) = \mathbf{Q}_n(\hat{x}) \cdots \mathbf{Q}_1(\hat{x})$.

*Proof.*

**Marginal Gaussianity.** Each $\mathbf{Q}_j(\hat{x}) = \prod_{k=1}^{K} \mathbf{R}(\mathbf{v}_k^{(j)}(\hat{x}))$ is a product of Householder reflections, each of which is orthogonal by construction. Since a finite product of orthogonal matrices is orthogonal, $\mathbf{Q}_j(\hat{x})^\top \mathbf{Q}_j(\hat{x}) = \mathbf{I}_C$ for all $j$, and hence $\mathbf{P}_n(\hat{x})^\top \mathbf{P}_n(\hat{x}) = \mathbf{I}_C$.

For any orthogonal $\mathbf{P}$ and $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_C)$:

$$\mathbf{P}\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{P}\mathbf{I}_C\mathbf{P}^\top) = \mathcal{N}(\mathbf{0}, \mathbf{P}\mathbf{P}^\top) = \mathcal{N}(\mathbf{0}, \mathbf{I}_C). \tag{10}$$

Applying this to $\tilde{\mathbf{z}}_n(\hat{x}) = \mathbf{P}_n(\hat{x})\,\mathbf{z}_0(\hat{x})$ yields $\tilde{\mathbf{z}}_n(\hat{x}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_C)$ for all $n$ and $\hat{x}$. For odd $K$, $\det \mathbf{Q}_j(\hat{x}) = -1$; this does not affect Proposition 1, since any orthogonal matrix preserves $\mathcal{N}(\mathbf{0}, \mathbf{I}_C)$, and the $K = 1$ results in Table 5 confirm that odd $K$ works in practice.

**Spatial independence.** The transformation at each location $\hat{x}$ acts solely on $\mathbf{z}_0(\hat{x})$ without coupling to neighboring locations. For any two distinct locations $\hat{x} \neq \hat{x}'$:

$$\mathrm{Cov}[\tilde{\mathbf{z}}_n(\hat{x}),\, \tilde{\mathbf{z}}_n(\hat{x}')] = \mathbf{P}_n(\hat{x})\ \mathrm{Cov}[\mathbf{z}_0(\hat{x}),\, \mathbf{z}_0(\hat{x}')]\,\mathbf{P}_n(\hat{x}')^\top = \mathbf{P}_n(\hat{x})\,\mathbf{0}\,\mathbf{P}_n(\hat{x}')^\top = \mathbf{0}, \tag{11}$$

where $\mathrm{Cov}[\mathbf{z}_0(\hat{x}), \mathbf{z}_0(\hat{x}')] = \mathbf{0}$ follows from the spatial independence of the initial noise. Since the vectors $\tilde{\mathbf{z}}_n(\hat{x})$ are jointly Gaussian (each is a linear function of the initial noise) with zero cross-location covariance, they are mutually independent. □

## E.2 Cross-Frame Covariance Derivation

We derive the full covariance structure of the motion-embedded noise. For frames $n$, $m$ and latent locations $\hat{x}$, $\hat{x}'$:

$$\begin{aligned}\mathrm{Cov}[\tilde{\mathbf{z}}_n(\hat{x}),\, \tilde{\mathbf{z}}_m(\hat{x}')] &= \mathbb{E}\big[\mathbf{P}_n(\hat{x})\,\mathbf{z}_0(\hat{x})\,\mathbf{z}_0(\hat{x}')^\top\,\mathbf{P}_m(\hat{x}')^\top\big] \\ &= \mathbf{P}_n(\hat{x})\,\mathbb{E}\big[\mathbf{z}_0(\hat{x})\,\mathbf{z}_0(\hat{x}')^\top\big]\,\mathbf{P}_m(\hat{x}')^\top = \delta_{\hat{x}\hat{x}'}\,\mathbf{P}_n(\hat{x})\,\mathbf{P}_m(\hat{x}')^\top,\end{aligned} \tag{12}$$

