TrajLoc: Trajectory-Attention Localization for Multi-Object Motion Control

Omer Sela[1,2] **Inbar Huberman-Spiegelglas**[1]
Michael Rotman[1] **Sagie Benaim**[1,3] **Avi Ben-Cohen**[1]
[1]Amazon Prime Video [2]Tel Aviv University [3]Hebrew University of Jerusalem
sela-omer.github.io/traj-loc

Abstract

Controlling the motion of multiple objects in image-to-video (I2V) generation requires preserving object identities while enforcing adherence to distinct target trajectories. This becomes particularly challenging as the number of objects increases and their paths intersect or occlude one another. Existing approaches entangle multiple trajectories within a shared, dense conditioning signal, making object-level correspondence difficult to preserve in crowded scenes. We depart from this paradigm and enforce a strict, per object spatial constraint that isolates instances independently. Our method, TrajLoc, achieves this directly within the attention layers by substituting the cross-attention weights of each object token with a Gaussian heatmap centered on its target location at every frame. The same per object token interface carries trajectory and depth through a learned embedding and preserves identity by encoding first frame appearance in place of an object token. Evaluations across six datasets, featuring up to 20 simultaneously controlled objects and out of distribution real world scenes, demonstrate that our method consistently improves both visual fidelity and trajectory adherence. Applied to two architecturally distinct backbones (CogVideoX 5B and WaN 2.1 14B), our approach achieves average gains of +4.3 dB PSNR and a 51% reduction in trajectory end point error compared to the strongest baselines.

1 Introduction

Modern video diffusion models generate highly realistic videos [41, 29, 47], yet controlling the motion of *multiple* interacting objects, each following a distinct trajectory while preserving identity, remains a significant challenge. Generating videos with precise object control is essential for applications such as simulation and synthetic data generation for autonomous driving [14, 35, 12], robot learning [6, 21], and professional video editing [39, 34], where multiple objects must follow prescribed motion patterns while maintaining visual consistency. As the number of controlled objects grows and trajectories begin to interact, preserving object-level correspondence becomes increasingly difficult, often leading to identity confusion and motion drift.

While recent work has made progress on trajectory-conditioned generation [44, 4, 31, 22, 37, 36], existing methods struggle to scale to crowded scenes with many moving objects and are therefore typically evaluated on only one to three objects with limited interactions. This limitation is tied to how trajectory control is represented. The user input is inherently sparse: each object trajectory is specified by only a 2D coordinate per frame, requiring at most $2T$ scalar values for a video of T frames. Existing methods inflate this into dense tensors of size up to $T \times H \times W \times C$, whether through rasterized motion patches [44], latent feature propagation [4, 31], rendered trajectory videos [22], or dense conditioning volumes [11, 32]. Because multiple trajectories are fused within the same dense signal, object-level correspondence degrades under occlusions and crossings. The dense

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



Figure 1: **TrajLoc.** Given a first frame and a set of target trajectories (left column, with colored polylines), the goal is to generate a video that moves each object along its prescribed path while preserving its visual identity. Top: multiple pedestrians on a synthetic urban scene. Bottom: sheep in a natural outdoor scene. The remaining columns show three uniformly spaced generated frames with the ground-truth position of each object marked as a colored dot. In both scenes, our method places each object on its target dot, showing accurate trajectory adherence and consistent appearance throughout the video (best viewed in color).

representation further inherits the full spatiotemporal footprint of the video, requiring additional trainable modules such as ControlNet encoders [42] or FiLM layers [44, 46], and reducing portability across different diffusion backbones.

Our core insight is that sparse trajectory signals are sufficient to achieve precise multi-object control when incorporated directly into the model's existing conditioning spaces, without any dense control tensors or additional condition modules. To this end, we directly modify the cross-attention map so that each object's text token attends only to video locations along its prescribed trajectory and is attenuated everywhere else. Beyond spatial localization, the cross-attention heatmaps alone cannot convey depth ordering or visual identity. We address this by injecting learned per-object representations directly into the text conditioning space.

Specifically, TrajLoc replace the cross-attention weight map of each object's text token with Gaussian heatmaps centered on the target locations at every frame, providing a hard, per-object spatial constraint that scales naturally since each object is handled independently. We further enrich the same conditioning spaces with information beyond 2D position, each object is represented by two dedicated text tokens: a trajectory token encoding motion and depth, and an appearance token encoding first-frame visual identity in place of the generic category word. Our method applies to both separate cross-attention architectures (WaN) and joint self-attention designs (CogVideoX) without architecture-specific modifications.

We evaluate the approach on CogVideoX-5B-I2V and WaN 2.1-14B-I2V across six datasets spanning synthetic objects, billiard scenes, pedestrian crowds with up to 20 interacting objects, and real-world surveillance footage (MOT17) and diverse real-world scenes (DAVIS) outside the training distribution. Across all six datasets and both backbone families, our method sets a new state of the art, leading on both reconstruction quality (PSNR) and trajectory fidelity (end-point error) in every backbone group on every dataset. Average gains are $+4.3$ dB PSNR (0.7–10 dB range) and a 51% average reduction in end-point error (11–96% range) relative to the strongest same-backbone baseline per metric and dataset. In summary, our contributions are:

(i) We introduce attention localization, a mechanism that replaces cross-attention weight columns with trajectory-aligned Gaussian heatmaps, providing a hard per-object spatial constraint without dense conditioning tensors or additional modules.
(ii) We encode each object's trajectory and appearance as dedicated tokens in the text prompt, conveying depth and visual identity through the same conditioning space the attention mechanism already operates on.
(iii) Since each object's motion, identity, and spatial attention are handled independently, our method scales naturally to crowded scenes with many interacting objects and preserves per-object control under occlusions and trajectory crossings.

(iv) We demonstrate state-of-the-art results on six datasets with up to 20 simultaneous objects across two architecturally distinct backbones, trained exclusively on synthetic data yet generalizing to real-world scenes.

2 Related Work

Dense trajectory conditioning for video generation The dominant approach converts sparse point inputs into dense spatiotemporal tensors. Tora [44] rasterizes trajectories into motion patches via a 3D VAE, injected via adaptive-norm (FiLM-style) modulation. MotionCtrl [36] and MotionPro [46] construct dense displacement fields processed by convolutional modules. A parallel family uses ControlNet-style encoders: Motion Prompting [11] conditions on sinusoidal track volumes, LeviTor [32] on dense heatmap volumes with depth and instance channels, MagicMotion [19] on VAE-encoded trajectory maps, and PoseTraj [17] on rendered pose-aware trajectory maps with 3D bounding-box supervision. DragAnything [37] and DragEntity [30] inject entity representations via ControlNet-style encoders, while InTraGen [22] renders and VAE-encodes two full-resolution RGB videos. Wan-Move [4] and ATI [31] propagate first-frame features along trajectories within the existing I2V condition tensor, adding no new parameters but still producing a video-latent-sized signal. In all cases, multiple trajectories are fused within a shared dense signal, making object-level correspondence difficult to preserve under occlusions. Our method avoids this entirely by operating at the object-token level, where each object's motion and identity are encoded independently.

Token-based control Boximator [33] uses bounding-box sequences as input, injecting them as tokens via GLIGEN-style [20] self-attention layers, and relies on the model to learn box-to-object correspondence from data rather than enforcing it, a task that becomes harder as the number of overlapping objects grows. FlexTraj [45] concatenates VAE-encoded condition videos as tokens, though these are still derived from dense video-resolution maps. SG-I2V [24] is a zero-shot approach that enforces cross-frame feature similarity within bounding-box regions by replacing self-attention keys and values with first-frame features, requiring no training but limited to coarse bounding-box control. In contrast, our method provides fine-grained point-level trajectory control through learned per-object tokens and hard attention replacement, scaling to tens of simultaneous objects.

