MoRight: Motion Control Done Right

Shaowei Liu[1,2] **Xuanchi Ren**[1] **Tianchang Shen**[1] **Huan Ling**[1] **Saurabh Gupta**[2]
Shenlong Wang[2] **Sanja Fidler**[1] **Jun Gao**[1]
[1]NVIDIA [2]University of Illinois Urbana-Champaign

https://research.nvidia.com/labs/sil/projects/moright

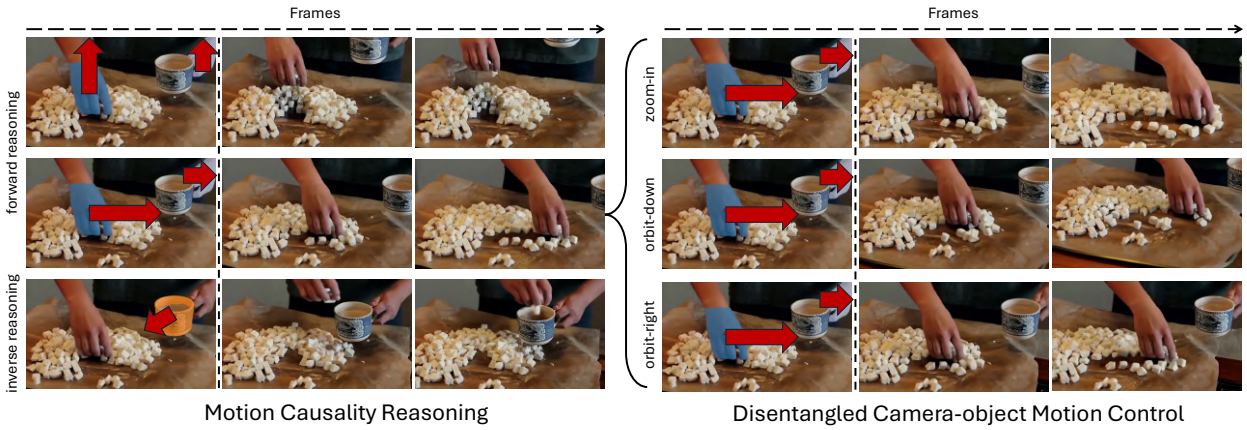

Figure 1: Given a single input image, our method enables controllable interactive motion generation with motion causality reasoning. *Left:* Users can provide active motion to drive scene dynamics (*forward reasoning*) or specify desired passive outcomes and recover plausible driving actions (*inverse reasoning*). *Right:* The model further enables disentangled control of object motion and camera viewpoint, allowing users to explore the scene with custom viewpoints and motions.

Abstract

Generating motion-controlled videos—where user-specified actions drive physically plausible scene dynamics under freely chosen viewpoints—demands two capabilities: (1) *disentangled motion control*, allowing users to separately control the object motion and adjust camera viewpoint; and (2) *motion causality*, ensuring that user-driven actions trigger coherent reactions from other objects rather than merely displacing pixels. Existing methods fall short on both fronts: they entangle camera and object motion into a single tracking signal and treat motion as kinematic displacement without modeling casual relationships between object motion. We introduce MoRight, a unified framework that addresses both limitations through disentangled motion modeling. Object motion is specified in a canonical static-view and transferred to an arbitrary target camera viewpoint via temporal cross-view attention, enabling disentangled camera and object control. We further decompose motion into *active* (user-driven) and *passive* (consequence) components, training the model to learn motion causality from data. At inference, users can either supply active motion and MoRight predict consequences (*forward reasoning*), or specify desired passive outcomes and MoRight recover plausible driving actions (*inverse reasoning*), all while freely adjusting the camera viewpoint. Experiments on three benchmarks demonstrate state-of-the-art performance in generation quality, motion controllability, and interaction awareness.

Keywords: Video Generation; Disentangled Motion Control; Causal Motion Reasoning

1. Introduction

Humans interact with the physical world as active agents: we move our viewpoint, manipulate objects, and reason about how actions lead to consequences. Yet existing video generation models lack this unified

capability [7, 11, 25, 47, 61]. Bridging this gap is essential for applications that demand interactive visual reasoning, from embodied AI agents [6, 52, 58] that must anticipate action outcomes, to world models [8, 24, 26, 26, 60] that simulate physical interactions for robotic planning, to immersive content creation where users freely navigate and manipulate scenes. A desirable video generation system should therefore offer joint controllability over both camera and object motion—generating visually coherent frames under arbitrary viewpoint changes while producing causally consistent scene dynamics driven by user-specified actions.

Existing controllable video generation methods [9, 13, 19, 48, 49, 55] work as *renderers*: given displacements for *all* pixels, they generate a visually realistic video that adheres to the displacements. These approaches have two key limitations in practice. First, they entangle camera and object motion, making joint control ambiguous because viewpoint changes alter pixel trajectories. Second, they interpret user-specified motion as simple kinematic displacement and largely ignore the consequences of the given trajectories. Models, therefore, focus on following trajectories rather than reasoning about causal relationships between objects. In reality, actions cause consequences—pushing a cup may cause it to slide and collide with other objects, while lifting a teapot may cause water to pour. Specifying the effects of all the objects' motion through motion prompts is often impractical. Without modeling these action–consequence relationships, motion-controlled generation cannot capture the causal structure of real-world interactions.

To overcome these challenges, we introduce MoRight, a unified framework for video generation with disentangled camera–object motion control and motion causality reasoning as shown in Fig. 1. Given a reference image, user-specified motion trajectories, and target viewpoints, MoRight generates videos where objects follow the desired motion and the scene is rendered from the specified cameras. For disentangled camera-object motion control, our key insight is that specifying the motion of objects under camera changes is inherently difficult. We therefore introduce a **dual-stream motion formulation**. The first branch models and generates object motion in the source image plane under a canonical static viewpoint, allowing users to easily specify dynamic trajectories. The second branch represents the target camera motion and transfers object dynamics from the canonical branch via temporal cross-view attention. This *cross-view motion transfer* enables independent control of camera and object motion while maintaining coherent scene dynamics. For motion causality reasoning, we achieve this by decomposing object motion into two categories during training: *active motion*, representing user-driven actions, and *passive motion*, representing their causal outcomes. By conditioning on either active or passive motion signals, the video model learns to generate all the scene dynamics, capturing action–consequence relationships within the scene. This yields two complementary capabilities: forward reasoning of scene evolution from user actions, and inverse reasoning of plausible actions that produce a desired outcome.

We evaluate MoRight on three benchmarks covering diverse interaction scenarios. Results show that MoRight outperforms existing methods in generation quality, motion controllability, and interaction awareness, validating the effectiveness of disentangled motion control and causal motion reasoning.

In summary, our contributions are threefold. (1) We propose a disentangled framework for camera and object motion control, enabling users to draw motion trajectories directly in the image plane while freely adjusting viewpoints to generate coherent videos. (2) We empower video generation models with motion reasoning capability by modeling action–consequence relationships, allowing user-driven motions to meaningfully interact with the environment and produce consistent scene dynamics. (3) We demonstrate that MoRight supports both forward and inverse reasoning: given active motion inputs, it predicts future scene evolution; given desired passive outcomes, it recovers plausible actions that achieve them. Together, these advances establish a unified framework for controllable and reasoning-aware video generation.

2. Related Work

2.1. Motion-Controlled Video Generation

Controllable video generation has been studied across a spectrum of motion granularity, from coarse region-level signals such as bounding boxes [37, 49, 57] and optical flow fields [28, 31, 42, 43, 65], to dense per-pixel trajectories [13, 19], to sparse keypoint tracks [30, 55, 56, 62] that balance precision with ease of specification. Despite steady progress, trajectory-based methods share two fundamental limitations. First, because trajectories are defined in pixel space, they inevitably entangle object and camera motion: any viewpoint change alters all trajectories, making joint control ill-posed without explicit foreground–background decomposition [63]. Second, producing physically plausible motion typically demands carefully crafted trajectories from dedicated motion-generation models or laborious manual annotation. MoRight overcomes both issues by disentangling camera and object motion by design, and by accepting lightweight inputs—simple strokes or sparse tracklets—that the model completes into coherent, interaction-aware dynamics.

2.2. Camera–Object Motion Disentanglement

Separating camera motion from scene dynamics is a long-standing challenge in controllable generation [12, 19, 21, 42, 65]. Existing methods [45, 53, 54] attempt to decouple the two by treating them as independent control signals, yet they typically rely on privileged information such as per-frame depth [12, 21], 3D object trajectories [15, 22, 29, 29], or foreground–background segmentation masks, and pre-warp all signals to their anticipated future locations. These assumptions implicitly require the full video sequence or 3D motion to be known in advance, severely limiting applicability in image-to-video settings where only a single reference frame is available. MoRight instead introduces a canonical static-view branch for object dynamics and transfers them to arbitrary target viewpoints via cross-view attention, eliminating the need for explicit 3D supervision or pre-computed scene decomposition.

2.3. Interaction and Causal Reasoning in Video Generation

A complementary line of work seeks to move beyond kinematic animation toward causally grounded video generation. One family of methods incorporates external physics engines [10, 32, 35] or conditions on explicit action representations such as force vectors [20] to model specific phenomena (e.g., fluid flow, rigid-body collisions). While effective in constrained domains, these approaches are tailored to particular interaction types and require a simulation module in the loop, limiting their generality. Another family delegates causal reasoning to vision–language or large language models [38, 59], which first predict outcomes in text and then transfer them to the video generator through intermediate representations such as flow fields, edge maps, or depth [36, 59]. This two-stage pipeline is prone to error propagation: imprecise textual predictions are further degraded during cross-modal conversion, yielding spatially inaccurate dynamics. MoRight sidesteps both limitations by learning latent action–response structure directly from video data. Decomposing motion into active (user-driven) and passive (consequence) components allows the model to perform cause–effect reasoning at the pixel level within a single forward pass, enabling both forward prediction of scene consequences and inverse inference of plausible underlying actions.

3. Approach

Given a single image I, we aim to generate a video of T frames $\mathbf{x} \in \mathbb{R}^{T\times H\times W\times 3}$ that follows the user-defined motion of an object represented as pixel trajectories $\mathcal{T}$, and the specified camera motion sequence $\{C_i\}_{i=1}^{T}$. The generated video should be able to model the causality of motion and produce a coherent dynamics within the scene. To achieve this, we first present our approach for disentangled camera and object motion control

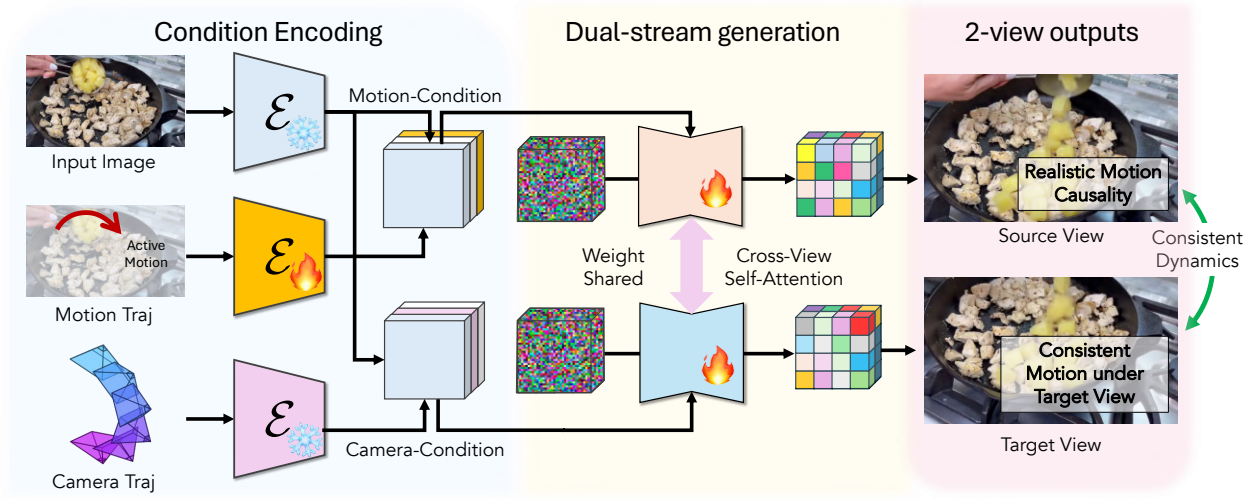

Figure 2: **Model architecture.** Our model adopts a dual-stream architecture with shared weights to disentangle object motion from camera motion. The canonical stream encodes motion trajectories using a track encoder and learns motion in a fixed canonical view. The target stream encodes camera pose signals through a camera encoder. The resulting motion and camera conditions are injected into every attention block of the network. Cross-view self-attention connects the two streams, transferring motion learned in the canonical view to the target view and enabling disentangled camera–object motion generation.

in Sec. 3.1 and Fig. 2, and then introduce motion causality modeling in Sec. 3.2. Sec. 3.3 describes our training data curation pipeline, and training details along with the inference pipeline are provided in Sec. 3.4.