Table 7: Comparison of Gaussianity preservation and computational efficiency between methods that produce temporally correlated noise (1024×1024 resolution, $C = 16$) driven by optical flow; statistics for our method are computed on $\tilde{Z}$ (i.e., $\gamma = 0$), and all GPU times are measured on an NVIDIA RTX PRO 6000 Blackwell GPU.

| Method | Moran's I ($\approx 0$) | Moran's I p-value↑ | K-S Test (index) ↓ | K-S Test p-value ↑ | Std. ($\approx 1.0$) | Skewness ($\approx 0$) | Ex. Kurtosis ($\approx 0$) | GPU Time ms/Latent↓ |
|---|---|---|---|---|---|---|---|---|
| *Warping + Interpolation Operation* | | | | | | | | |
| Nearest | 0.1150 | 0.012 | 0.0107 | 0.115 | 1.0015 | 0.0016 | -0.0116 | - |
| Bilinear | 0.6218 | 0.008 | 0.3318 | 0.000 | 0.2179 | 0.0720 | 0.4753 | - |
| Bicubic | -0.0671 | 0.020 | 0.0670 | 0.000 | 1.0489 | 0.0075 | 0.2194 | - |
| PYoCo [13] | 0.0002 | 0.478 | 0.0033 | 0.512 | 1.000 | 0.0005 | 0.0000 | - |
| *Optical Flow Driven Noise Warping Methods* | | | | | | | | |
| CaV [9] | 0.6170 | 0.004 | 0.3379 | 0.000 | 0.2060 | 0.0590 | 4.065 | - |
| HIWYN [8]$_{L=3}$ | 0.0001 | 0.502 | 0.0037 | 0.422 | 0.9957 | 0.0007 | -0.0001 | 38.00 |
| GWTF [7] | 0.0002 | 0.507 | 0.0035 | 0.494 | 0.9997 | -0.0006 | 0.0037 | 7.63 |
| InfRes [10] | 0.0003 | 0.512 | 0.0034 | 0.496 | 1.0001 | 0.0004 | 0.0011 | 23.94 |
| *Ours (HME)* | | | | | | | | |
| $K = 2$ | 0.0003 | 0.482 | 0.0032 | 0.535 | 0.9998 | -0.0006 | 0.0014 | **2.80** |
| $K = 4$ | 0.0003 | 0.475 | 0.0032 | 0.531 | 0.9999 | -0.0005 | 0.0014 | <u>5.18</u> |
| $K = 6$ | 0.0003 | 0.466 | 0.0033 | 0.527 | 1.0001 | -0.0007 | 0.0015 | 7.59 |
| $K = 8$ | 0.0003 | 0.493 | 0.0033 | 0.528 | 0.9998 | -0.0004 | 0.0011 | 10.01 |
| $K = 12$ | 0.0002 | 0.534 | 0.0032 | 0.546 | 1.0001 | -0.0005 | 0.0013 | 14.78 |

where $\delta_{\hat{x}\hat{x}'}$ is the Kronecker delta, following from the zero mean and spatial independence of $\mathbf{z}_0$. Noise at distinct locations is therefore uncorrelated across all frames, and at the same location the covariance reduces to Eq. (7); for $n > m$,

$$\mathrm{Cov}[\tilde{\mathbf{z}}_n(\hat{x}),\, \tilde{\mathbf{z}}_m(\hat{x})] = \mathbf{P}_n(\hat{x})\, \mathbf{P}_m(\hat{x})^\top = \mathbf{Q}_n(\hat{x}) \cdots \mathbf{Q}_{m+1}(\hat{x}), \tag{13}$$