Cross-attention manipulation in diffusion models Editing attention maps for spatial control has been explored in the image domain. Prompt-to-Prompt [13] edits cross-attention for image editing, Attend-and-Excite [2] optimizes latents to strengthen cross-attention at subject positions, and Peekaboo [16] masks spatiotemporal attention for video region control. Training-free video methods follow a similar principle. TrailBlazer [23] scales cross-attention maps over bounding-box keyframes, FreeTraj [26] combines trajectory-shaped noise with attention guidance, and MotionZero [3] optimizes latents against box-alignment losses on cross-attention. For temporal rather than spatial control, Mind the Time [38] learns a time-aware positional encoding that biases cross-attention toward the event active in each segment of a text-to-video generation. All of these provide soft guidance signals. Our approach differs in that we replace attention weight columns with spatial distributions, providing a hard constraint applied at every layer during both training and inference.

3 Method

Given a first frame containing N objects and a target trajectory for each, we generate a video such that every object moves along its prescribed path while preserving its visual identity. Each object i is identified by a semantic category o_i (e.g., `ball`, `girl`) and a discretized trajectory $\tau_i(t) = (x_i(t), y_i(t), d_i(t))$ for $t = 1, \ldots, T$ frames, where $(x_i(t), y_i(t))$ denotes the center-of-mass location at frame t and $d_i(t)$ the depth relative to the screen plane under a fixed camera geometry.

We build upon a pretrained image-to-video (I2V) model conditioned on text and the first frame. In a standard I2V pipeline, video tokens attend to all text tokens without spatial bias, providing no explicit signal about *where* each referenced object should appear at each frame. To address this, we express trajectory control entirely within the existing conditioning spaces of the pretrained model, avoiding any external dense signals. We achieve this through two levels of intervention. Spatially, we substitute the learned cross-attention weights of each object token o_i with spatiotemporal Gaussian heatmaps derived from $\tau_i(t)$, imposing a hard per-object spatial constraint at every frame (Sec. 3.1). While this forces the model to attend to the correct regions, spatial heatmaps alone cannot convey

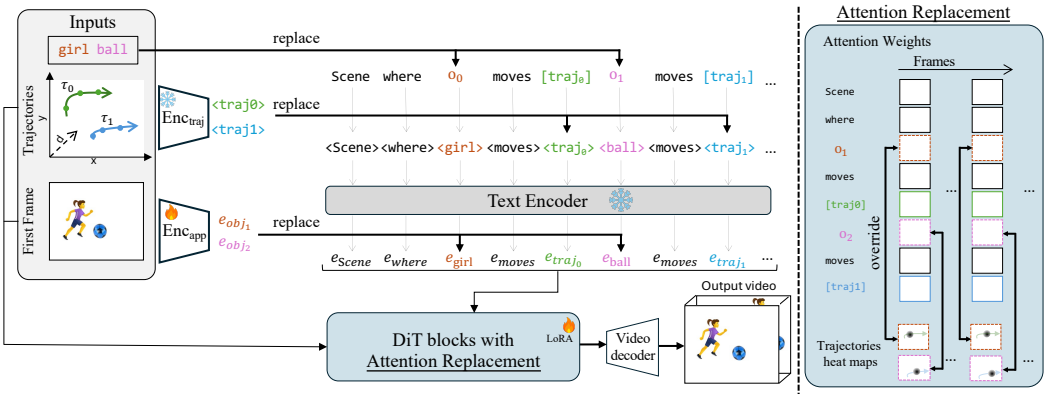

Figure 2: **An overview of TrajLoc.** A structured text prompt "Scene where o_0 moves [traj$_0$], o_1 moves [traj$_1$], ..." is constructed, where each o_i is the object's given category name. The trajectory tokens [traj$_i$] are replaced with learned embeddings from the pretrained (frozen) Enc_{traj}, which independently encodes each target trajectory $(x_i(t), y_i(t), d_i(t))$. The learned appearance encoder Enc_{app} encodes each object's appearance from the first frame into an embedding that replaces the corresponding object token. The resulting conditioning is fed to an I2V DiT model, where attention localization is applied within each transformer block (right). **Right:** The standard cross-attention weights at each object category o_i are overridden by trajectory-aligned Gaussian heatmaps, centered at the target coordinates $(x_i(t), y_i(t))$ at every frame.

depth or visual identity. Semantically, we address this gap by modifying the text conditioning space. We organize the inputs into a structured textual instruction such that each object i is described with its semantic category o_i, a verb, and a trajectory token,

$$\texttt{"Scene where } o_0 \texttt{ moves [traj}_0\texttt{] and } o_1 \texttt{ moves [traj}_1\texttt{] ..."} \tag{1}$$

To capture motion and depth $d_i(t)$ we replace the placeholder tokens [traj$_i$] with trajectory embeddings (Sec. 3.2.1). To preserve the appearance of each specific instance in the input image we replace the semantic category o_i embedding with visual features extracted from the first frame (Sec. 3.2.2). Both encoders are learned through the training procedure detailed in Sec. 3.3. Unlike dense conditioning approaches, our representation scales with the number of objects N rather than the video resolution. The complete pipeline is illustrated in Figure 2.

3.1 Attention Localization

The trajectory $\tau_i(t)$ of each object is used to steer the model's attention toward specific spatiotemporal locations. This is achieved by substituting the cross-attention weight column at the object token index with a spatiotemporal Gaussian heatmap centered on the target trajectory across all attention layers.

Trajectory heatmap construction For each object i we construct a spatiotemporal trajectory heatmap by placing a two dimensional isotropic Gaussian with standard deviation σ centered at its trajectory location $(x_i(t), y_i(t))$ for every frame t in which the object is visible. The resulting heatmap sequence is bilinearly resampled onto the latent spatiotemporal grid and normalized per frame to sum to one while frames in which the object is absent remain zero. Stacking these maps across all frames and vectorizing the result yields the per object heatmap $\mathbf{h}_i \in \mathbb{R}^M$ where M is the total number of spatiotemporal latent tokens. As visualized in the right panel of Figure 2 these Gaussian maps are created for every trajectory τ_i at each frame. Algorithm 1 (Appendix I) details this construction and the replacement below.

Column replacement We denote the cross-attention weight matrix $\mathbf{A}$ and the final attention output $\mathbf{O}$ as:

$$\mathbf{A} = \text{softmax}\left(d^{-1/2}\,\mathbf{Q}_{\text{vid}}\mathbf{K}_{\text{txt}}^{\top}\right), \qquad \mathbf{O} = \mathbf{A}\mathbf{V}_{\text{txt}} \tag{2}$$

where $\mathbf{A} \in \mathbb{R}^{M \times L}$ represents the alignment between the M visual patches and L text tokens. We modify this process by identifying the specific column index q_i in $\mathbf{A}$ that corresponds to the object category o_i, i.e., the position of the token o_i in the tokenized prompt of Eq. 1, and substituting its

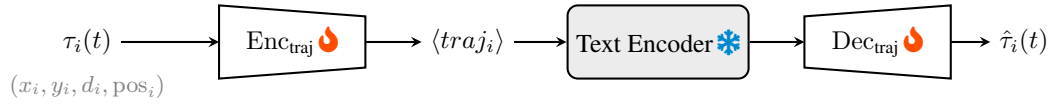

Figure 3: **Trajectory autoencoder pretraining.** The trajectory encoder maps each object trajectory $\tau_i(t) = (x_i(t), y_i(t), d_i(t))$ and a temporal position channel to a token embedding $\langle \text{traj}_i \rangle$ in the text encoder space. The embedding passes through the frozen text encoder before a decoder reconstructs the original trajectory, ensuring the representation remains informative after text encoder processing.

original weights with the spatiotemporal Gaussian heatmap $\mathbf{h}_i$. To maintain the probabilistic nature of the attention mechanism we then renormalize each row of the modified matrix $\mathbf{A}'$ such that:

$$\mathbf{A}'[m, q_i] \leftarrow \mathbf{h}_i[m], \qquad \mathbf{A}'[m, :] \leftarrow \mathbf{A}'[m, :] \,/\, \sum_{k=1}^{L} \mathbf{A}'[m, k] \tag{3}$$

where $m \in \{1, \ldots, M\}$ indexes video-token positions. The final attention output is then computed as $\mathbf{O}' = \mathbf{A}'\mathbf{V}_{\text{txt}}$. This replacement ensures that the influence of the object token is strictly localized to the video regions defined by the prescribed trajectory while leaving other text tokens and their respective spatial alignments intact. As illustrated in Figure 2, this mechanism overrides the learned attention weights with Gaussian heatmaps at every object token position, steering the generation toward the prescribed trajectories.