3.1. Disentangled Camera-Object Motion Control

Most existing approaches adopt pixel-wise trajectories as motion control signals. However, such representations inherently entangle object motion with viewpoint changes. The video model must implicitly reason how an object moves, and the camera changes, without explicit geometric cues, when conditioned on these signals. Our key insight is that object motion is intrinsically unambiguous when expressed in a canonical camera. Building on this observation, we decouple the motion signal from the viewpoint transformation through a dual-stream generation framework. The first stream synthesizes a canonical video in a static camera, where object motion can be directly and faithfully controlled. The second stream generates the target video with both camera and object motion. The two streams can interact with each other through self-attention as shown in Fig. 2. By jointly denoising both streams, the model learns to transfer motion cues from canonical space to arbitrary camera poses, where the canonical stream serves as an anchor for motion control. This design resolves motion–camera entanglement at generation time while naturally supporting heterogeneous supervision—including motion-only, camera-only, and fully coupled data—as detailed in Sec. 3.3.

Preliminaries. We build upon a DiT-based latent video diffusion models [14, 47]. A pretrained VAE encoder first encodes the video into a latent space $\mathbf{z}_0 = \mathcal{E}(\mathbf{x}) \in \mathbb{R}^{\hat{T}\times\hat{H}\times\hat{W}\times d}$. The diffusion model is trained in this latent space via flow matching [34]. Specifically, we first sample a noise from a Gaussian distribution: $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, and form $\mathbf{z}_t = (1-t)\,\mathbf{z}_0 + t\,\boldsymbol{\epsilon}$ for $t \in [0, 1]$. The DiT $\mathcal{G}_\theta$ is trained to regress the velocity:

$$\mathcal{L} = \mathbb{E}_{\mathbf{z}_0,\, t,\, \boldsymbol{\epsilon}} \left[\left\| \mathcal{G}_\theta(\mathbf{z}_t, t, \mathbf{c}) - (\boldsymbol{\epsilon} - \mathbf{z}_0) \right\|^2 \right], \tag{1}$$

where $\mathbf{c}$ denotes conditioning signals (*e.g.*, text). At inference, an ODE solver integrates the learned velocity from a noise to a clean latent $\hat{\mathbf{z}_0}$, from which a decoder reconstructs the original video $\hat{\mathbf{x}} = \mathcal{D}(\hat{\mathbf{z}_0})$.

Dual-stream generation. The user first provide the object motion $\{\boldsymbol{\tau}_i^{\text{can}}\}_{i=1}^{T}$ in the canonical frame. The first

stream generates the videos only with object motion, and the second stream generates the videos with both object and camera motion. Concretely, the two streams receive their individual conditions:

$$\mathbf{c}^{\text{can}} = \{I,\ C_1,\ \{\boldsymbol{\tau}_i^{\text{can}}\}\}, \qquad \mathbf{c}^{\text{tar}} = \{I,\ \{C_i\},\ \emptyset\},$$

where C_1 denotes the camera in the first image and is an identity matrix, and $\emptyset$ denotes empty object motion.

In the following, we assume we have the paired training data: one ground truth video $\mathbf{x}^{\text{can}}$ with object motion only, and the corresponding video $\mathbf{x}^{\text{tar}}$ with both object and camera motion. Extensions to other training data are described in Sec. 3.3 and Sec. 3.4. We add independent noise with the same timestep t to each stream and obtain $\mathbf{z}_t^{\text{can}}$ and $\mathbf{z}_t^{\text{tar}}$. We then concatenate two latents along the temporal dimension and jointly denoise them. We slightly modify the positional embedding to indicate the difference between the two streams, with details in the supplement. In this way, we reuse the same DiT weights for two streams, and the only difference is their input and conditioning. The two streams naturally exchange information in the self-attention layers of each transformer block. During inference, the two streams are jointly denoised, and we provide the output from the target stream $\mathbf{x} = \mathcal{D}(\hat{\mathbf{z}}_0^{\text{tar}})$ to the user, and the canonical stream serves as an "virtual" anchor.

Condition injection and motion transfer. We inject camera and motion conditions into the latents of DiT at every transformer block. Specifically:

Camera encoding. We follow Gen3C [64] and warp the first image I using the corresponding camera pose and estimated depth [33]. We then encode the warped frames via encoder $\mathcal{E}$ from VAE and obtain the latent $\mathbf{z}^{\text{cam}} \in \mathbb{R}^{\hat{T}\times\hat{H}\times\hat{W}\times d}$. For the canonical stream, we use the identity matrix for warping.

Motion encoding. Following [19], we build a per-pixel trajectory map where pixels along one trajectory share the same temporal-correspondence embedding. We then encode it via a lightweight encoder to obtain $\mathbf{e}^{\text{trk}} \in \mathbb{R}^{\hat{T}\times\hat{H}\times\hat{W}\times d}$. For the target stream, we simply set $\mathbf{e}^{\text{trk}} = \mathbf{0}$ since the condition is empty.

Condition injection. The camera and motion encodings are fused via learned linear projections and added into the latent feature at each transformer block. Let $\mathbf{f}$ denote the feature of one block:

$$\mathbf{f}^i \leftarrow \mathbf{f}^i + W_{\text{cam}}\,\mathbf{z}^{i,\text{cam}} + W_{\text{trk}}\,\mathbf{e}^{i,\text{trk}}, \quad i \in \{\text{can},\ \text{tar}\}. \tag{2}$$

The features from the two streams are then concatenated and passed through the self-attention layer:

$$\left[\,\mathbf{f}^{\text{can}};\ \mathbf{f}^{\text{tar}}\,\right] := \text{SelfAttn}\big(\left[\,\mathbf{f}^{\text{can}};\ \mathbf{f}^{\text{tar}}\,\right]\big), \tag{3}$$

allowing target-view tokens to attend to motion-conditioned canonical tokens and vice versa, implicitly exchanging the motion information in latent space. This inject-then-synchronize repeats at every block, progressively transferring motion across views, as shown in Fig. 2.

3.2. Motion Causality Modeling

Disentangling the camera from motion alone is insufficient for realistic interactions: when a hand pushes a cup, the cup must slide; when a ball strikes a stack of blocks, the blocks must scatter. We term this as *motion causality*: the ability to reason plausible consequences from the given actions, and vice versa.

To model this, we decompose the motion tracks of all foreground objects $\mathcal{T} = \{\boldsymbol{\tau}_i\}_{i=1}^{T}$ into two complementary components as shown in Fig. 3: $\boldsymbol{\tau}_i = \boldsymbol{\tau}_i^{\text{act}} \cup \boldsymbol{\tau}_i^{\text{pas}}$, where $\boldsymbol{\tau}_i^{\text{act}}$ captures the *active* (causal) motion—the intentional action applied to the scene (*e.g.*, a hand pushing)—and $\boldsymbol{\tau}_i^{\text{pas}}$ captures the *passive*

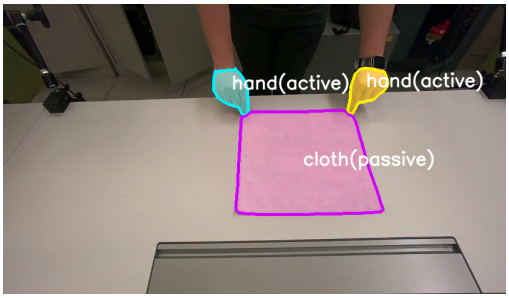

Figure 3: Active vs. passive motion. The *active* object (hand) initiates the action, while the *passive* object (cloth) responds.

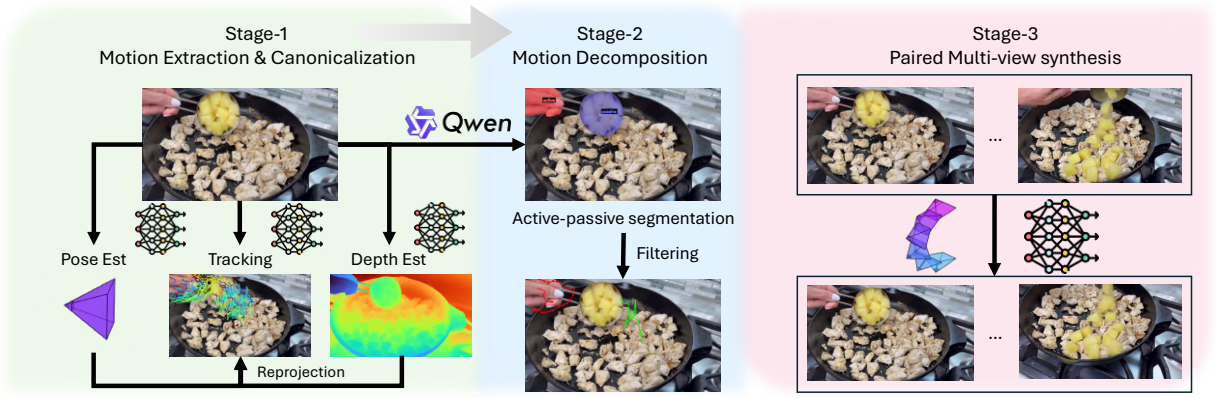

Figure 4: **Data curation pipeline.** Foundation models [22, 27, 39] extract depth, camera poses, and tracks from raw videos. A VLM [3] segments tracks into active/passive regions. We further optionally use a video-to-video model [18] to generate paired videos with the same object motion but different camera motions.

(consequential) motion—the reaction from other objects (*e.g.*, the pushed object sliding). Reasoning such causality is critical in applications such as embodied AI.

The central mechanism to enable the causality modeling is through *motion dropout*. Specifically, during training, we randomly dropout one motion component from the input, and supervise the model on the full video containing both active and passive motion:

$$\tilde{\boldsymbol{\tau}}_i := \begin{cases} \boldsymbol{\tau}_i^{\text{act}}, & \xi < p, \\ \boldsymbol{\tau}_i^{\text{pas}}, & \text{otherwise}, \end{cases} \tag{4}$$

where ξ is sampled from a uniform distribution $\mathcal{U}(0,1)$, p is the dropout probability and $\tilde{\boldsymbol{\tau}}_i$ is used as tracking condition to video model.

In our model training, we do not distinguish active motion or passive motion when feeding them into the model, and only rely on the model's capability to reason the dropped component to generate plausible videos. Over-training, this asymmetric supervision encourages the video models to internalize the causal relationship between actions and their consequences, rather than simply replaying the provided trajectories.

At inference, this learned causality enables two complementary applications: *forward reasoning* (action $\rightarrow$ reaction), where users specify an action and the model generates the resulting consequences; and *inverse reasoning* (reaction $\rightarrow$ action), where users prescribe a desired outcome and the model synthesizes a plausible action that drives it. We demonstrate these capabilities in Sec. 4.4.

3.3. Training Data Curation

Curating data to train our dual-stream model is challenging since most real-world videos are single-view and always entangle camera and object motion, while our model requires paired videos depicting the same dynamics under different viewpoints. We first describe our data annotation pipeline to extract the pixel trajectories, camera poses, and active/passive motion, and provide details of our data curation for training afterwards. The overview of pipeline is shown in Fig. 4.

Motion extraction and canonicalization. Given a video $\mathbf{x}$, we estimate per-frame depth maps $\{D_i\}$, camera poses $\{C_i\}$, and intrinsics K using ViPE [27], and extract dense pixel trajectories $\mathcal{T} = \{\boldsymbol{\tau}_i\}_{i=1}^{T}$ with AllTracker [22]. Each trajectory is unprojected to 3D and reprojected into the first frame:

$$\boldsymbol{\tau}_i^{\text{can}} = \pi\big(K,\, C_0\, C_u^{-1}\, \pi^{-1}(K,\, \boldsymbol{\tau}_i,\, D_i)\big), \tag{5}$$

where π^{-1} lifts 2D points to 3D using depth and intrinsics, and π projects onto the image plane of I_0. We assume constant intrinsics across the video.