an orthogonal matrix determined only by the flows between frames $m$ and $n$ at $\hat{x}$. Its Frobenius norm is $\sqrt{C}$ for all $n$, $m$ and $\hat{x}$, so the strength of the cross-frame dependence is reflected not in its norm but in its alignment with the identity: the average channel-wise correlation $\frac{1}{C}\,\mathrm{tr}\big(\mathbf{P}_n(\hat{x})\mathbf{P}_m(\hat{x})^\top\big) = \frac{1}{C}\mathbb{E}\big[\tilde{\mathbf{z}}_n(\hat{x})^\top \tilde{\mathbf{z}}_m(\hat{x})\big] \in [-1, 1]$ equals 1 if and only if $\mathbf{Q}_n(\hat{x}) \cdots \mathbf{Q}_{m+1}(\hat{x}) = \mathbf{I}_C$.

### E.3 Experimental Results

**Metrics and Baselines.** To evaluate the Gaussianity preservation ability of our method, we follow InfRes [10] and GWTF [7] to present Moran's I and Kolmogorov-Smirnov (K-S) test results. Moran's I measures spatial autocorrelation of warped noise and the K-S test assesses the distributional fidelity of the warped noise against a standard normal distribution. We further report noise Std., skewness and excess kurtosis to characterize higher-order distributional properties such as the symmetry and shape of the resulting distribution compared to normal distribution. We also evaluate the noise processing efficiency of all methods using GPU time. Baselines mainly include optical flow driven warping methods (CaV, HIWYN [8], GWTF [7], InfRes [10]) designed for Gaussianity preservation. These benchmarks operate on the noise tensor alone and run at $1024 \times 1024$, $C = 16$, on an NVIDIA RTX PRO 6000 Blackwell GPU; this setting only measures noise properties, computational overhead and memory, and is independent of the $480 \times 720$ video generation experiments in Section 4.

**Performance Analysis.** As shown in Table 7, **HME** is successful in generating spatially Gaussian noise with Moran's I index values close to 0, with corresponding p-values ranging from 0.466 to 0.534, indicating no significant spatial autocorrelation, an important property of i.i.d. noise. Small K-S index values and p-values above 0.05 indicate the spatial distribution of resulting noise matches a standard normal distribution. **HME** obtains a K-S statistic of $\sim 0.0032$ and p-value of $\sim 0.53$ on average, demonstrating the distributional fit to a standard normal distribution. Furthermore, our method achieves near 1 Std., negligible Skewness and Excess Kurtosis, suggesting substantial normality. In contrast, all interpolation-based operations introduce significant spatial autocorrelation (Moran's I p-values of 0.008–0.020); bilinear and bicubic interpolation additionally fail the K-S test, while nearest-neighbor sampling preserves the marginal distribution (K-S p-value $= 0.115$) but not the spatial independence of the noise.

Meanwhile, dedicated flow-based methods also recover near-zero Moran's I, acceptable K-S p-values, and negligible skewness and excess kurtosis, but at runtimes of 7.63–38.00 ms/latent. Our method reduces the GPU runtime to $\mathbf{2.80}$ ms/latent at $K = 2$, a $2.7\times$ speedup over GWTF and $13.6\times$ over HIWYN. All our variants ($K = 2$ to 12) satisfy the Gaussianity criteria with runtime scaling gracefully from 2.80 to 14.78 ms/latent.

# F Extra Qualitative Comparisons

Video versions of the comparisons in this paper, along with additional results, are provided in the Supplementary Material; differences between methods are easier to spot in the videos.

## F.1 Motion Transfer

For clarity, we show qualitative motion-transfer comparisons with GWTF, DisMo and MotionPro in Fig. 5; quantitative comparisons with all baselines, including TTM, are given in Table 1.