The replacement adds no new trainable parameters and no new dense conditioning tensors as it operates on values the attention mechanism already computes. The formulation above is exact for architectures with explicit cross-attention (e.g., WaN 2.1), but infeasible for joint self-attention architectures such as CogVideoX, whose full attention matrix (approximately 30 GB per layer) exceeds GPU memory. For those architectures we derive an efficient approximation that preserves the replacement semantics, as described in Appendix D.

3.2 Token Level Conditioning

While attention localization forces the model to look at the correct regions, it remains agnostic to depth and visual identity. We resolve this by modifying the text conditioning embeddings of the structured instruction described in Eq. 1. The prompt is passed through a tokenizer and then a frozen text encoder to produce the initial conditioning embeddings, as illustrated in Figure 2. We apply two targeted replacements: the trajectory placeholder [traj_i] is replaced with a learned trajectory embedding (Sec. 3.2.1) and the semantic category o_i is replaced with a learned visual encoding from the first frame (Sec. 3.2.2).

3.2.1 Trajectory Encoding

The trajectory encoder Enc_{traj} maps each trajectory τ_i to a single token embedding $\langle \text{traj}_i \rangle \in \mathbb{R}^{4096}$ that replaces the corresponding placeholder in the text prompt. Each trajectory is represented as a $T \times 4$ sequence of per-frame coordinates $(x_i(t), y_i(t), d_i(t))$ and a temporal position channel. A lightweight convolutional encoder maps this sequence to $\langle \text{traj}_i \rangle$, matching the text encoder's token dimension. A key requirement is that the trajectory information survives the frozen text encoder's self-attention layers. We therefore pre-train Enc_{traj} jointly with a decoder that reconstructs τ_i from the text encoder's output at the corresponding token position (Figure 3). The reconstruction loss is

$$\mathcal{L}_{\text{traj}} = \text{MSE}(\hat{\tau}_i, \tau_i) + \lambda_{\text{vel}} \,\text{MSE}(\Delta\hat{\tau}_i, \Delta\tau_i), \tag{4}$$

where $\hat{\tau}_i$ is the reconstructed trajectory and $\Delta\tau_i \equiv \tau_i^t - \tau_i^{t-1}$ penalizes velocity errors. After pretraining, the decoder is discarded and Enc_{traj} is frozen. Architectural details are provided in Appendix E.1.

3.2.2 Appearance Encoding

A semantic category label such as `girl` in the text prompt does not distinguish between specific instances. To preserve each object's appearance as seen in the first frame, we replace the text token embedding of each category o_i with a visual encoding extracted from the input image.

The appearance encoder $\mathrm{Enc}_{\text{app}}$ takes the first-frame VAE latent and extracts a feature at each object's initial position (x_i^1, y_i^1), the first entry of its prescribed trajectory τ_i, using a lightweight convolutional network, then projects it into the text embedding space to produce e_{o_i}. This embedding replaces the corresponding object token representation after the text encoder, as illustrated in Figure 2. The output is normalized to match the statistics of the text encoder output space. Architectural details are provided in Appendix E.2.

3.3 Training

We first pre-train the trajectory encoder using the reconstruction objective of Eq. 4, which requires only trajectories and the first frame. We then fine-tune a pretrained I2V diffusion backbone using LoRA [15], jointly training the appearance encoder while keeping the trajectory encoder frozen. The diffusion training uses full video clips and combines the standard flow-matching loss with a trajectory-focused reconstruction term. Specifically, let $\mathcal{L}_{\text{diff}}$ denote the mean squared error of the noise prediction over all spatial positions, and $\mathcal{L}_{\text{bbox}}$ the corresponding error restricted to bounding boxes centered along each trajectory. The combined loss is:

$$\mathcal{L} = (1-\alpha)\mathcal{L}_{\text{diff}} + \alpha\mathcal{L}_{\text{bbox}}, \tag{5}$$

with α balancing accurate object placement against global scene coherence. During training, attention columns are overwritten using heatmaps generated from the ground-truth trajectories, so the model learns to generate videos *given* the forced attention pattern. At inference, the model needs only the first frame plus a category name and a trajectory for each object, with depth optional (Sec. 4.5).

4 Experiments

We evaluate TrajLoc on two architecturally distinct I2V backbones across six datasets, comparing against four state-of-the-art trajectory-conditioned baselines. We first describe the experimental setup (Sec. 4.1), then report quantitative results (Sec. 4.2), scaling with object count and trajectory crossings (Sec. 4.3), qualitative comparisons (Sec. 4.4), and ablations (Sec. 4.5).

4.1 Experimental setup

Implementation details We apply our method to two I2V backbones: WaN 2.1-14B, where attention localization is applied across all 40 transformer layers, and CogVideoX-5B, where we use our approximation (Appendix D.2) across all 42 layers. In both cases we adapt the attention layers with LoRA (rank 64, α_{LoRA}=64) and train Eq. 5 with bounding-box loss weight α=0.5 jointly with the appearance encoder while all other parameters are frozen. Trajectory heatmaps use σ=30 pixels at video resolution. Additional implementation details, including diffusion fine-tuning hyperparameters and training budget, are provided in Appendix E.3. Code and our real-data checkpoints (Appendix A) are available at `https://github.com/Sela-Omer/traj-loc`.

Training data We train on ∼25K static-camera clips drawn from four publicly available synthetic datasets, as no real-world dataset provides static-camera multi-object trajectories at the required scale. The sources are MOTSynth [8] (GTA V urban scenes with up to 20 pedestrians), and three subsets from the ViN suite [22]: MoVi-Extended (Kubric-rendered scenes with 2–12 geometric objects), Pool (overhead billiard scenes with 2–3 balls), and Football (synthetic soccer scenes with up to 2 players and a ball). All datasets provide ground-truth trajectories. Depth is derived from world coordinates for MOTSynth and estimated with Depth Anything V2 [40] for ViN. All clips are 49 frames at 720×480 with a static camera. A detailed breakdown of the training data is provided in Appendix F.1.

Evaluation data We evaluate on 446 held-out samples across six datasets. Four are in-distribution: 100 clips each from the MoVi-Extended, Pool, and Football ViN test splits, and 100 MOTSynth clips from 65 held-out scenes. Two are out-of-distribution (OOD): 33 real-world pedestrian clips from MOT17 [5], filtered to static-camera sequences, and 13 scenes from DAVIS 2017 [25] covering animals, vehicles, sports, and other categories absent from training, filtered for camera motion as detailed in Appendix F.3. All evaluation clips use a static camera and match the training resolution and frame count. Test samples are filtered so that every tracked object is visible in the first frame, giving every method a well-defined first-frame appearance input.

Table 1: **Evaluation on three synthetic ViN datasets**. 100 samples per dataset. Each baseline is evaluated on its native backbone. Our method leads on all metrics in both backbone groups. Best within each group in bold.