Active and passive motion decomposition. For the given video, we prompt a vision-language model (Qwen3 [3]) to identify the active and passive objects, then segment the first frame with SAM2 [39], yielding masks M^{act} and M^{pas}, for active and passive objects, respectively. Trajectories are assigned to each component by mask membership, producing $\boldsymbol{\tau}^{\text{can,act}}$ and $\boldsymbol{\tau}^{\text{can,pas}}$ for the motion dropout training in Eq. (4). We also generate per-video captions describing each motion component and only provide the caption of one component during training to prevent information leakage.

Synthetic data generation for paired two-view videos. We leverage a synthetic data generation pipeline to generate paired two-view videos for training. Specifically, we first curate the videos whose cameras are static by checking the displacement of camera poses estimated from ViPE [27]. We then synthesize corresponding moving-camera videos using a camera-control video-to-video model [17], providing supervision for the second stream. To increase camera diversity, we further augment the data with basic camera operations (*e.g.*, orbit, pan, zoom) as well as dynamic camera trajectories extracted from real videos.

Single-view real-world data for mixed-training. The generated paired videos inevitably contain visual artifacts. In our dual-stream mode, the abundant single-view real-world data can be leveraged for mixed-training. First, for the videos with static cameras (object motion only), we duplicate the video and treat it as the target video. In this case, the video model learns to exchange the motion condition from the first stream (with motion condition) into the target stream (without motion condition). Second, for the videos that exhibit both camera and object motion, we feed the condition into the video model as described in Sec. 3.1 and only supervise the second stream, leaving the loss at the first stream being zero. These two mixed-training strategies expose video models to real-world data with diverse camera and object motion, increasing the robustness and generalizability to various camera and motion configurations, while mitigating artifacts from synthetic data.

Rendered Graphics Data. We further incorporate synthetic data from SyncCamMaster [2] to expose our model to more camera diversity.

3.4. Training and Inference

We train the DiT using the flow matching loss from Eq. (1), and apply two complementary dropout strategies to encourage the model learn the motion causality.

Multi-granularity motion dropout. Per Eq. (4), we randomly retain either $\boldsymbol{\tau}^{\text{act}}$ or $\boldsymbol{\tau}^{\text{pas}}$ to encourage causal reasoning. We further obtain multi-granularity trajectories by averaging per-pixel trajectories within each patch. During training, we randomly select the granularity, enabling the model to capture both fine-grained pixel control and object-level manipulation.

Occlusion and track dropout. For the obtained trajectory, we randomly mask a subset of it to simulate occlusion and tracking failures that can happen during inference, improving robustness to missing or unreliable tracks.

Inference. At test time, users first specify motion by drawing sparse trajectories (simple curves or strokes on the first image) to indicate the desired direction and magnitude of movement, along with an optional text prompt and target camera poses $\{C_i\}_{i=1}^{T}$. We further perform occlusion-aware masking by approximating visibility ordering from the first-frame depth. The model then jointly denoises both streams, with the second stream being presented to the user.

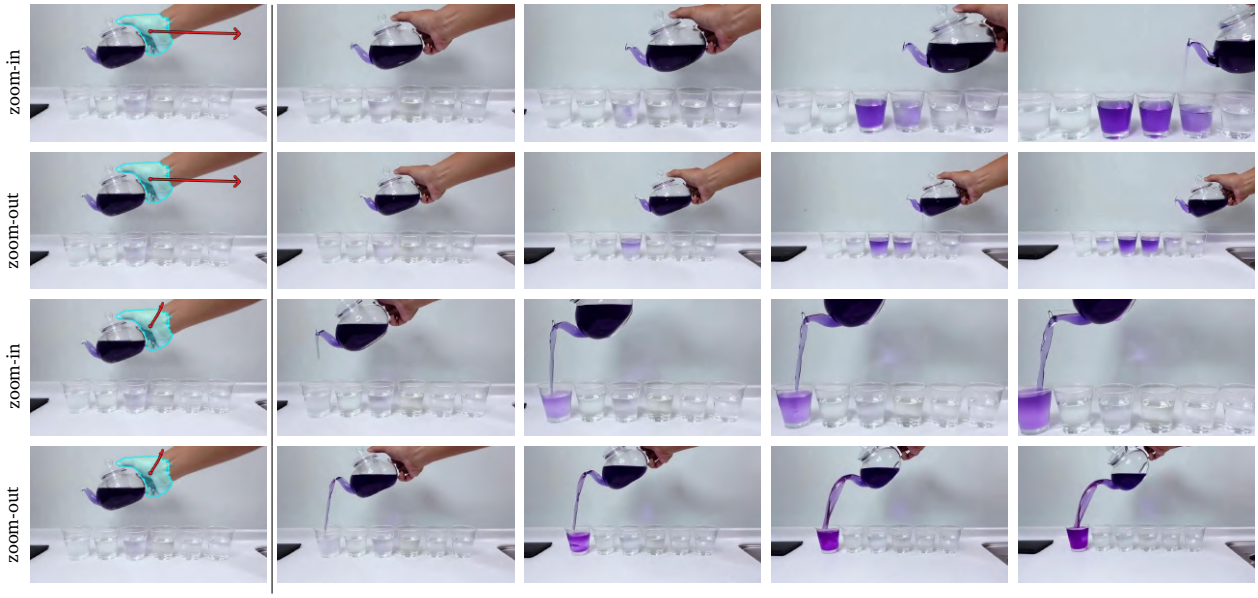

Figure 5: **Controllable generation.** MoRight generates coherent videos under different camera viewpoints. Rows 1–2 and 3–4 share object motion but differ in camera motion, while rows 1–3 and 2–4 share camera motion but differ in object motion.

4. Experiments

4.1. Implementation Details

We build upon the pretrained Wan2.1-14B [47] and fine-tune only the camera encoder, trajectory encoder, and self-attention layers. The trajectory embedding dimension is 64, and the camera encoder uses 32 channels. We train the model with 15K iterations on 64 GPUs with a global batch size of 16, using AdamW with a learning rate of 3×10^{-5} and weight decay 0.001. Trajectory dropout is set to 0.1 and text-conditioning dropout to 0.2.

Following the data curation pipeline in Sec. 3.3, we collect 72K static-view real videos collected from the Internet, 16K paired dynamic-view videos synthesized via camera-control video-to-video generation, and 3.4K synthetic interaction videos from SyncMaster [2]. All videos are processed at 480p resolution. At inference, we sample 35 diffusion steps; generating one video takes approximately 15 minutes on a single A100 GPU.

4.2. Experiment Settings

Evaluation metrics. We evaluate our model across four different aspects: *Video quality*: PSNR and SSIM against reference videos, and FID and FVD [46] for distribution-level similarity. *Camera accuracy*: rotation and translation errors [1, 2, 23] between reference poses and poses estimated from generated videos using ViPE [27]; we report **median** errors across frames to mitigate estimation noise. *Motion accuracy*: end-point error (EPE) [13, 19], the ℓ_2 distance between ground-truth object tracks and predicted tracks extracted with AllTracker; we report the **median** EPE to reduce the impact of outlier tracks. *Motion realism*: Physical Commonsense (PC) and Semantic Adherence (SA) from VideoPhy [4], both 5-point scores normalized to $[0, 1]$. All evaluations are conducted at 480p resolution.

Evaluation Datasets. We evaluate on three datasets spanning diverse interaction scenarios. DynPose-100K [41] is an in-the-wild dataset with highly dynamic camera motion; we manually select 50 videos exhibiting strong viewpoint changes and clear object interactions. WISA [50] is a large-scale physical-dynamics dataset; we select 50 videos from categories including collision, deformation, elasticity, liquid, and rigid-body motion. We

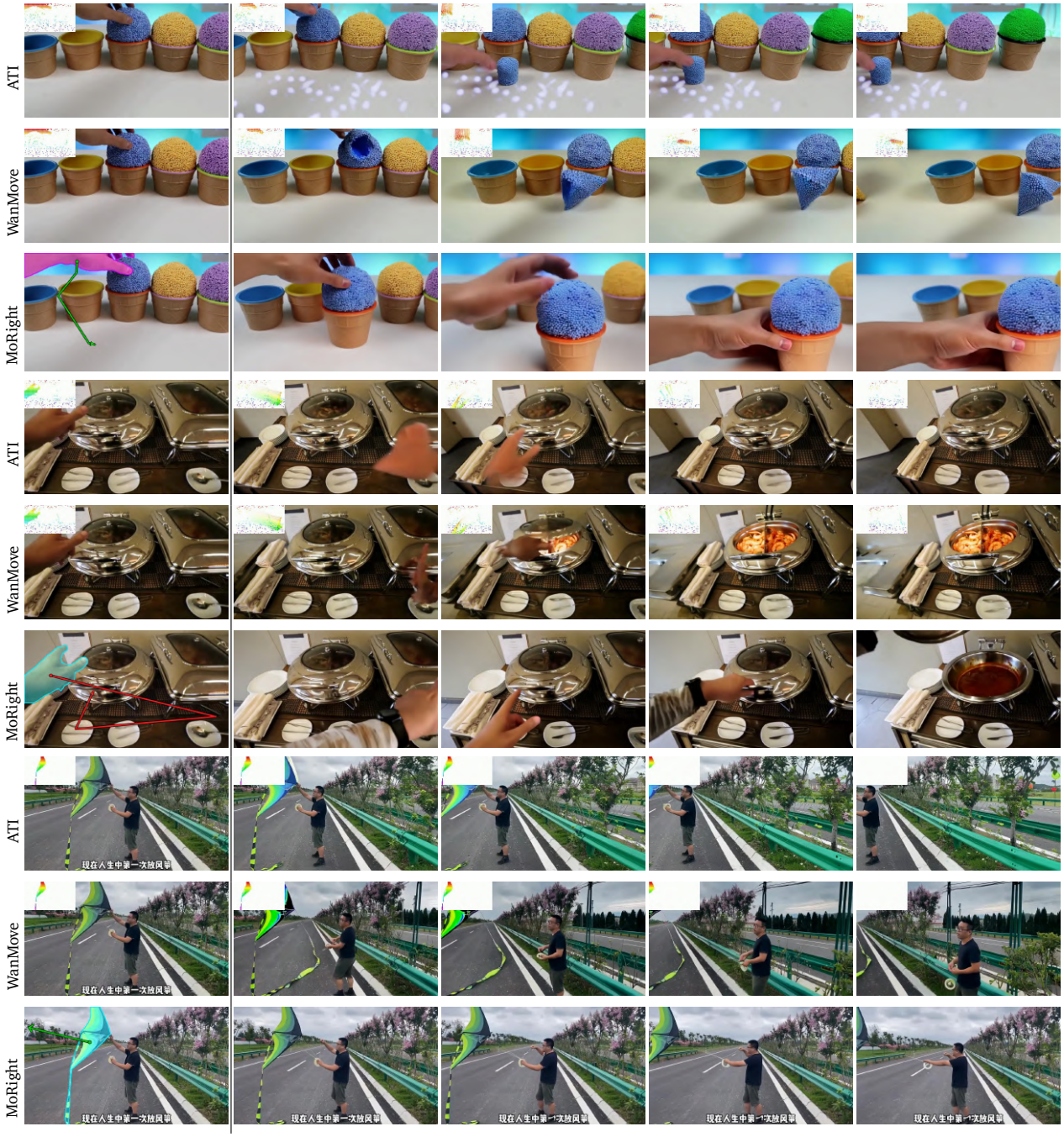

Figure 6: **Qualitative comparison of ATI [48], WanMove [13], and MoRight** on interactive motion generation with camera control. All methods use the same input image. ATI and WanMove rely on pixel-aligned per-frame tracks (top-left), which entangle camera and object motion and require privileged future tracks. In contrast, MoRight uses only reprojected first-frame tracks. The first two rows show active motion reasoning, and the third row shows passive motion reasoning. Our model disentangles camera and object control and produces more coherent interactions.