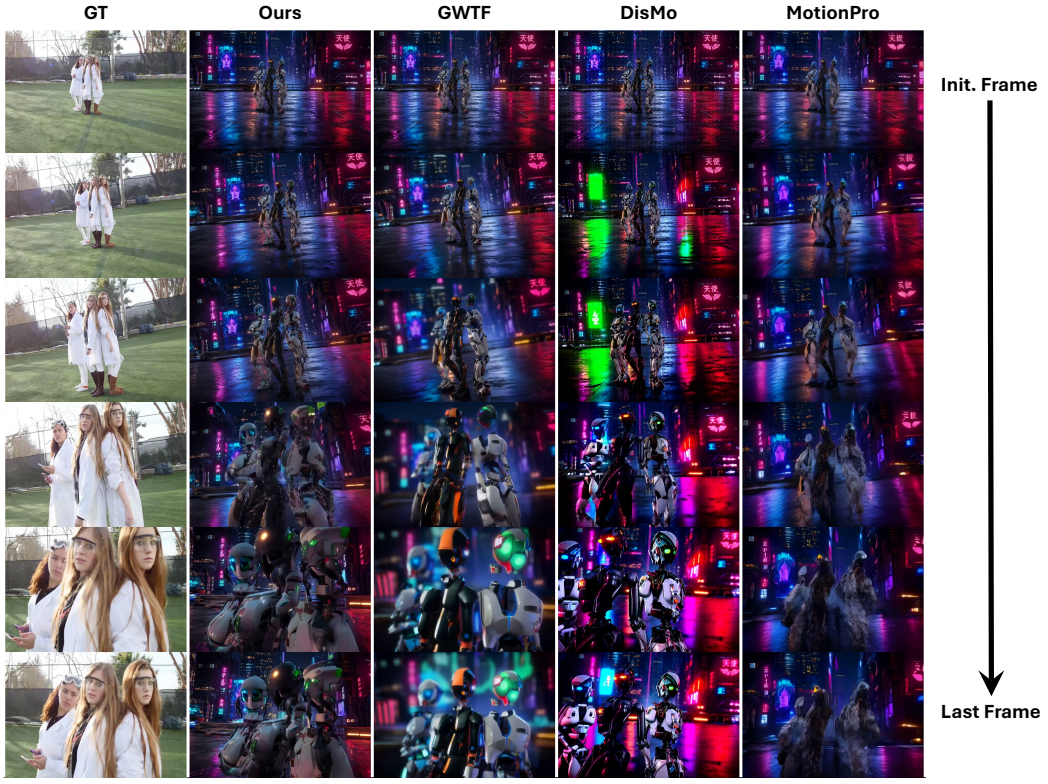



Figure 5: Motion transfer example: we demonstrate strong motion adherence to the reference video while maintaining high generation quality and fidelity. *Prompt used: "Cyberpunk rooftop at night, wet reflective ground, neon city skyline, three humanoid robots standing slightly left of center in a triangular group pose, aligned like three girls in an open field. White-blue robot, black-orange robot, white-red-green robot. Wide composition, large empty foreground and right-side space, glowing Japanese signs, holographic billboards, photorealistic, cinematic, ultra detailed."*

## F.2 Qualitative Comparisons From MC-Bench

Following the experiment described in Section 4.2, we present additional results beyond those shown in Fig. 3, further illustrating our method's performance against leading approaches on the MC-Bench dataset.