		MoVi-Extended				Pool				Football			
	Method	PSNR↑	LPIPS↓	FVD↓	EPE↓	PSNR↑	LPIPS↓	FVD↓	EPE↓	PSNR↑	LPIPS↓	FVD↓	EPE↓
Wan 14B	**Ours**	**29.24**	**.131**	**23.39**	**0.57**	**33.07**	**.049**	**31.34**	**0.08**	**24.27**	**.141**	**24.24**	**0.05**
	ATI	20.33	.355	37.50	2.01	21.13	.303	57.78	0.37	19.68	.250	46.32	1.33
	Wan-Move	19.73	.331	40.38	2.18	22.94	.191	33.98	0.29	18.36	.296	69.84	3.81
CogV	**Ours**	**29.77**	**.108**	**14.98**	**0.53**	**35.64**	**.025**	**8.79**	**0.07**	**24.79**	**.090**	**13.87**	**0.03**
	MagicMotion	19.79	.293	44.67	1.92	29.52	.068	18.93	0.13	22.28	.161	50.56	0.33
	Tora	20.87	.336	50.91	1.50	31.10	.128	50.39	0.12	19.34	.280	63.91	2.87

Table 2: **Evaluation on MOTSynth and OOD datasets.** MOTSynth: 100 synthetic urban scenes. MOT17: 33 real-world surveillance clips. DAVIS: 13 diverse real-world scenes. MOT17 and DAVIS are never seen during training. All baselines are trained on real-world video data, while our method is trained on synthetic data only. Best within each backbone group in bold.

		MOTSynth				MOT17				DAVIS			
	Method	PSNR↑	LPIPS↓	FVD↓	EPE↓	PSNR↑	LPIPS↓	FVD↓	EPE↓	PSNR↑	LPIPS↓	FVD↓	EPE↓
Wan 14B	**Ours**	**21.39**	**.244**	**41.43**	**2.11**	**19.94**	**.277**	**51.46**	**2.17**	**18.30**	**.301**	88.37	**2.31**
	ATI	17.76	.395	62.32	8.61	17.05	.416	65.35	4.28	14.68	.443	104.90	14.02
	Wan-Move	18.29	.354	54.29	3.60	16.27	.405	60.20	3.41	15.86	.352	**85.51**	4.96
CogV	**Ours**	**22.34**	**.232**	**39.70**	**2.32**	**21.59**	**.265**	**53.92**	**2.37**	**18.39**	**.267**	89.39	**2.43**
	MagicMotion	21.17	.258	43.07	2.61	19.88	.307	56.13	3.38	17.74	.294	**70.77**	3.68
	Tora	19.20	.346	70.84	2.91	17.76	.400	113.9	2.70	17.25	.329	75.78	3.16

Baselines We compare against four recent state-of-the-art trajectory-conditioned methods, each of which turns the sparse trajectory input into a dense video-sized conditioning signal. On the Wan2.1-14B backbone, **Wan-Move** [4] populates the I2V condition latent with first-frame VAE features at trajectory positions, and **ATI** [31] does the same with Gaussian-weighted blending across each trajectory. On the CogVideoX-5B backbone, **Tora** [44] rasterizes trajectories into motion patches via a 3D VAE and convolutional encoder, and **MagicMotion** [19] renders bounding-box trajectory maps and processes them through a ControlNet copy of the backbone. All baselines use their released default inference configurations and per-video captions (Appendix F.2, F.4), providing richer scene descriptions than our template-based prompts. For ATI and Wan-Move, whose released models output 81 frames, we patch the frame count to 49 to match our evaluation protocol. Appendix F.5 confirms this patch does not disadvantage them relative to their native 81-frame output.

Evaluation metrics We report four complementary metrics. PSNR measures pixel-level reconstruction quality. LPIPS [43] captures perceptual similarity. FVD [28] evaluates distributional quality of generated video. To assess motion fidelity, we measure the L2 distance between the ground-truth tracks and the tracks estimated from the generated videos, denoting this metric as end-point error (EPE), following Wan-Move [4] and Motion Prompting [11].

4.2 Quantitative evaluation

Table 1 reports results on the three synthetic ViN datasets. Table 2 reports results on MOTSynth, MOT17, and DAVIS, where MOT17 and DAVIS are never seen during training. Our method leads on PSNR, LPIPS, and EPE across both backbones on all datasets, with PSNR gains of 4–10 dB on Pool and Football and 1–3 dB on the crowded multi-object settings (MOTSynth, MOT17). The improvements are particularly pronounced in trajectory adherence, with EPE reductions ranging from 11% to over 10× depending on the dataset and baseline. The same gains persist on MOT17 and DAVIS, which are never seen during training, confirming that our method generalizes beyond the synthetic training distribution. This is particularly notable given that all baselines are trained on large-scale real-world video data, while our method is trained exclusively on synthetic scenes. We acknowledge that the baselines were not trained on our synthetic evaluation datasets and could benefit from fine-tuning on them. However, on MOT17 and DAVIS, which are real-world scenes closer to the baselines' training distribution, our method still leads, demonstrating generalization beyond our

Table 3: **Scaling with object count on MOTSynth-Multi.** 20 held-out clips per bin with 5, 10, 15, and 17–20 controlled pedestrians. Best within each backbone group in bold.

		5 pedestrians				10 pedestrians				15 pedestrians				17–20 pedestrians			
	Method	PSNR	LPIPS	FVD	EPE	PSNR	LPIPS	FVD	EPE	PSNR	LPIPS	FVD	EPE	PSNR	LPIPS	FVD	EPE
Wan 14B	**Ours**	**20.40**	**.293**	**56.65**	**4.39**	**20.46**	**.286**	**59.45**	**2.63**	**20.66**	**.257**	**62.14**	**2.08**	**21.10**	**.240**	**48.47**	**0.99**
	ATI	16.77	.440	90.35	9.15	17.45	.418	77.19	4.93	18.38	.358	70.64	3.15	19.31	.304	57.11	1.79
	Wan-Move	17.27	.423	78.86	6.82	16.92	.426	83.83	4.69	18.22	.355	77.42	2.98	19.22	.300	54.65	1.67
CogV	**Ours**	**21.12**	**.290**	**54.04**	5.51	**21.26**	**.268**	**49.98**	**2.75**	**21.67**	**.240**	**54.41**	**1.92**	**21.62**	**.237**	49.04	**1.11**
	MagicMotion	20.03	.315	68.65	**5.30**	20.93	.285	67.13	3.08	21.17	.250	55.95	2.33	21.40	**.237**	**42.61**	1.24
	Tora	18.79	.386	104.5	5.81	18.00	.417	127.4	3.55	16.47	.463	152.4	2.79	17.66	.454	162.9	1.64

Table 4: **Trajectory crossings on MoVi-Extended.** Test clips binned by the number of object pairs whose paths cross in the image plane. Best within each backbone group in bold.

		0 crossings (56 clips)				1 crossing (28 clips)				2+ crossings (16 clips)			
	Method	PSNR↑	LPIPS↓	FVD↓	EPE↓	PSNR↑	LPIPS↓	FVD↓	EPE↓	PSNR↑	LPIPS↓	FVD↓	EPE↓
Wan 14B	**Ours**	**30.26**	**.120**	**25.53**	**0.48**	**29.30**	**.132**	**33.54**	**0.58**	**25.56**	**.169**	**35.57**	**0.90**
	ATI	20.76	.344	46.09	1.86	20.61	.336	54.56	1.91	18.33	.425	67.56	2.72
	Wan-Move	20.12	.316	49.48	1.79	19.88	.327	56.36	2.60	18.08	.390	65.75	2.84
CogV	**Ours**	**30.61**	**.101**	**17.83**	**0.46**	**29.62**	**.107**	**23.09**	**0.51**	**27.13**	**.131**	**22.88**	**0.84**
	MagicMotion	20.43	.272	53.35	1.63	19.99	.290	61.71	1.91	17.18	.374	71.23	2.97
	Tora	21.73	.315	54.88	1.42	20.07	.356	84.36	1.33	19.26	.374	77.62	2.07

synthetic training data. Appendix A complements this by training our method on real moving-camera video. On MagicBench [19] and MoveBench-Multi [4], it also leads in PSNR, LPIPS, and FVD.