Table 1: **Controllable video generation on DynPose-100K and Cooking.** We compare models with camera and object motion control. Tracking-based methods (MP, ATI, WanMove) require privileged foreground/background tracks, while MoRight uses only first-frame reprojected trajectories and camera poses. Despite weaker inputs, MoRight achieves comparable visual quality and more accurate motion control. All methods use the Wan2.1-14B backbone; * denotes models reimplemented and trained by us. Best and second-best results are marked in **bold** and underline.

		DynPose-100K					Cooking				
Method	Full info	PSNR ↑	SSIM ↑	Rot ↓	Trans ↓	EPE ↓	PSNR ↑	SSIM ↑	Rot ↓	Trans ↓	EPE ↓
Wan2.1 [47]	×	11.23	0.435	-	-	–	14.23	0.527	-	-	–
Gen3c* [40]	×	12.45	0.507	5.46	4.09	-	15.37	**0.613**	**1.97**	**10.03**	-
MP* [19]	✓	11.72	0.455	6.76	6.04	7.56	15.68	0.564	2.50	12.24	**4.25**
ATI [48]	✓	13.18	0.493	5.62	6.54	8.43	15.93	0.582	4.25	16.94	5.87
WanMove [13]	✓	**13.91**	**0.521**	**4.12**	**3.56**	8.05	16.42	0.589	2.93	13.27	5.47
Ours	×	12.30	0.457	4.55	4.61	**7.64**	**16.44**	0.594	2.16	10.11	4.27

Table 2: **Interactive motion generation on WISA [50] and Cooking.** We compare motion-conditioned video generation models on WISA and Cooking, evaluating video quality (FID, FVD) and motion realism (PC, SA). Prior methods require detailed motion captions with full interaction descriptions, while MoRight uses only a single active motion description yet achieves comparable quality with stronger physical commonsense reasoning. All methods use the Wan2.1-14B backbone for fair comparison; * denotes models reimplemented and trained by us. Best and second-best results are marked in **bold** and underline.

		WISA				Cooking			
Method	Full info	FID ↓	FVD ↓	PC ↑	SA ↑	FID ↓	FVD ↓	PC ↑	SA ↑
MP* [19]	✓	57.29	975.94	0.75	0.82	43.49	759.53	0.87	0.89
ATI [48]	✓	69.80	990.82	0.75	**0.83**	55.80	881.94	0.85	**0.90**
WanMove [13]	✓	61.34	1088.23	0.73	**0.83**	53.51	882.90	0.84	0.87
Ours	×	**52.95**	**876.03**	**0.76**	0.82	**39.94**	**730.46**	**0.88**	0.89

further collect 50 real-world cooking videos, featuring complex hand-object interactions.

4.3. Disentangled Camera-Object Motion Control

Existing works on controllable video generation typically focus on either camera or object motion in isolation. Motion-conditioned baselines [13, 19, 48] typically receive privileged signals of both foreground and background tracks of all the pixel trajectories for control, while our method only uses reprojected trajectories defined on the canonical frame, without access to future-frame pixel trajectories. We compare our method with state-of-the-art baselines to evaluate the quality in motion control and camera control. While challenging, this evaluation allows us to demonstrate our model's unique ability to generate faithful motion from disentangled controls—a task that is fundamentally more difficult than baselines.

Baselines and setup. We compare with several controllable video generation methods. Wan2.1 [47] is our base model without motion control. Gen3C [40] only supports camera control. Recent state-of-the-art motion-conditioned models (Motion Prompting (MP) [19], ATI [48] and WanMove [13] all takes dense pixel

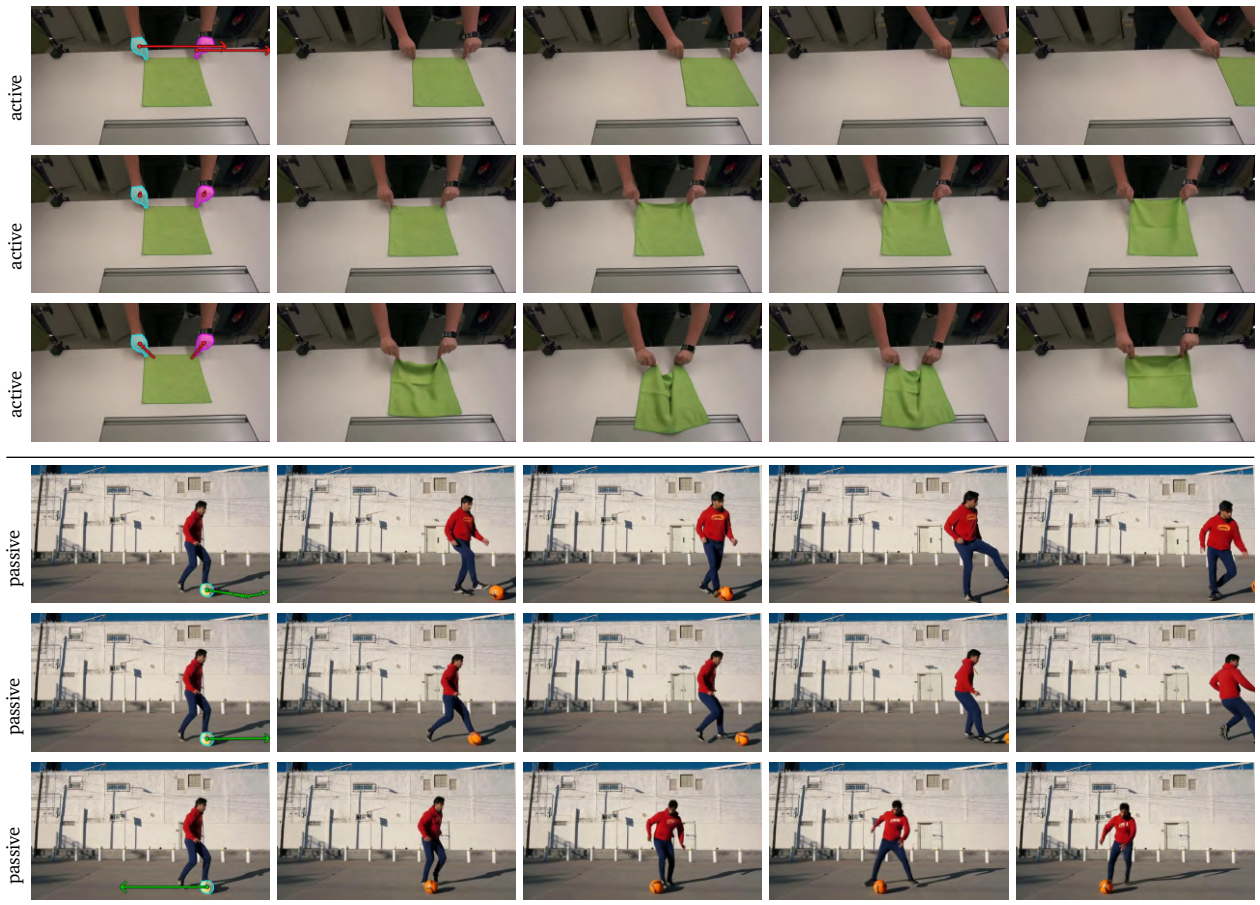

Figure 7: **Causal interaction reasoning.** In the first 3 rows, we provide active motion (*e.g.*, hand movement) as input, and the model infers the resulting passive motion (*e.g.*, cloth movement). In the last 3 rows, we provide passive motion (*e.g.*, ball movement), and the model infers the corresponding active motion (*e.g.*, human movement).

tracks as input. For a fair comparison, we retrain Gen3C and MP in our setup, and all methods share the same Wan2.1-14B backbone. Evaluation is conducted on DynPose-100K [41] and Cooking dataset. Motion accuracy is evaluated in future-frame pixel space via EPE for all methods.

Results. Quantitative results are provided in Tab. 1, with controllable results in Fig. 5, qualitative comparisons in Fig. 6. On DynPose-100K [41], WanMove [13] achieves the best overall numbers. Our method is slightly behind under highly dynamic camera motion, where errors in camera pose estimation and trajectory reprojection can degrade the input control signals. Nevertheless, MoRight attains comparable controllability to methods that rely on privileged future-frame tracking information and achieves the best EPE for object motion accuracy. We observe that ATI [48] and WanMove [13], which couple camera and object motion in a single tracking signal, tend to favor the dominant motion mode in highly dynamic settings—sometimes sacrificing camera accuracy or object tracking fidelity. On the Cooking benchmark, our method achieves the best overall performance in both visual quality and motion control accuracy.

4.4. Motion Causality Modeling

Setup. We evaluate the causality of modeling motion on WISA [50] and Cooking, measuring both generation quality (FID, FVD) and motion realism (PC, SA). We compare with MP* [19], ATI [48], and WanMove [13], all

Table 3: **Ablation of motion controllability and reasoning on the Cooking benchmark.** We ablate architectural choices, causal reasoning, hybrid training, and different motion input granularities. Our full model achieves the best overall performance across photometric quality, controllability, and motion realism, while remaining robust to different motion granularities and input conditions (active and passive).

	Photometric				Controllability			Motion	
Setting	FID ↓	FVD ↓	PSNR ↑	SSIM ↑	Rot ↓	Trans ↓	EPE ↓	PC ↑	SA ↑
cascaded	41.74	728.80	15.98	0.569	2.69	11.50	5.05	0.87	0.89
w/o fixed view	51.17	997.83	14.15	0.515	3.36	14.57	14.30	0.87	0.89
w/o reasoning	44.04	784.19	15.55	0.562	2.88	12.49	5.05	0.87	0.88
w/o mixed training	41.94	808.96	16.29	0.583	2.22	12.80	**4.09**	0.87	0.89
ours (coarse)	**39.83**	**725.88**	**16.45**	**0.594**	2.21	10.98	4.37	**0.88**	0.88
ours (passive)	44.20	838.67	15.99	0.588	2.21	11.04	7.27	0.87	0.88
ours (active)	39.94	730.46	16.44	0.594	**2.16**	**10.11**	4.27	**0.88**	0.89

following the same input protocol as Sec. 4.3. Every model receives *active motion* representing the user-specified action (e.g., a hand pushing an object); the goal is to generate plausible interaction outcomes. Baseline methods use their original prompts containing both motion descriptions and expected consequences. Our model receives only the active motion description, without specifying passive outcomes, and must infer the resulting interactions.

Results. Quantitative results are reported in Tab. 2. MoRight achieves the highest PC score on WISA, indicating strong physical commonsense, and the best video quality (FID, FVD) on both datasets. For SA, we rewrite the input prompt to remove passive motion descriptions and avoid information leakage; consequently, our score is slightly lower than methods that use full prompts containing both actions and outcomes, yet remains comparable. This confirms that MoRight's generations stay semantically aligned with intended outcomes while demonstrating genuine causal motion reasoning rather than relying on prompt-supplied answers. Qualitative examples of both reasoning modes—*forward* (action → reaction) and *inverse* (reaction → action)—are shown in Fig. 7. Our model generates plausible reactions when providing the active motion, and can reason the meaningful active motion when providing passive motion.

4.5. Ablation Studies

We ablate model design, training strategies, and input conditions on the Cooking dataset in Tab. 3.

Model design and training. *Cascaded pipeline* (row 1): a naive solution for disentangling camera-object motion is to first generate motion-controlled video under a static camera, followed by a Gen3C-style camera controller to move the camera. However, this approach introduces error accumulation between two stages, yielding larger control errors. *W/o fixed-view branch* (row 2): we only train with dynamic camera views and jointly encode the reprojected tracks and camera embeddings, removing the canonical-view anchor. The model struggles to disentangle camera and object motion, resulting in significantly worse camera and tracking accuracy. *W/o motion reasoning* (row 3): we disable active/passive decomposition during training. This approach increases FID/FVD and reduces PC, indicating degraded interaction quality. *W/o mixed supervision* (row 4): We only train the model on paired data. It slightly degrades camera accuracy, as the paired subset contains limited camera motion diversity.

Input conditions. We vary the motion input configuration to evaluate the robustness of our models. Specifically,

we evaluate with coarse segment-level trajectories vs. fine-grained pixel tracks, as well as active vs. passive motion inputs. Performance remains stable across all settings, confirming that MoRight flexibly handles different motion granularities and types while maintaining strong controllability and causal reasoning capability.