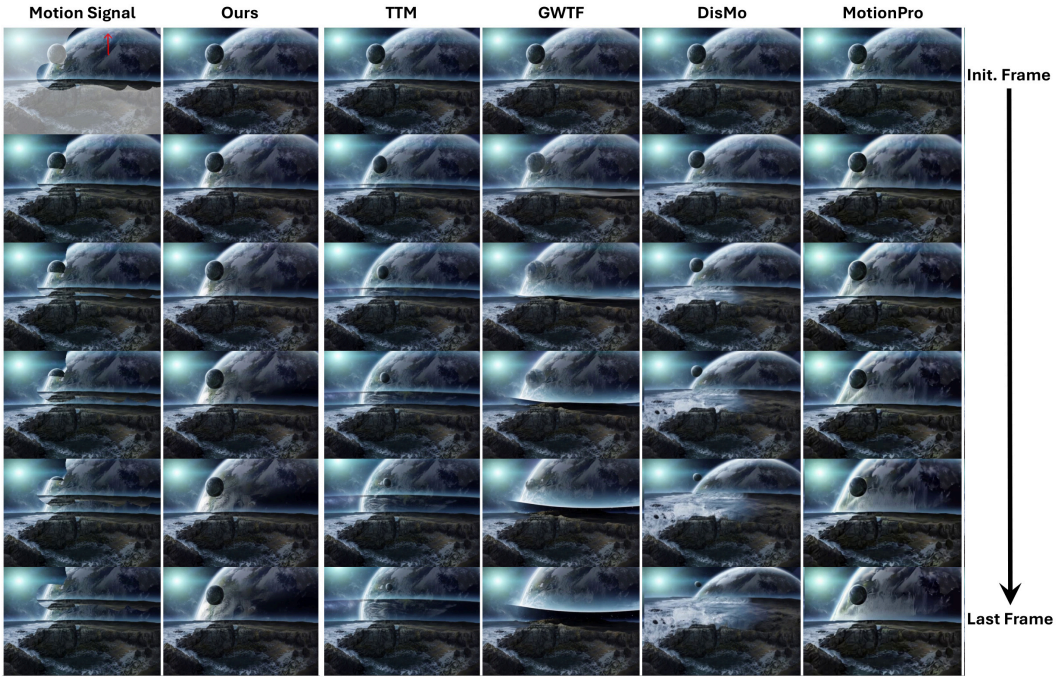



Figure 6: Example from MC-Bench: we highlight the disentanglement between the moving subject (masked planet) and the background in our method.

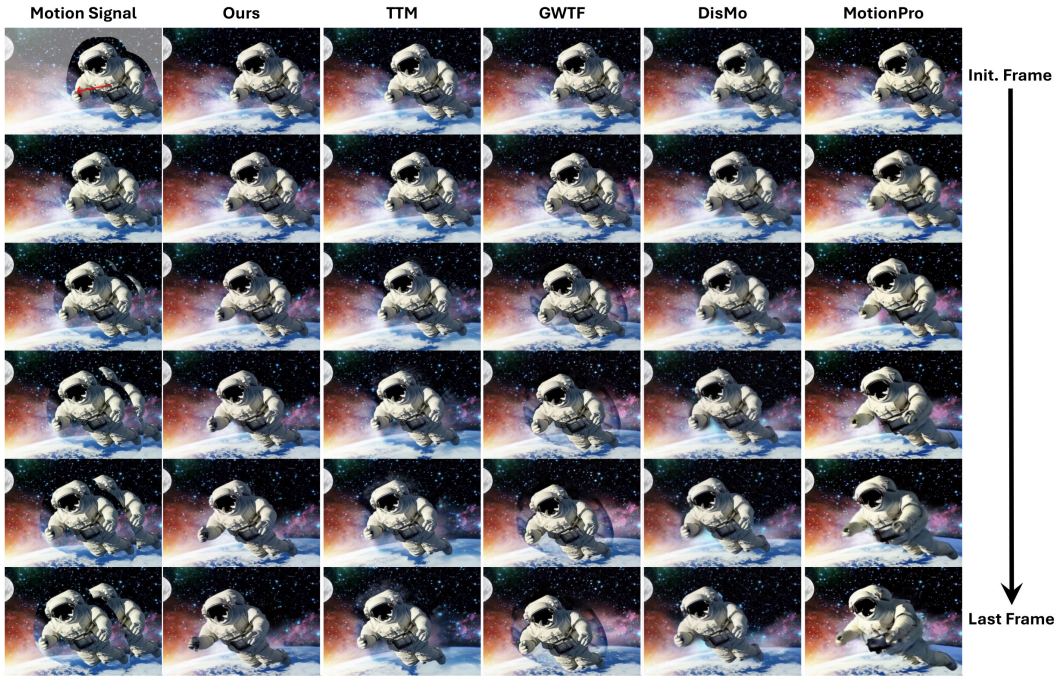



Figure 7: Example from MC-Bench: we highlight the disentanglement between the moving astronaut and the background in our method.