4.3 Object count and trajectory crossings

MOTSynth-Multi To measure how performance changes as scenes become crowded, we construct MOTSynth-Multi from held-out MOTSynth test scenes: four bins of 20 clips with 5, 10, 15, and 17–20 controlled pedestrians (Appendix G). As shown in Table 3, our method with WaN leads on every metric at every object count. With CogVideoX, it leads on PSNR and LPIPS at every count (tied on LPIPS in the densest bin) and on FVD and EPE in three of the four bins.

Trajectory crossings We also bin the MoVi-Extended test clips by the number of object pairs whose paths cross (Table 4). All methods degrade as crossings increase, yet ours leads on every metric in every bin and, even with two or more crossings, beats every baseline on crossing-free clips.

4.4 Qualitative Evaluation

To qualitatively assess our method, we compare against all baselines on representative scenes in Figure 4. Each row shows three generated frames spanning the video (frames 9, 29, 49) where colored dots mark the ground-truth position of each object at that frame. The top row shows the original video with prescribed trajectory paths overlaid.

In the top row of panels, CogVideoX-5B is compared against MagicMotion and Tora on the real-scene DAVIS fish scene (5 fish). MagicMotion fails to place objects at several target locations, for example the orange trajectory dot has no corresponding fish in the generated frame. Tora preserves all five entities but places them at incorrect positions, such as the large fish at the top right of the frame. Our method maintains all five fish at their target locations with consistent identity throughout the sequence. WaN 2.1-14B is compared against ATI and Wan-Move on the real-scene MOT17 (4 pedestrians). ATI loses tracking on the dark-blue-trajectory pedestrian. Wan-Move makes the light-blue-trajectory and orange-trajectory pedestrians hover unnaturally and hallucinates additional details. Our method keeps all four pedestrians aligned with their prescribed trajectories. The bottom-left panel compares CogVideoX-5B against MagicMotion and Tora on a MoVi-Extended scene (6 geometric objects). Both baselines change object appearance and lose tracking on several trajectories, while our method accurately follows all six prescribed paths. The bottom-right panel shows WaN 2.1-14B against ATI and Wan-Move on a MOTSynth scene (10 pedestrians). ATI fails to follow individual trajectories such as the blue one on the right, which stops tracking its prescribed path, and Wan-Move duplicates

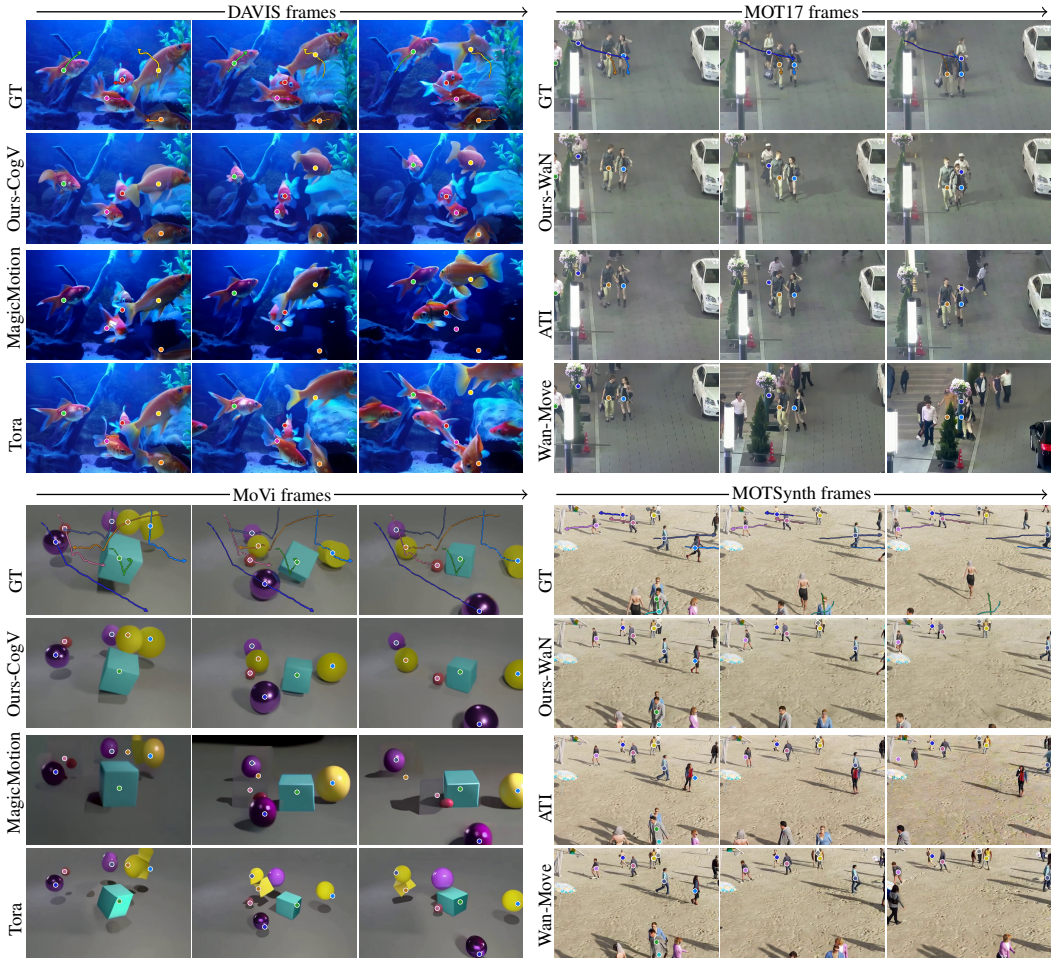

Figure 4: **Qualitative comparison.** Top-left: CogVideoX-5B results on DAVIS fish (5 objects, real-world). Top-right: WaN 2.1-14B results on MOT17 (4 pedestrians, real-world). Bottom-left: CogVideoX-5B results on MoVi-Extended (6 objects, synthetic). Bottom-right: WaN 2.1-14B results on MOTSynth (10 pedestrians, synthetic). Each row shows three generated frames from a different method, with ground-truth object positions overlaid as colored dots. Our method places all objects at their target locations throughout the sequence, while baselines exhibit missing objects, positional drift, or hallucinated entities. Best viewed in color.

the gray-trajectory pedestrian at the top of the final frame. Our method keeps every pedestrian aligned with its prescribed trajectory. Additional qualitative comparisons on MoVi, MOTSynth, MOT17, and DAVIS are provided in Appendix B. Full video results, which better convey object motion along the prescribed trajectories and identity preservation across frames, are available on our project page (`https://sela-omer.github.io/traj-loc/`).

4.5 Ablation study

We examine the contribution of each component on WaN 2.1-14B (Table 5), both by removing one component at a time and by building the model up from the bare backbone, on MoVi and MOTSynth, the in-distribution datasets with the most objects per scene, so that the measured effect of each component reflects the component itself rather than domain shift. Attention localization alone recovers most of the gap between the backbone and the full model (row 2: +3.8 dB PSNR on MoVi and +1.7 dB on MOTSynth), whereas the tokens without it barely change the backbone (rows 4–5). The tokens carry per-object motion, depth, and appearance, but only attention localization binds each token to its object in the video; without it, they act as diffuse, global conditioning. Removing the attention localization from the full model causes the largest degradation across all metrics, with PSNR dropping by 6.7 dB and EPE increasing over 3$\times$ on MoVi, confirming that the hard spatial constraint is the most critical component. Removing the trajectory tokens primarily affects

Table 5: **Ablation study on WaN 2.1-14B** with additive and leave-one-out configurations. A/T/P/d: attention localization, trajectory token, appearance token, depth. Leave-one-out PSNR varies by at most 0.10 dB over nine seeds (Appendix H.3); Pool and Football are in Appendix H.1. Best in bold.