5. Conclusion

We present MoRight, a unified framework for controllable and interaction-aware video generation. MoRight addresses two key limitations of prior motion-controlled methods: (1) entangled camera and object motion, resolved through a dual-stream design that enables independent control of object trajectories and camera viewpoints; and (2) limited causal reasoning, addressed by decomposing motion into active (user-driven) and passive (consequence) components to learn action–response dynamics. At inference, MoRight supports both forward prediction—generating scene outcomes from active motion—and inverse reasoning—inferring actions from desired passive results. Experiments on DynPose-100K, WISA, and Cooking show strong performance in generation quality, motion control, and interaction awareness, establishing MoRight as a step toward more interactive and physically grounded video generation.

References

[1] Bai, J., Xia, M., Fu, X., Wang, X., Mu, L., Cao, J., Liu, Z., Hu, H., Bai, X., Wan, P., et al.: Recammaster: Camera-controlled generative rendering from a single video. arXiv preprint arXiv:2503.11647 (2025) 8

[2] Bai, J., Xia, M., Wang, X., Yuan, Z., Fu, X., Liu, Z., Hu, H., Wan, P., Zhang, D.: Syncammaster: Synchronizing multi-camera video generation from diverse viewpoints. Proc. ICLR (2025) 7, 8

[3] Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., Liu, J., Liu, C., Liu, Y., Liu, D., Liu, S., Lu, D., Luo, R., Lv, C., Men, R., Meng, L., Ren, X., Ren, X., Song, S., Sun, Y., Tang, J., Tu, J., Wan, J., Wang, P., Wang, P., Wang, Q., Wang, Y., Xie, T., Xu, Y., Xu, H., Xu, J., Yang, Z., Yang, M., Yang, J., Yang, A., Yu, B., Zhang, F., Zhang, H., Zhang, X., Zheng, B., Zhong, H., Zhou, J., Zhou, F., Zhou, J., Zhu, Y., Zhu, K.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025) 6, 7, 18

[4] Bansal, H., Lin, Z., Xie, T., Zong, Z., Yarom, M., Bitton, Y., Jiang, C., Sun, Y., Chang, K.W., Grover, A.: Videophy: Evaluating physical commonsense for video generation. arXiv preprint arXiv:2406.03520 (2024) 8, 20

[5] Bansal, H., Peng, C., Bitton, Y., Goldenberg, R., Grover, A., Chang, K.W.: Videophy-2: A challenging action-centric physical commonsense evaluation in video generation. arXiv preprint arXiv:2503.06800 (2025) 20

[6] Bar, A., Zhou, G., Tran, D., Darrell, T., LeCun, Y.: Navigation world models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 15791–15801 (2025) 2

[7] Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S.W., Fidler, S., Kreis, K.: Align your latents: High-resolution video synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22563–22575 (2023) 2

[8] Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., Ramesh, A.: Video generation models as world simulators. OpenAI technical reports (2024), https://openai.com/research/video-generation-models-as-world-simulators 2

[9] Burgert, R., Xu, Y., Xian, W., Pilarski, O., Clausen, P., He, M., Ma, L., Deng, Y., Li, L., Mousavi, M., et al.: Go-with-the-flow: Motion-controllable video diffusion models using real-time warped noise. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 13–23 (2025) 2

[10] Chen, B., Jiang, H., Liu, S., Gupta, S., Li, Y., Zhao, H., Wang, S.: Physgen3d: Crafting a miniature interactive world from a single image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6178–6189 (2025) 3

[11] Chen, H., Zhang, Y., Cun, X., Xia, M., Wang, X., Weng, C., Shan, Y.: Videocrafter2: Overcoming data limitations for high-quality video diffusion models. arXiv preprint arXiv:2401.09047 (2024) 2

[12] Chen, Y., Men, Y., Yao, Y., Cui, M., Bo, L.: Perception-as-control: Fine-grained controllable image animation with 3d-aware motion representation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14380–14389 (2025) 3

[13] Chu, R., He, Y., Chen, Z., Zhang, S., Xu, X., Xia, B., Wang, D., Yi, H., Liu, X., Zhao, H., et al.: Wan-move: Motion-controllable video generation via latent trajectory guidance. arXiv preprint arXiv:2512.08765 (2025) 2, 3, 8, 9, 10, 11, 21, 24, 26

[14] Cosmos, T.: Cosmos world foundation model platform for physical ai. arXiv preprint arXiv:2501.03575 (2025) 4

[15] Doersch, C., Yang, Y., Vecerik, M., Gokay, D., Gupta, A., Aytar, Y., Carreira, J., Zisserman, A.: Tapir: Tracking any point with per-frame initialization and temporal refinement. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10061–10072 (2023) 3

[16] Elfwing, S., Uchibe, E., Doya, K.: Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. Neural networks **107**, 3–11 (2018) 18

[17] Fu, X., Tang, S., Shi, M., Liu, X., Gu, J., Liu, M.Y., Lin, D., Lin, C.H.: Plenoptic video generation. arXiv preprint arXiv:2601.05239 (2025) 7

[18] Fu, X., Tang, S., Shi, M., Liu, X., Gu, J., Liu, M.Y., Lin, D., Lin, C.H.: Plenoptic video generation. arXiv preprint arXiv:2601.05239 (2026) 6

[19] Geng, D., Herrmann, C., Hur, J., Cole, F., Zhang, S., Pfaff, T., Lopez-Guevara, T., Doersch, C., Aytar, Y., Rubinstein, M., Sun, C., Wang, O., Owens, A., Sun, D.: Motion prompting: Controlling video generation with motion trajectories. arXiv preprint arXiv:2412.02700 (2024) 2, 3, 5, 8, 10, 11

[20] Gillman, N., Herrmann, C., Freeman, M., Aggarwal, D., Luo, E., Sun, D., Sun, C.: Force prompting: Video generation models can learn and generalize physics-based control signals. arXiv preprint arXiv:2505.19386 (2025) 3

[21] Gu, Z., Yan, R., Lu, J., Li, P., Dou, Z., Si, C., Dong, Z., Liu, Q., Lin, C., Liu, Z., et al.: Diffusion as shader: 3d-aware video diffusion for versatile video generation control. In: SIGGRAPH (2025) 3

[22] Harley, A.W., You, Y., Sun, X., Zheng, Y., Raghuraman, N., Gu, Y., Liang, S., Chu, W.H., Dave, A., You, S., et al.: Alltracker: Efficient dense point tracking at high resolution. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5253–5262 (2025) 3, 6

[23] He, H., Xu, Y., Guo, Y., Wetzstein, G., Dai, B., Li, H., Yang, C.: Cameractrl: Enabling camera control for text-to-video generation. arXiv preprint arXiv:2404.02101 (2024) 8

[24] He, X., Peng, C., Liu, Z., Wang, B., Zhang, Y., Cui, Q., Kang, F., Jiang, B., An, M., Ren, Y., et al.: Matrix-game 2.0: An open-source real-time and streaming interactive world model. arXiv preprint arXiv:2508.13009 (2025) 2

[25] Hong, W., Ding, M., Zheng, W., Liu, X., Tang, J.: Cogvideo: Large-scale pretraining for text-to-video generation via transformers. arXiv preprint arXiv:2205.15868 (2022) 2

[26] Hong, Y., Mei, Y., Ge, C., Xu, Y., Zhou, Y., Bi, S., Hold-Geoffroy, Y., Roberts, M., Fisher, M., Shechtman, E., et al.: Relic: Interactive video world model with long-horizon memory. arXiv preprint arXiv:2512.04040 (2025) 2

[27] Huang, J., Zhou, Q., Rabeti, H., Korovko, A., Ling, H., Ren, X., Shen, T., Gao, J., Slepichev, D., Lin, C.H., et al.: Vipe: Video pose engine for 3d geometric perception. arXiv preprint arXiv:2508.10934 (2025) 6, 7, 8, 18

[28] Jin, W., Dai, Q., Luo, C., Baek, S.H., Cho, S.: Flovd: Optical flow meets video diffusion model for enhanced camera-controlled video synthesis. In: Proc. CVPR (2025) 3

[29] Karaev, N., Rocco, I., Graham, B., Neverova, N., Vedaldi, A., Rupprecht, C.: Cotracker: It is better to track together. In: Proc. ECCV (2024) 3

[30] Li, Q., Xing, Z., Wang, R., Zhang, H., Dai, Q., Wu, Z.: Magicmotion: Controllable video generation with dense-to-sparse trajectory guidance. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12112–12123 (2025) 3

[31] Li, Z., Niklaus, S., Snavely, N., Wang, O.: Neural scene flow fields for space-time view synthesis of dynamic scenes. In: CVPR (2021) 3

[32] Li, Z., Yu, H.X., Liu, W., Yang, Y., Herrmann, C., Wetzstein, G., Wu, J.: Wonderplay: Dynamic 3d scene generation from a single image and actions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9080–9090 (2025) 3

[33] Lin, H., Chen, S., Liew, J., Chen, D.Y., Li, Z., Shi, G., Feng, J., Kang, B.: Depth anything 3: Recovering the visual space from any views. arXiv preprint arXiv:2511.10647 (2025) 5

[34] Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022) 4

[35] Liu, S., Ren, Z., Gupta, S., Wang, S.: Physgen: Rigid-body physics-grounded image-to-video generation. In: European Conference on Computer Vision. pp. 360–378. Springer (2024) 3

[36] Lv, J., Huang, Y., Yan, M., Huang, J., Liu, J., Liu, Y., Wen, Y., Chen, X., Chen, S.: Gpt4motion: Scripting physical motions in text-to-video generation via blender-oriented gpt planning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1430–1440 (2024) 3

[37] Ma, W.D.K., Lewis, J.P., Kleijn, W.B.: Trailblazer: Trajectory control for diffusion-based video generation. arXiv preprint arXiv:2401.00896 (2023) 3

[38] Pan, C., Yaman, B., Nesti, T., Mallik, A., Allievi, A.G., Velipasalar, S., Ren, L.: Vlp: Vision language planning for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14760–14769 (2024) 3

[39] Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K.V., Carion, N., Wu, C.Y., Girshick, R., Dollár, P., Feichtenhofer, C.: Sam 2: Segment anything in images and videos. arXiv preprint arXiv:2408.00714 (2024), https://arxiv.org/abs/2408.00714 6, 7, 18

[40] Ren, X., Shen, T., Huang, J., Ling, H., Lu, Y., Nimier-David, M., Müller, T., Keller, A., Fidler, S., Gao, J.: Gen3c: 3d-informed world-consistent video generation with precise camera control. In: CVPR. pp. 6121–6132 (2025) 10, 18

[41] Rockwell, C., Tung, J., Lin, T.Y., Liu, M.Y., Fouhey, D.F., Lin, C.H.: Dynamic camera poses and where to find them. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12444–12455 (2025) 8, 11

[42] Shi, X., Huang, Z., Wang, F.Y., Bian, W., Li, D., Zhang, Y., Zhang, M., Cheung, K.C., See, S., Qin, H., et al.: Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling. In: ACM SIGGRAPH 2024 Conference Papers. pp. 1–11 (2024) 3

[43] Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E., Sebe, N.: First order motion model for image animation. In: NeurIPS (2019) 3

[44] Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. Neurocomputing **568**, 127063 (2024) 18

[45] Tulyakov, S., Liu, M.Y., Yang, X., Kautz, J.: Mocogan: Decomposing motion and content for video generation. In: CVPR (2018) 3

[46] Unterthiner, T., van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., Gelly, S.: Towards accurate generative models of video: A new metric & challenges. arXiv preprint arXiv:1812.01717 (2018) 8

[47] Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025) 2, 4, 8, 10, 18

[48] Wang, A., Huang, H., Fang, Z., Yang, Y., Ma, C.: Ati: Any trajectory instruction for controllable video generation. arXiv preprint **arXiv:2505.22944** (2025) 2, 9, 10, 11, 21, 24, 26

[49] Wang, J., Zhang, Y., Zou, J., Zeng, Y., Wei, G., Yuan, L., Li, H.: Boximator: Generating rich and controllable motions for video synthesis. arXiv preprint arXiv:2402.01566 (2024) 2, 3

[50] Wang, J., Ma, A., Cao, K., Zheng, J., Zhang, Z., Feng, J., Liu, S., Ma, Y., Cheng, B., Leng, D., et al.: Wisa: World simulator assistant for physics-aware text-to-video generation. arXiv preprint arXiv:2503.08153 (2025) 8, 10, 11

[51] Wang, R., Xu, S., Dai, C., Xiang, J., Deng, Y., Tong, X., Yang, J.: Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision (2024), https://arxiv.org/abs/2410.19115 19