### F.3 Qualitative Comparisons From DAVIS

Following the experiment described in Section 4.2, we present additional results beyond those shown in Fig. 3, further illustrating our method's performance against leading approaches on the DAVIS dataset.

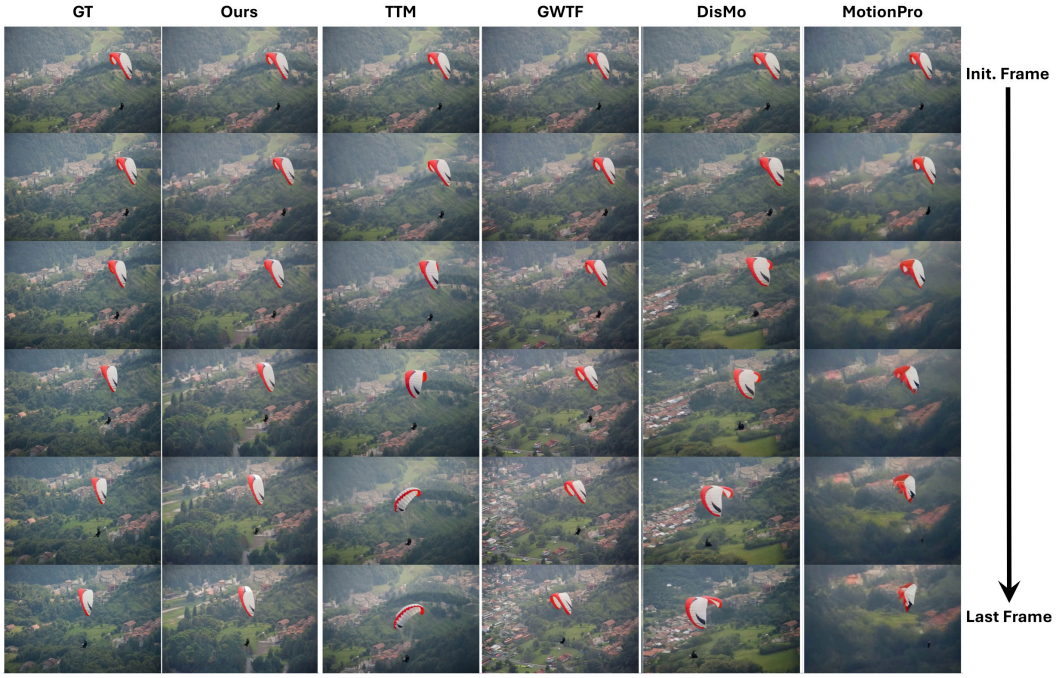



Figure 8: Example from DAVIS: our method shows better motion adherence.




Figure 9: Example from DAVIS: our method shows better motion adherence.

### F.4 Qualitative Comparisons From DL3DV

Following the experiment described in Section 4.2, we present additional camera-control results beyond those shown in Fig. 4.