						MoVi				MOTSynth			
#	Configuration	A	T	P	d	PSNR↑	LPIPS↓	FVD↓	EPE↓	PSNR↑	LPIPS↓	FVD↓	EPE↓
1	Backbone only					22.95	.284	34.27	1.46	19.02	.337	62.60	4.05
2	Attention only	✓				26.79	.159	28.63	0.74	20.69	.266	46.44	2.26
3	w/o trajectory token	✓		✓		27.87	.146	27.15	0.65	21.30	.256	44.01	2.68
4	Trajectory token only		✓		✓	23.53	.272	37.03	1.32	19.35	.314	53.79	3.34
5	w/o attention localization		✓	✓	✓	22.58	.313	40.86	1.87	19.31	.316	55.96	3.22
6	w/o appearance token	✓	✓		✓	28.40	.146	28.09	0.67	20.78	.258	44.95	2.21
7	w/o depth channel	✓	✓	✓		27.56	.145	27.66	0.62	21.32	.250	43.11	2.28
8	**Ours**	✓	✓	✓	✓	**29.24**	**.131**	**23.39**	**0.57**	**21.39**	**.244**	**41.43**	**2.11**

depth-sensitive scenes, with EPE on MOTSynth increasing from 2.11 to 2.68. Notably, removing the depth channel alone produces degradation comparable to removing the full trajectory token on MoVi, suggesting that depth encoding is a key contribution of the token. On Pool, however, the full token provides substantially more than depth alone (PSNR drops by 4 dB vs. less than 1 dB; Appendix H.1), indicating it also captures spatial motion information. Removing the appearance token costs 0.6 dB PSNR on MOTSynth and increases FVD on both datasets. The full model achieves the best result on every metric, demonstrating that all components are complementary. The bounding-box loss and inserting the trajectory token before the text encoder both matter (Appendix H.2), while depth is optional (Appendix H.4). An ablation of the attention localization approximation used for CogVideoX-5B is provided in Appendix D.5.

5 Limitations and Broader Impacts

Our main experiments use static-camera scenes at 720×480 resolution, and the training data is predominantly synthetic, which introduces a domain gap on natural scenes. While the method generalizes to real-world data such as DAVIS and MOT17, we occasionally observe the model falling back on synthetic appearance priors, e.g., generating GTA-V pedestrian sprites in place of out-of-distribution objects. Failure cases are shown in Appendix C. Our real-data models (Appendix A) handle camera motion only implicitly, through background trajectories. Explicit camera control, higher resolutions, and finer-grained, 3D-aware motion control remain future work. As with other generative models, our method carries dual-use potential. Its ability to produce realistic, controllable videos can benefit synthetic data generation, simulation, and professional video editing, but also risks misuse for generating misleading content. We believe it is crucial to develop detection tools for AI-generated videos and to adopt safeguards such as watermarking for safer generative AI usage.

6 Conclusion

We presented TrajLoc, a method for multi-object trajectory-guided video generation that challenges the prevailing assumption that trajectory conditioning requires dense video-sized representations. Instead, we observe that the model's existing conditioning spaces, namely the attention mechanism and the text tokens representations, are sufficient to carry per-object trajectory and identity control. Accordingly, we enforce spatial adherence by replacing attention weight columns with Gaussian heatmaps at the object's associated token and encode each object's trajectory and appearance as dedicated tokens in the text prompt. Evaluated on six datasets spanning synthetic objects, billiard scenes, pedestrian crowds with up to 20 interacting objects, and diverse real-world scenes never seen during training, our method leads on PSNR, LPIPS, and EPE within each backbone group across all six datasets.

Acknowledgments and Disclosure of Funding

This work was funded by Amazon. The authors declare no competing interests.

References

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

[2] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. In *SIGGRAPH*, 2023.

[3] Changgu Chen, Junwei Shu, Gaoqi He, Changbo Wang, and Yang Li. Motion-zero: Zero-shot moving object control framework for diffusion-based video generation. In *AAAI*, 2025.

[4] Ruihang Chu, Yefei He, Zhekai Chen, Shiwei Zhang, Xiaogang Xu, Bin Xia, Dingdong Wang, Hongwei Yi, Xihui Liu, Hengshuang Zhao, Yu Liu, Yingya Zhang, and Yujiu Yang. Wan-move: Motion-controllable video generation via latent trajectory guidance. In *NeurIPS*, 2025.

[5] Patrick Dendorfer, Aljoša Ošep, Anton Milan, Konrad Schindler, Daniel Cremers, Ian Reid, Stefan Roth, and Laura Leal-Taixé. Motchallenge: A benchmark for single-camera multiple target tracking. *IJCV*, 2021.

[6] Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *NeurIPS*, 2023.

[7] Georgios D. Evangelidis and Emmanouil Z. Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(10): 1858–1865, 2008. doi: 10.1109/TPAMI.2008.113.

[8] Matteo Fabbri, Guillem Brasó, Gianluca Maugeri, Orcun Cetintas, Riccardo Gasparini, Aljoša Ošep, Simone Calderara, Laura Leal-Taixé, and Rita Cucchiara. Motsynth: How can synthetic data help pedestrian detection and tracking? In *ICCV*, 2021.

[9] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Image Analysis (SCIA 2003)*, volume 2749 of *Lecture Notes in Computer Science*, pages 363–370. Springer, 2003. doi: 10.1007/3-540-45103-X_50.

[10] Ziqi Gao, Jieyu Zhang, Wisdom Oluchi Ikezogwo, Jae Sung Park, Tario G. You, Daniel Ogbu, Chenhao Zheng, Weikai Huang, Yinuo Yang, Winson Han, Quan Kong, Rajat Saini, and Ranjay Krishna. Synthetic visual genome 2: Extracting large-scale spatio-temporal scene graphs from videos. In *ECCV*, 2026.

[11] Daniel Geng, Charles Herrmann, Junhwa Hur, Forrester Cole, Serena Zhang, Tobias Pfaff, Tatiana Lopez-Guevara, Carl Doersch, Yusuf Aytar, Michael Rubinstein, et al. Motion prompting: Controlling video generation with motion trajectories. In *CVPR*, 2025.

[12] Mariam Hassan, Sebastian Stapf, Ahmad Rahimi, Pedro Rezende, Yasaman Haghighi, David Brüggemann, Isinsu Katircioglu, Lin Zhang, Xiaoran Chen, Suman Saha, et al. Gem: A generalizable ego-vision multimodal world model for fine-grained ego-motion, object dynamics, and scene composition control. In *CVPR*, pages 22404–22415, 2025.

[13] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. In *ICLR*, 2023.

[14] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. GAIA-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

[15] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022.

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

[17] Longbin Ji, Lei Zhong, Pengfei Wei, and Changjian Li. Posetraj: Pose-aware trajectory control in video diffusion. In *CVPR*, 2025.

[18] Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker3: Simpler and better point tracking by pseudo-labeling real videos. In *ICCV*, 2025.

[19] Quanhao Li, Zhen Xing, Rui Wang, Hui Zhang, Qi Dai, and Zuxuan Wu. Magicmotion: Controllable video generation with dense-to-sparse trajectory guidance. In *ICCV*, 2025.

[20] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *CVPR*, 2023.