[52] Wang, X., Zhu, Z., Huang, G., Chen, X., Zhu, J., Lu, J.: Drivedreamer: Towards real-world-drive world models for autonomous driving. In: European conference on computer vision. pp. 55–72. Springer (2024) 2

[53] Wang, Y., Bilinski, P., Bremond, F., Dantcheva, A.: G3AN: Disentangling appearance and motion for video generation. In: CVPR (2020) 3

[54] Wang, Y., Bremond, F., Dantcheva, A.: Inmodegan: Interpretable motion decomposition generative adversarial network for video generation. arXiv preprint arXiv:2101.03049 (2021) 3

[55] Wang, Z., Yuan, Z., Wang, X., Chen, T., Xia, M., Luo, P., Shan, Y.: Motionctrl: A unified and flexible motion controller for video generation. In: SIGGRAPH (2024) 2, 3

[56] Wu, W., Li, Z., Gu, Y., Zhao, R., He, Y., Zhang, D.J., Shou, M.Z., Li, Y., Gao, T., Zhang, D.: Draganything: Motion control for anything using entity representation. In: Proc. ECCV (2024) 3

[57] Xing, J., Mai, L., Ham, C., Huang, J., Mahapatra, A., Fu, C.W., Wong, T.T., Liu, F.: Motioncanvas: Cinematic shot design with controllable image-to-video generation. In: SIGGRAPH (2025) 3

[58] Yang, M., Du, Y., Ghasemipour, K., Tompson, J., Schuurmans, D., Abbeel, P.: Learning interactive real-world simulators. arXiv preprint arXiv:2310.06114 **1**(2), 6 (2023) 2

[59] Yang, X., Li, B., Zhang, Y., Yin, Z., Bai, L., Ma, L., Wang, Z., Cai, J., Wong, T.T., Lu, H., et al.: Vlipp: Towards physically plausible video generation with vision and language informed physical prior. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12360–12370 (2025) 3

[60] Yang, Z., Ge, W., Li, Y., Chen, J., Li, H., An, M., Kang, F., Xue, H., Xu, B., Yin, Y., et al.: Matrix-3d: Omnidirectional explorable 3d world generation. arXiv preprint arXiv:2508.08086 (2025) 2

[61] Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al.: Cogvideox: Text-to-video diffusion models with an expert transformer. arXiv preprint arXiv:2408.06072 (2024) 2

[62] Yin, S., Wu, C., Liang, J., Shi, J., Li, H., Ming, G., Duan, N.: Dragnuwa: Fine-grained control in video generation by integrating text, image, and trajectory. arXiv preprint arXiv:2308.08089 (2023) 3

[63] Yu, S., Fang, J.Z., Zheng, J., Sigurdsson, G.A., Ordonez, V., Piramuthu, R., Bansal, M.: Zero-shot controllable image-to-video animation via motion decomposition. In: ACM MM (2024) 3

[64] Zhang, Q., Wang, C., Siarohin, A., Zhuang, P., Xu, Y., Yang, C., Lin, D., Zhou, B., Tulyakov, S., Lee, H.Y.: SceneWiz3D: Towards text-guided 3D scene composition. In: Proc. CVPR (2024) 5

[65] Zhang, Z., Long, F., Qiu, Z., Pan, Y., Liu, W., Yao, T., Mei, T.: Motionpro: A precise motion controller for image-to-video generation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 27957–27967 (2025) 3

Appendix – MoRight: Motion Control Done Right

A. Implementation Details

A.1. Network Architecture

Our model builds on the Wan2.1 I2V-14B [47]. We first encode the two-view videos and then concatenate the tokens along the temporal dimension before feeding them into the model. The two streams share the same spatial RoPE [44] embeddings but use different temporal indices. The object tracking condition, represented as a trajectory map, is encoded by a lightweight temporal encoder with RMSNorm, SiLU [16], and two $3 \times 1 \times 1$ Conv3D layers that downsample the temporal dimension by $4\times$ to match the Wan latent resolution. For camera motion control, we follow Gen3C [40] by warping the first frame with the camera trajectory and encoding it with the VAE, producing features in the same latent space and resolution.

Both camera and tracking features are linearly projected to the Wan hidden dimension (5120) and added to the video tokens before the self-attention layer of each Wan2.1 transformer block. During training, we only train the lightweight temporal encoder and the self-attention layers together with the camera and tracking encoders in each block, and freeze other parts of the network.

A.2. Training Data Curation

In training data curation, we need to identify active and passive object and its motion in a given video. We first identify those objects by querying Qwen3-VL [3] and use SAM2 [39] for video object segmentation. The system prompt to Qwen3-VL [3] is shown in Fig. 8. We further use Qwen3 to rewrite video captions by decomposing object motion into active or passive descriptions, ensuring that each rewritten caption contains only one type of motion. During training, the original caption and the rewritten caption are randomly sampled with equal probability, encouraging the model to infer plausible interaction consequences. To generate paired multi-view data, we select videos from our collected Internet videos with nearly static cameras using the camera poses provided by ViPE [27], requiring a maximum rotation of $0.5°$ and translation of $5\,\text{mm}$.

System: You are a video understanding specialist. Follow all rules exactly and output only valid JSON.

- `active_dynamic`: objects that move by their own power or actuation (e.g., person, hand, animal, robot, drone, car, bus, train, boat, ship, airplane, motorcycle).
- `passive_dynamic`: objects that move only because of other objects or forces. If a person/hand is visible manipulating an item, that item is `passive_dynamic`.

User: You are given a video. Identify all objects that actually move:

- `active_dynamic`: objects moving by their own power (or actuation).
- `passive_dynamic`: objects that move only because other objects or natural forces cause them to move.

Figure 8: Prompt used for active and passive object identification for Qwen3 [3] in data curation pipeline.

A.3. Training

During training, we apply several data augmentations to improve robustness. For each sample, we randomly simplify the input trajectories with probability 0.5, where tracks are averaged per object such that all pixels of the same object share a single trajectory. To encourage the model to reason about motion causality, we randomly provide *active* or *passive* motion tracks with probabilities 0.8 and 0.2, respectively. We further apply motion dropout by randomly dropping visible tracks with probability 0.2 to simulate occlusion and tracking

errors commonly observed in off-the-shelf trackers at inference time. In addition, we randomly truncate tracks after a sampled middle frame to simulate partial observations of motion. During training, we randomly sample between 500 and 2000 tracks per iteration. At inference time, we fix the number of input tracks to 1500 for all experiments to ensure consistent evaluation. Finally, since multi-view supervision is critical for learning camera–object disentanglement, we control the sampling ratio of multi-view and single-view training samples, ensuring that single-view data is sampled at a lower rate to prevent the model from overfitting to single-view motion patterns.

A.4. Inference

At inference time, users can freely select objects in the first frame and specify their motion trajectories. Motion control can be provided either in a coarse manner, where the entire object moves with a shared trajectory, or in a fine-grained manner using sparse point tracks. To facilitate interaction-driven editing, we also provide several simple motion primitives for hand interactions, such as push, pull (along a specified direction), and reach (toward a target location). For passive motion control, users can define arbitrary 2D trajectories; in practice, we use straight-line trajectories with different directions when performing inverse reasoning (Fig. 12).

To enable flexible control, we implement an interactive GUI as shown in Fig. 9. Starting from a single input image, users draw motion trajectories directly on the first frame while specifying camera motion independently through a sequence of camera poses (the first frame is treated as the identity pose). The interface supports trajectory visualization across time steps and occlusion checking using the first-frame depth estimated by MoGe [51], enabling intuitive editing of object dynamics and camera viewpoints during generation.

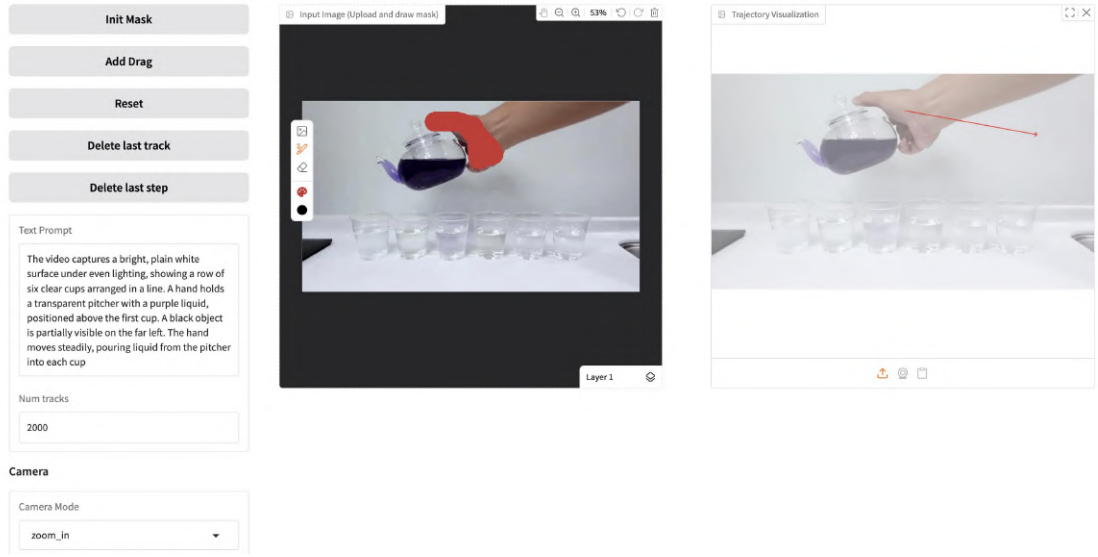

Figure 9: **Interactive demo interface.** Our system enables users to control both object and camera motion from a single image. Users draw trajectories on the first frame to specify object motion (active or passive), either by moving a selected region using keypoint trajectories or by defining fine-grained motion paths for detailed control.

A.5. Evaluation

The **Cooking Benchmark** is constructed from real-world cooking videos collected from YouTube that contain rich hand–object interactions. These scenes involve diverse manipulation behaviors such as pushing, cutting, and picking, making them a suitable testbed for evaluating interactive motion reasoning. The benchmark contains 50 video clips covering a variety of kitchen environments and object interactions.

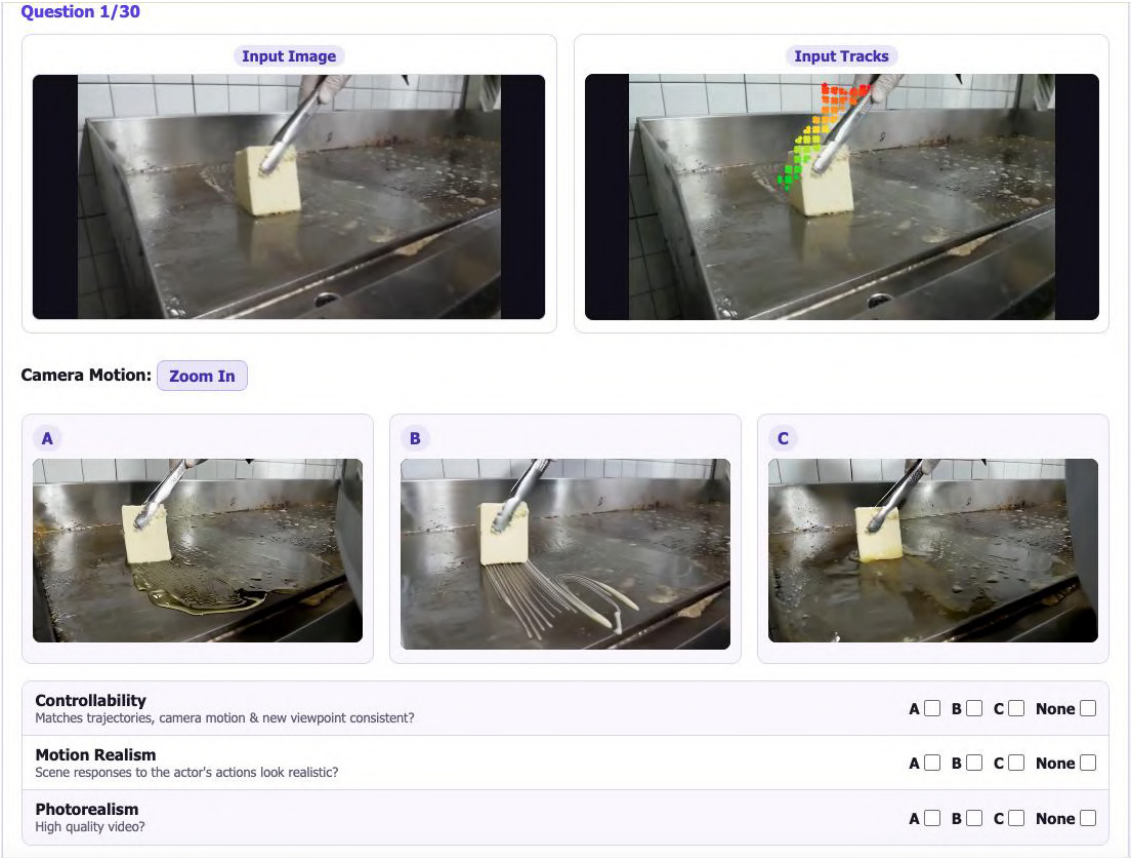

Figure 10: **Human perceptual evaluation interface.** Given an input image, object trajectories, and a target camera motion, participants evaluate generated videos under three criteria: *Controllability* (matching object tracks and camera motion), *Motion Realism* (physically plausible interactions and scene responses), and *Photorealism* (overall visual quality). For each criterion, participants select the best video (multiple selections allowed for ties, or *None* if none satisfy it). The study contains 30 randomly selected video sets with shuffled candidate order.