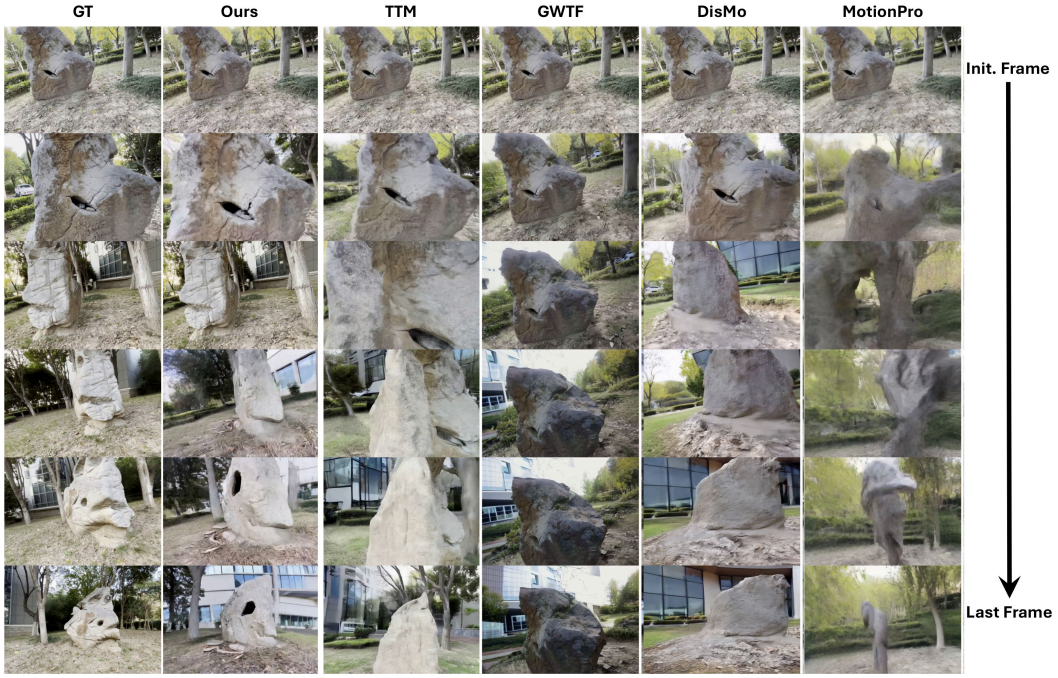



Figure 10: Example from DL3DV: our method shows better camera motion adherence under optical flow control.

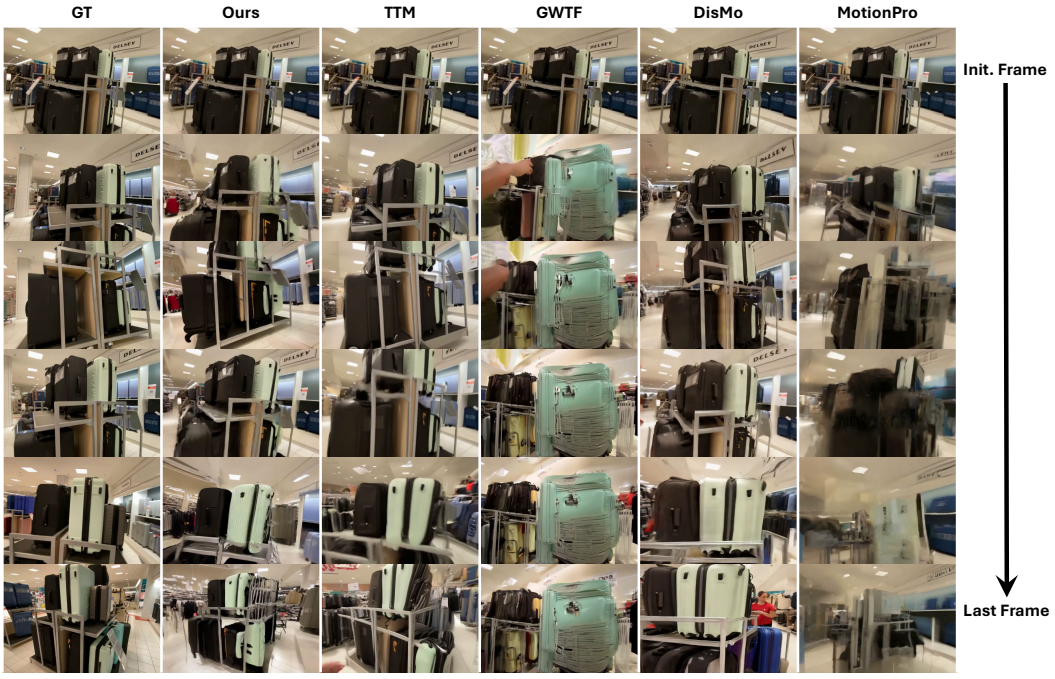



Figure 11: Example from DL3DV: our method shows better camera motion adherence under optical flow control.