[21] Junbang Liang, Ruoshi Liu, Ege Ozguroglu, Sruthi Sudhakar, Achal Dave, Pavel Tokmakov, Shuran Song, and Carl Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. In *Conference on Robot Learning (CoRL)*, 2024.

[22] Zuhao Liu, Aleksandar Yanev, Ahmad Mahmood, Ivan Nikolov, Saman Motamed, Wei-Shi Zheng, Xi Wang, Lei Sun, Luc Van Gool, and Danda Pani Paudel. Intragen: Trajectory-controlled video generation for object interactions. *arXiv preprint arXiv:2411.16804*, 2024.

[23] Wan-Duo Kurt Ma, John P Lewis, and W Bastiaan Kleijn. Trailblazer: Trajectory control for diffusion-based video generation. In *SIGGRAPH Asia*, 2024.

[24] 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.

[25] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017.

[26] Haonan Qiu, Zhaoxi Chen, Zhouxia Wang, Yingqing He, Menghan Xia, and Ziwei Liu. Freetraj: Tuning-free trajectory control in video diffusion models, 2024. URL `https://arxiv.org/abs/2406.16863`.

[27] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment Anything in Images and Videos. In *Int. Conf. Learn. Represent.*, 2025.

[28] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges, 2019. URL `https://arxiv.org/abs/1812.01717`.

[29] 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.

[30] Zhang Wan, Sheng Tang, Jiawei Wei, Ruize Zhang, and Juan Cao. Dragentity: Trajectory guided video generation using entity and positional relationships. In *ACM MM*, 2024.

[31] Angtian Wang, Haibin Huang, Jacob Zhiyuan Fang, Yiding Yang, and Chongyang Ma. Ati: Any trajectory instruction for controllable video generation. *arXiv preprint arXiv:2505.22944*, 2025.

[32] Hanlin Wang, Hao Ouyang, Qiuyu Wang, Wen Wang, Ka Leong Cheng, Qifeng Chen, Yujun Shen, and Limin Wang. Levitor: 3d trajectory oriented image-to-video synthesis. In *CVPR*, 2025.

[33] Jiawei Wang, Yuchen Zhang, Jiaxin Zou, Yan Zeng, Guoqiang Wei, Liping Yuan, and Hang Li. Boximator: Generating rich and controllable motions for video synthesis. In *ICML*, 2024.

[34] Qinghe Wang, Yawen Luo, Xiaoyu Shi, Xu Jia, Huchuan Lu, Tianfan Xue, Xintao Wang, Pengfei Wan, Di Zhang, and Kun Gai. Cinemaster: A 3d-aware and controllable framework for cinematic text-to-video generation. In *SIGGRAPH*, 2025.

[35] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drive-Dreamer: Towards real-world-driven world models for autonomous driving. In *ECCV*, 2024.

[36] 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*, 2024.

[37] 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.

[38] Zi-Yi Wu, Aliaksandr Siarohin, Willi Menapace, Ivan Skorokhodov, Yuwei Fang, Varnith Chordia, Igor Gilitschenski, and Sergey Tulyakov. Mind the time: Temporally-controlled multi-event video generation. In *CVPR*, 2025.

[39] Jinbo Xing, Long Mai, Cusuh Ham, Jiahui Huang, Aniruddha Mahapatra, Chi-Wing Fu, Tien-Tsin Wong, and Feng Liu. Motioncanvas: Cinematic shot design with controllable image-to-video generation. In *SIGGRAPH*, 2025.

[40] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. In *NeurIPS*, 2024.

[41] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. CogVideoX: Text-to-Video Diffusion Models with An Expert Transformer. In *ICLR*, 2025.

[42] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.

[43] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[44] Zhenghao Zhang, Junchao Liao, Menghao Li, Zuozhuo Dai, Bingxue Qiu, Siyu Zhu, Long Qin, and Weizhi Wang. Tora: Trajectory-oriented diffusion transformer for video generation. In *CVPR*, 2025.

[45] Zhiyuan Zhang, Can Wang, Dongdong Chen, and Jing Liao. Flextraj: Image-to-video generation with flexible point trajectory control. *arXiv preprint arXiv:2510.08527*, 2025.

[46] 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*, 2025.

[47] 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.

Appendix Contents

A Real-Data Models for Moving-Camera Footage **15**

A.1 Training Data and Inputs 15

A.2 Benchmarks and Protocol 15

A.3 Results 16

B Additional Qualitative Comparisons **19**

C Failure Cases **22**

C.1 GTA-V Training-Distribution Leakage on Out-of-Distribution Scenes 22

C.2 First-Frame Rendering Artifacts on the WaN Backbone 24

C.3 Unnatural Motion of Untracked Objects 24

D Joint Self-Attention: Derivations and Feasibility **25**

D.1 Attention-Weight Matrix Sizes: Cross-Attention vs. Joint Self-Attention 25

D.2 Weight Replacement Approximation in Self-Attention 25

D.3 Exact Form of Joint-Attention Replacement 26

D.4 Additive Correction: Supervised/Non-Supervised Form 27

D.5 Additive vs. Replacement Correction on CogVideoX 28

E Encoder Architectures and Training **28**

E.1 Trajectory Encoder Details 28

E.2 Appearance Encoder Details 29

E.3 Additional Implementation Details 30

F Evaluation Protocol **30**

F.1 Training Data Composition 30

F.2 Inference Settings 30

F.3 DAVIS Static-Camera Evaluation Subset 30

F.4 Per-Video Captions 31

F.5 WaN Baseline Frame Count 32

G MOTSynth-Multi **33**

H Extended Ablations **35**

H.1 Additive and Leave-One-Out Ablation on Pool and Football 35

H.2 Design Choices 35

H.3 Seed Variance of the Leave-One-Out Ablation 35

H.4 Depth as an Optional Input 36

I Attention Localization Step by Step **37**

A Real-Data Models for Moving-Camera Footage

The models of Sec. 4 are trained on synthetic, static-camera clips, while all four baselines are trained on large-scale real-world video. To evaluate our method under training conditions closer to those of the baselines, we retrain it on real, moving-camera footage and evaluate it on MagicBench [19] and MoveBench-Multi [4], benchmarks introduced with MagicMotion and Wan-Move, respectively. The method itself is unchanged: attention localization, the trajectory and appearance tokens, the training objective, and all hyperparameters are those of Sec. 3 and Sec. 4.1. Only the training data and the inputs derived from it differ, as described below. The checkpoints released with our code are the real-data models of this section.

A.1 Training Data and Inputs

Clips and objects We build the training set from SA-V [27], a large and diverse corpus of real videos with frequent camera motion, using the SVG2 [10] panoptic-mask and scene-graph annotations. From about 41K source videos we cut 120,643 clips of 49 frames at 720×480, taking up to three non-overlapping clips per video and keeping clips with at least two annotated objects. In each clip we keep the most prominent objects by image area, merge parts such as clothing into their parent person or vehicle, and discard abstract or atmospheric labels.

Trajectories Under camera motion, an object's image position can change abruptly. We track each object with CoTracker3 [18], seeding a small set of candidate points at the frame where the object is most visible and keeping the candidate whose track stays inside the object's mask the longest. At every frame, the tracked position is accepted only if it falls inside the object's mask and is marked as absent otherwise, which prevents the track from jumping to a similar-looking neighbor under occlusion or fast camera motion.

Camera motion through background trajectories Our method has no dedicated camera input. Because the SA-V annotations are panoptic, static scene elements such as walls, ground, or vegetation are tracked in the same way and given to the model as ordinary trajectory-controlled objects. Camera and scene motion are therefore conveyed by the same per-object mechanism as the motion of the foreground objects.

Prompts and depth The prompt follows the template of Eq. 1, naming every tracked object with its trajectory token, and is followed by a scene caption generated by Qwen2.5-VL [1] with the protocol of Appendix F.4. The depth channel is set to zero, so the trajectory token encodes only image-plane motion.