For motion quality evaluation, we adopt **Physical Commonsense (PC)** and **Semantic Adherence (SA)**, from [4, 5]. PC measures whether the generated video follows real-world physical behaviors. SA evaluates whether the generated video remains semantically consistent with the input text prompt. For SA, we use the original caption of each video as the evaluation prompt. We use the automatic evaluation rater provided to score the generated videos and report the resulting normalized scores across different methods.

B. Human Evaluation Analysis

Besides the automatic objective metrics, we conduct a human perceptual evaluation to assess generation quality. For each method, the inputs include an image, object motion trajectories, and camera motion controls. ATI and WanMove take pixel-aligned per-frame tracks projected from privileged 3D trajectories, including both foreground and background tracks as well as full interaction trajectories (active and passive). In contrast, our method only takes first-frame active motion trajectories without privileged information and relies on the model to infer plausible interactions.

We randomly sample 30 examples from the union of all test datasets. For each example, candidate videos from different methods are displayed in randomly shuffled order to avoid positional bias. Participants evaluate the results under three criteria: *Controllability* (whether object motion follows the input trajectories and the

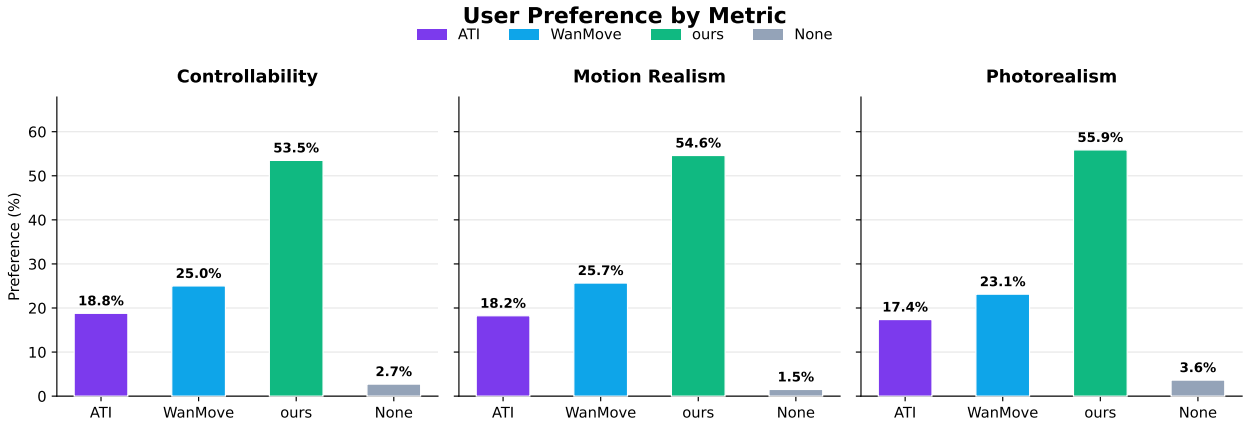

Figure 11: **Human perceptual evaluation.** From 330 responses by 11 participants, our method is preferred across controllability, motion realism, and photorealism, outperforming ATI [48] and WanMove [13], which rely on privileged 3D tracks but lack interaction reasoning.

camera follows the specified motion), *Motion Realism* (whether interactions and scene responses are physically plausible), and *Photorealism* (visual quality). For each criterion, participants select the best video among the candidates; multiple selections are allowed for ties, and *None* can be chosen if none satisfy the criterion. The interface contains 30 questions corresponding to the sampled examples. After applying a sanity check to remove unreliable submissions, we obtain responses from 11 valid participants, resulting in 330 responses for each criterion. The evaluation layout is shown in Fig. 10.

The results of the human perceptual evaluation are summarized in Fig. 11. Our method is preferred in the majority of cases, achieving 53.5%, 54.6%, and 55.9% preference for controllability, motion realism, and photorealism, respectively, outperforming ATI [48] (18.8%, 18.2%, 17.4%) and WanMove [13] (25.0%, 25.7%, 23.1%). Despite using privileged 3D trajectories and full interaction tracks, the baselines lack interaction motion reasoning capability and rely on entangled camera–object motion representations, leading to inferior results. In contrast, our approach explicitly reasons about interactions while disentangling camera and object motion, resulting in more controllable and realistic video generation.

C. More Qualitative Results

C.1. Interactive Motion Generation

We demonstrate the causal reasoning ability of our model in Fig. 12. We generate videos and select the first-stream generation (static view) to visualize the interaction dynamics. The input motion tracks are overlaid on the generated frames, where colored tracks indicate either user actions (active motion) or passive trajectories. Our model supports both forward and inverse reasoning. In forward reasoning (first two samples), the model predicts plausible scene consequences given the specified active motion. In inverse reasoning (last sample), the model infers feasible driving actions that could lead to the observed passive outcomes. These examples highlight the model's ability to reason about causal interactions between actions and objects.

C.2. Disentangled Camera-Object Control

We present additional disentangled controllable generation results in Fig. 13. Object motion trajectories are overlaid on the input image. We demonstrate three different object motions and three different camera motions (orbit-left, zoom-in, and zoom-out), resulting in 9 generated videos in 3 group. Each group shares the same object motion while varying the camera viewpoint, highlighting the model's ability to maintain consistent

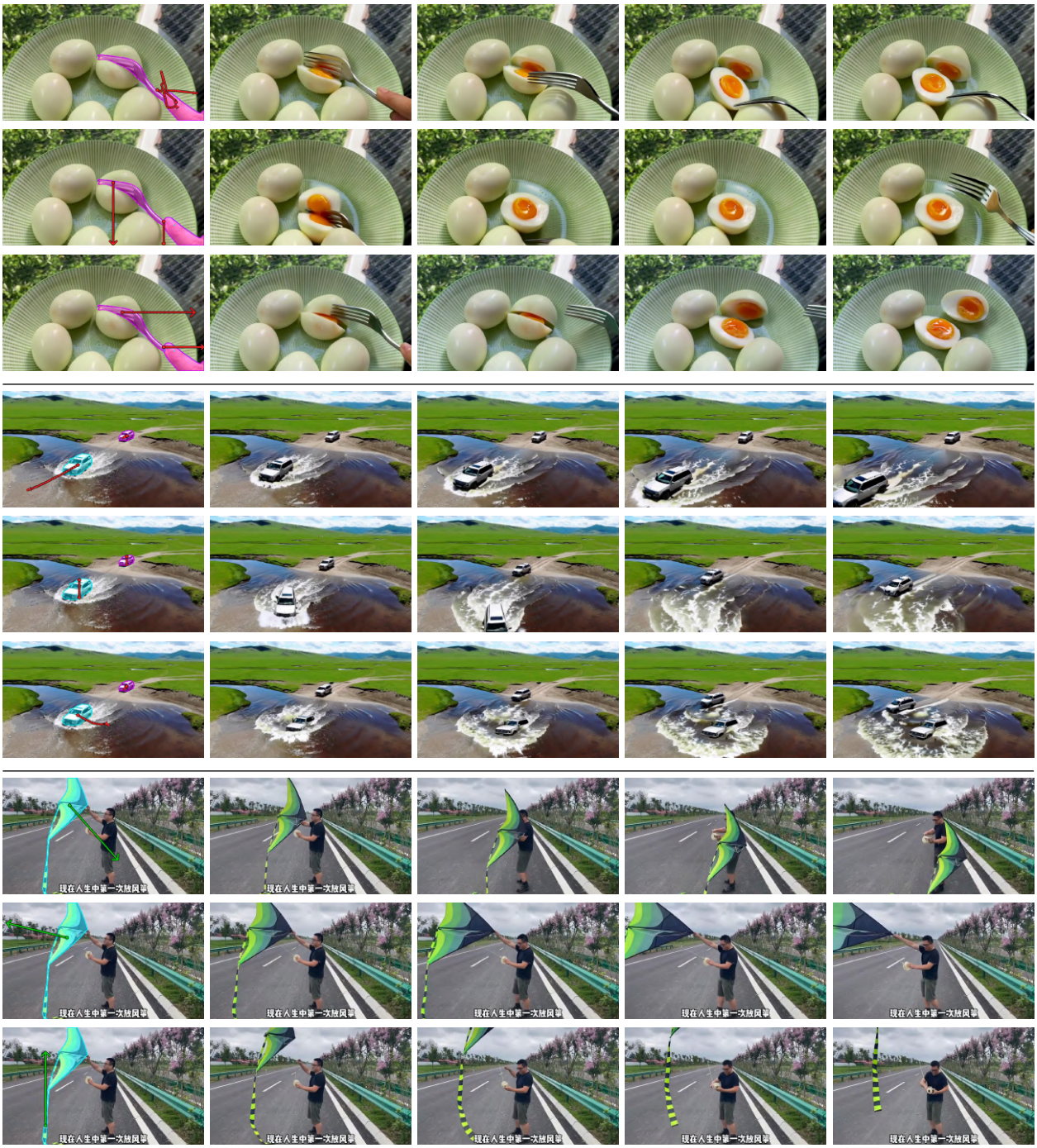

Figure 12: **Causal interaction reasoning.** Input tracks are shown in color and overlaid on the generated static reference-view video. The tracks represent user actions (active) or passive trajectories. Given these inputs, our model either predicts plausible consequences (forward reasoning) or recovers feasible driving actions that produce the desired outcomes (inverse reasoning, last row).