## G Qualitative Results for Different Noise Degradation Levels

We qualitatively ablate the noise degradation level $\gamma$, as analyzed in Section 4.3.

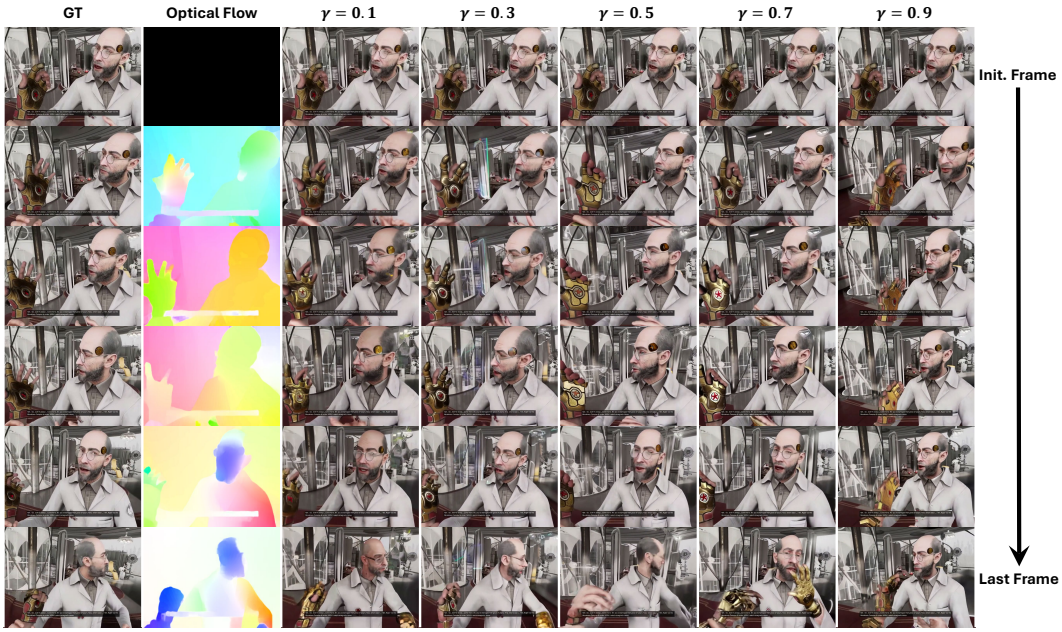



Figure 12: The effect of noise degradation level $\gamma$ on generated videos. As degradation decreases, the video adheres more to the input flow.

## 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: **We have clearly state our contribution, scope and technical novelty of our method in the abstract and introduction. The claims we made do match the theoretical and experimental results, and are generalizable to most of the relevant task settings.**

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We have discussed the limitations and assumptions of our methods in the paper, and we also analyzed the computation efficiency of the proposed algorithms.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: We have provided mathematical proofs and theoretical explanations for our method in the paper, and we provide experimental results to support our theoretical assumptions.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: We have clearly described our algorithm and data we used, and we will open-source the code for reproducibility.

Guidelines:

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

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

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

   Answer: [Yes]

   Justification: We will release full code of our method and evaluation scripts to reproduce all experimental results.

   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 experimental settings and details are presented.

   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: [N/A]

   Justification: We do not report statistical significance with error bars in the paper.

   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: Details of experiments are presented in the Experiments section.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We reviewed and followed the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We acknowledge the potential misuse of video generation models, including the creation of deepfakes or misleading media.

    Guidelines:

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

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All data and models used in this paper are under public license.

    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: [N/A]

    Justification: The paper does not release new assets

    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: [N/A]

    Justification: No human subjects are involved in the paper.

    Guidelines:

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

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

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

    Answer: [N/A]

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

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs.

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.