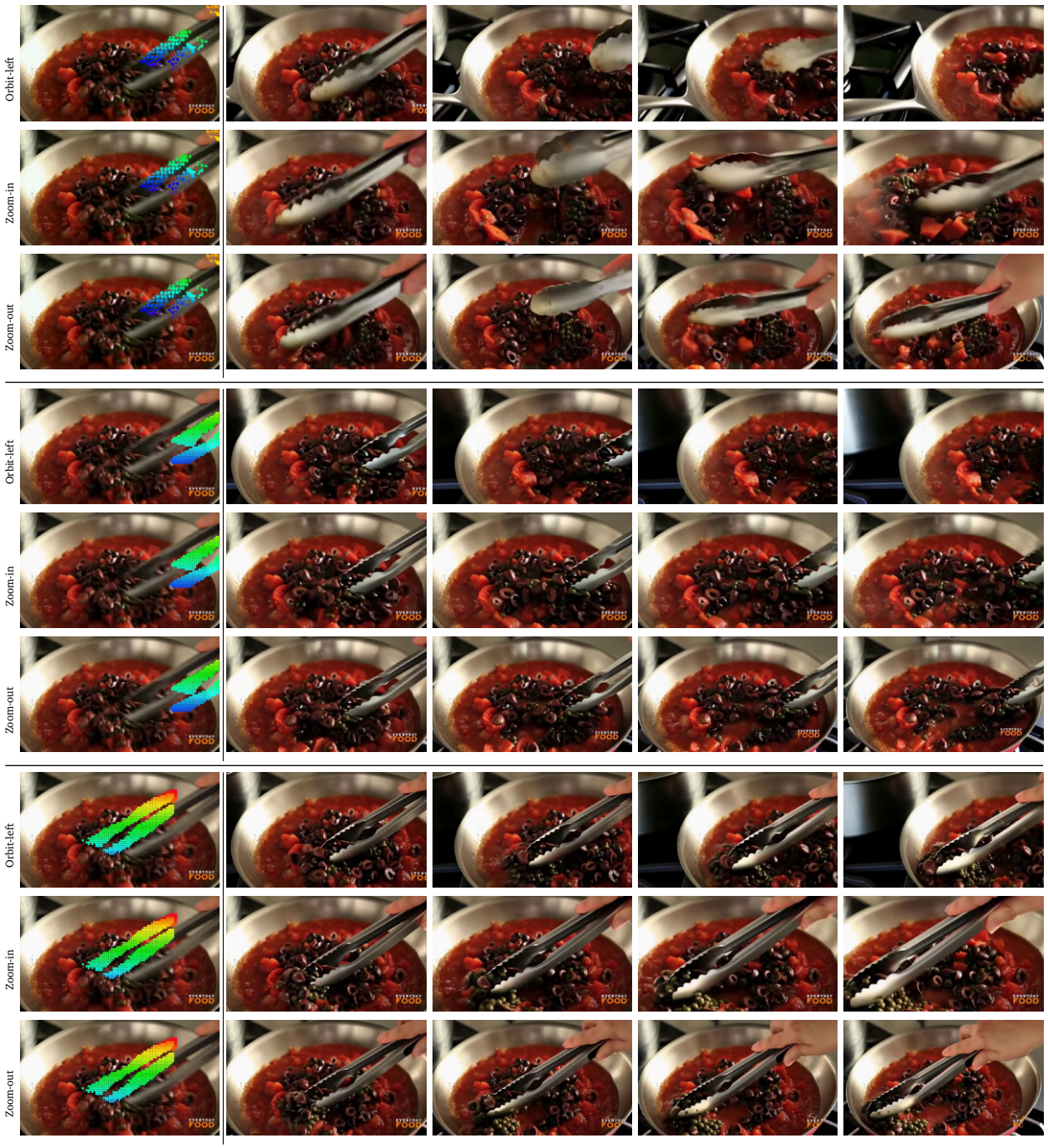

Figure 13: **Additional controllable generation-1.** Object motion trajectories are overlaid on the input image. For each video, we show different camera and object motion control. Each group shares the same object motion but uses different camera motions. Minor variations under the same object motion but different camera motions arise from the stochastic nature of interaction generation.

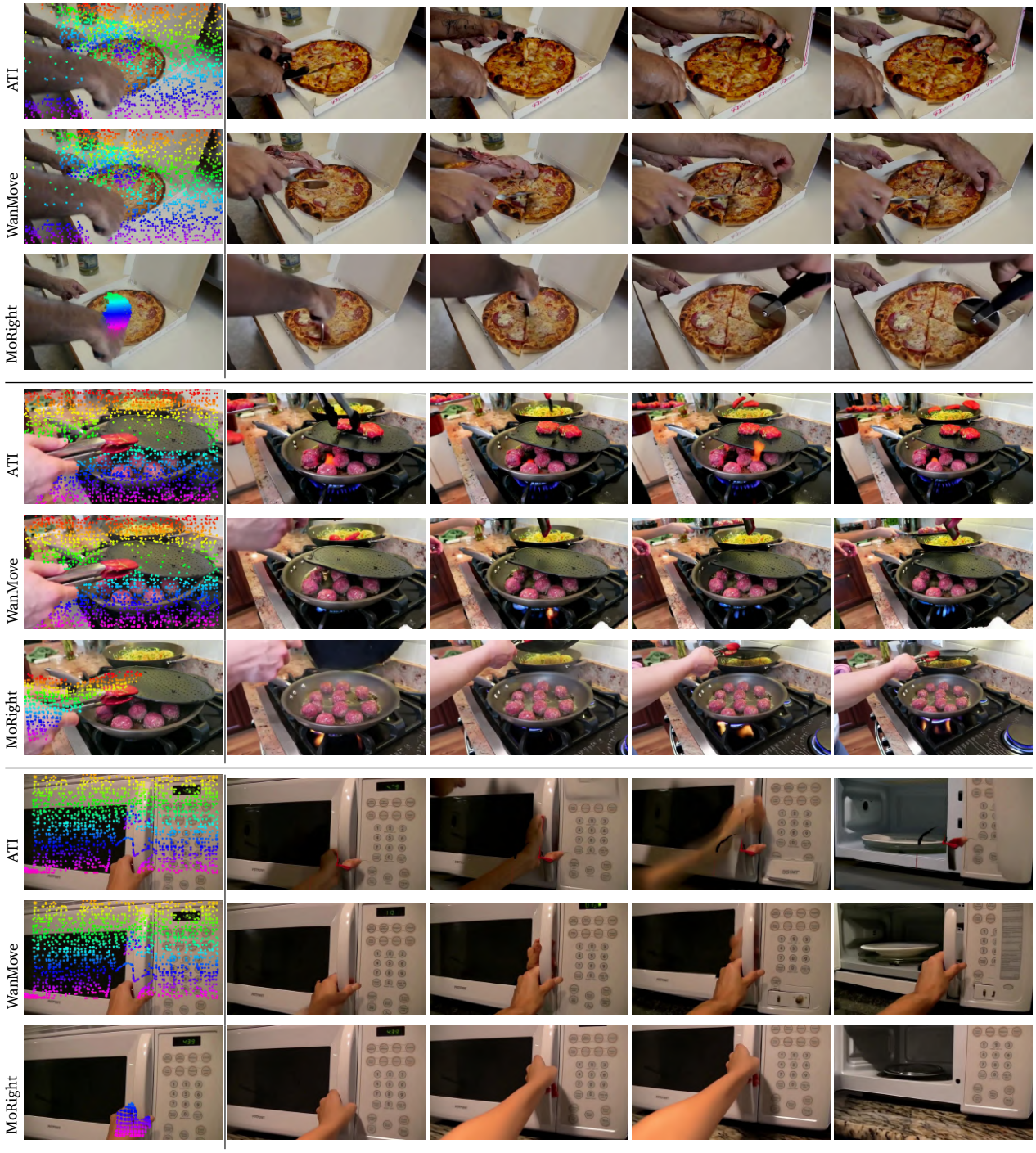

Figure 14: **Additional qualitative comparison** with ATI [48], WanMove [13], and MoRight. ATI and WanMove rely on privileged 3D trajectories (with depth) projected to pixel-aligned per-frame tracks and take full interaction trajectories (active and passive) as input. In contrast, MoRight uses only first-frame active tracks without privileged information and infers plausible interactions.

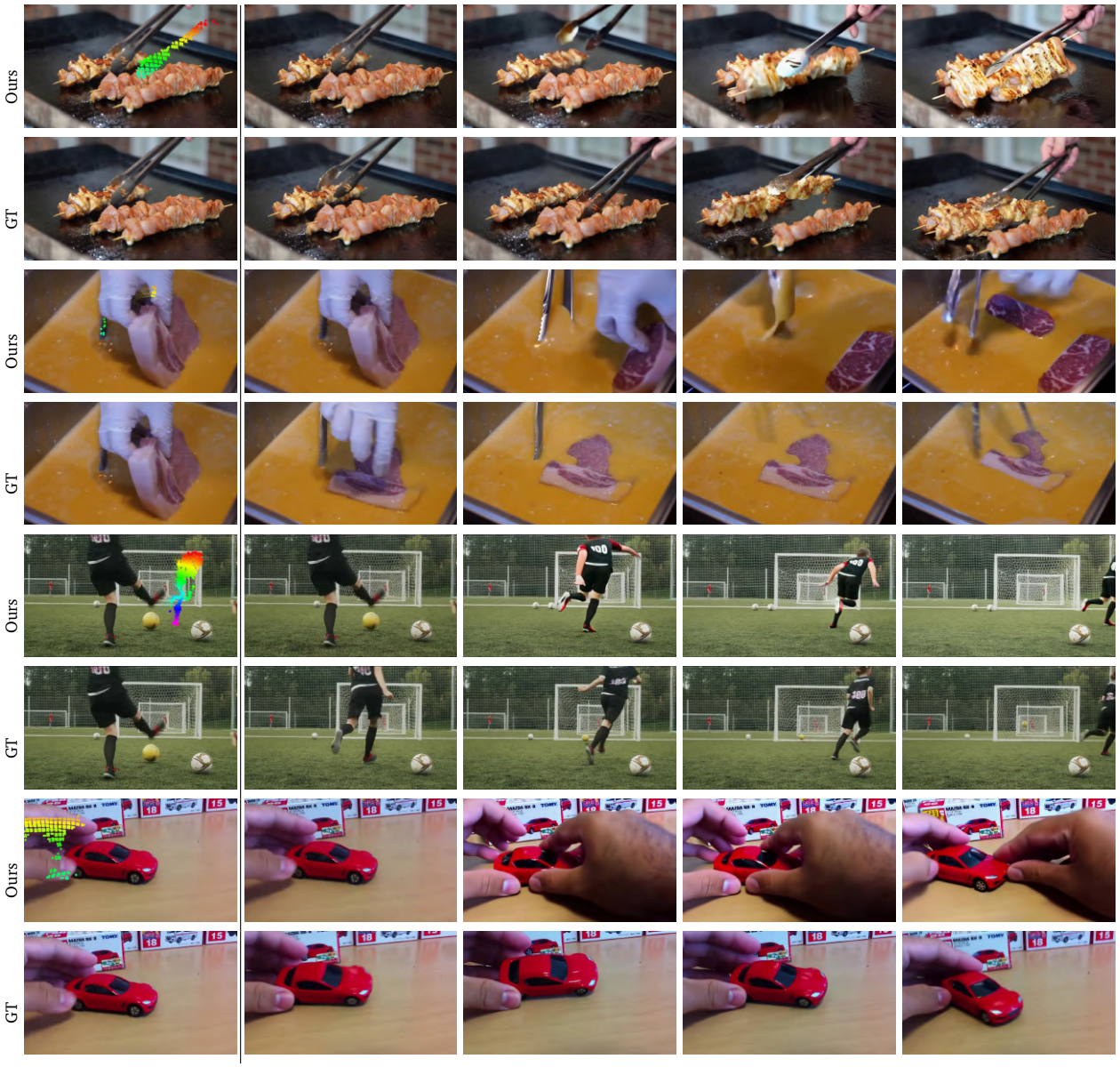

Figure 15: **Limitation analysis.** Input tracks are overlaid on the first frame as in previous figures. (1) Incorrect interaction reasoning may lead to implausible outcomes (two kabobs merging). (2) Unnatural motion can occur when input tracks become temporally sparse due to occlusion (hand example). (3) Physically unrealistic dynamics may appear, such as objects disappearing during motion (soccer ball). (4) Hallucinated content may emerge in later frames (extra hand).

object dynamics under different camera controls. Minor variations under the same object motion but different camera motions arise from the stochastic nature of interaction generation.

C.3. Qualitative Comparison

We provide additional qualitative comparisons with ATI [48] and WanMove [13] in Fig. 14. Both baselines rely on privileged 3D trajectories (with depth) projected to pixel-aligned per-frame tracks and take full interaction trajectories (active and passive) as input. In contrast, our method only requires 2D motion trajectories on the first frame, while camera motion is introduced in the second stream of our dual-stream architecture. Despite using weaker inputs, our approach achieves stronger controllability and produces more coherent interactions while maintaining disentangled camera–object motion.

D. Limitation Analysis

Despite promising results, our method still exhibits several limitations, as illustrated in Fig. 15. First, the model may produce incorrect interaction reasoning, leading to implausible outcomes such as two kabobs merging into a single object. Second, unnatural motion can occur when the input trajectories become temporally sparse due to occlusion, making it difficult for the model to reliably infer the intended motion (e.g., the hand example). Third, the generated motion may violate physical consistency, such as objects disappearing during interaction (soccer example). Fourth, the model may occasionally hallucinate new content in later frames, such as an extra hand appearing during generation.

In addition, our method has difficulty modeling very complex or fast camera motion (*e.g.*, drastic egomotion). Our camera control is designed for common smooth camera trajectories, and when the input camera motion changes drastically, the predicted interaction dynamics may degrade.