Training We fine-tune from the released CogVideoX-5B and WaN 2.1-14B I2V weights with the configuration of Appendix E.3 on a single H100 GPU per backbone, with batch size 3 for CogVideoX and 2 for WaN. We release and report the checkpoints at step 28,500 for CogVideoX ($\sim$8 days) and step 49,000 for WaN ($\sim$8.7 days).

A.2 Benchmarks and Protocol

Benchmarks MagicBench [19] is the benchmark released with MagicMotion. It contains 600 real clips with per-frame instance masks, organized into six bins of 100 clips with 1 to 6 or more annotated objects, and ships neither trajectories nor captions. MoveBench-Multi is the multi-object subset of MoveBench [4], the benchmark released with Wan-Move. It contains 192 clips with two to four annotated objects and ships ground-truth point tracks, first-frame masks, and a caption per clip. Both benchmarks include many clips with camera motion. Every clip is converted to our format of 49 frames at 720×480, taking the first 49 frames of each MoveBench-Multi clip.

Inputs On MoveBench-Multi, foreground trajectories are the benchmark's ground-truth tracks; on MagicBench, they are obtained from the instance masks with the tracking procedure above. Since neither benchmark annotates the background, we discover up to four background regions per clip automatically by running SAM2 [27] in automatic mask-generation mode on the first frame, discarding regions that overlap the foreground, keeping the largest remaining regions, and tracking them with CoTracker3. MoveBench-Multi clips use their shipped caption, and MagicBench clips receive a Qwen2.5-VL caption. All methods, including ours, receive the same first frame, foreground and background trajectories, and caption. All methods are scored with the metric code of Sec. 4.1.

A.3 Results

Table S1 reports both benchmarks. With both backbones, our method achieves the best PSNR, LPIPS, and FVD on both benchmarks, and the best EPE in three of the four backbone and benchmark combinations; on MagicBench with WaN, Wan-Move attains a lower EPE (2.28 vs. 2.39). The margins, 0.2–1.1 dB PSNR over the strongest same-backbone baseline, are smaller than in Tables 1–2, where four of the six datasets lie within our training distribution, but consistent across both benchmarks and backbones. Figures S1 and S2 compare all methods on scenes from both benchmarks, and Figure S3 shows further results of our models alone.

Table S1: **Real-data models on MagicBench and MoveBench-Multi.** MagicBench (600 clips) and MoveBench-Multi (192 clips) both include moving-camera footage. All methods receive the identical first frame, foreground and background trajectories, and caption; no depth is used. Best within each backbone group in bold.

		MagicBench				**MoveBench-Multi**			
	Method	PSNR↑	LPIPS↓	FVD↓	EPE↓	PSNR↑	LPIPS↓	FVD↓	EPE↓
Wan 14B	**Ours**	**18.65**	**.276**	**11.51**	2.39	**18.95**	**.244**	**26.91**	**1.74**
	ATI	17.18	.336	12.82	2.61	18.57	.274	28.62	1.98
	Wan-Move	17.57	.293	11.67	**2.28**	17.83	.259	27.32	1.79
CogV	**Ours**	**18.83**	**.267**	**10.94**	**1.97**	**19.35**	**.228**	**26.03**	**1.62**
	MagicMotion	18.19	.300	13.37	2.51	19.17	.242	27.41	1.71
	Tora	15.88	.431	44.75	3.03	17.51	.319	42.23	1.96

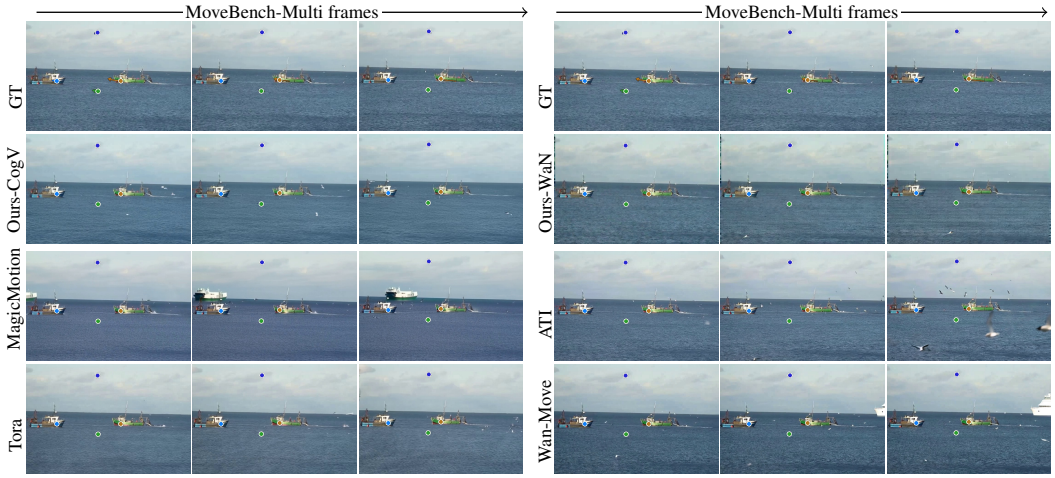

Figure S1: **Qualitative comparison on MoveBench-Multi with our real-data models.** The same scene is shown for both backbones: CogVideoX-5B against MagicMotion and Tora (left) and WaN 2.1-14B against ATI and Wan-Move (right). Dots include the background regions given to every method as additional trajectories. MagicMotion and Wan-Move insert a large white ship that is absent from the ground truth, and ATI fills the foreground with seagulls, while our method keeps both fishing boats on their trajectories without inserting new vessels.

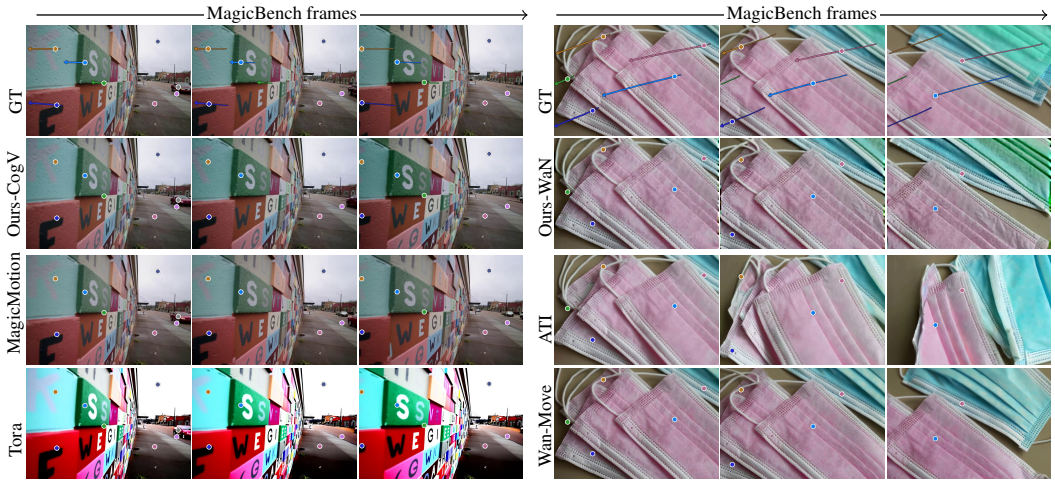

Figure S2: **Qualitative comparison on MagicBench with our real-data models.** Both scenes are filmed by a moving camera, whose motion is conveyed to every method through the trajectories of background regions. Left: CogVideoX-5B against MagicMotion and Tora; Tora oversaturates the painted wall and washes out the sky. Right: WaN 2.1-14B against ATI and Wan-Move; ATI folds and distorts the masks as the camera pans. Our method follows the camera motion of the ground truth, revealing the blue masks at the top right, while preserving the appearance of the scene.

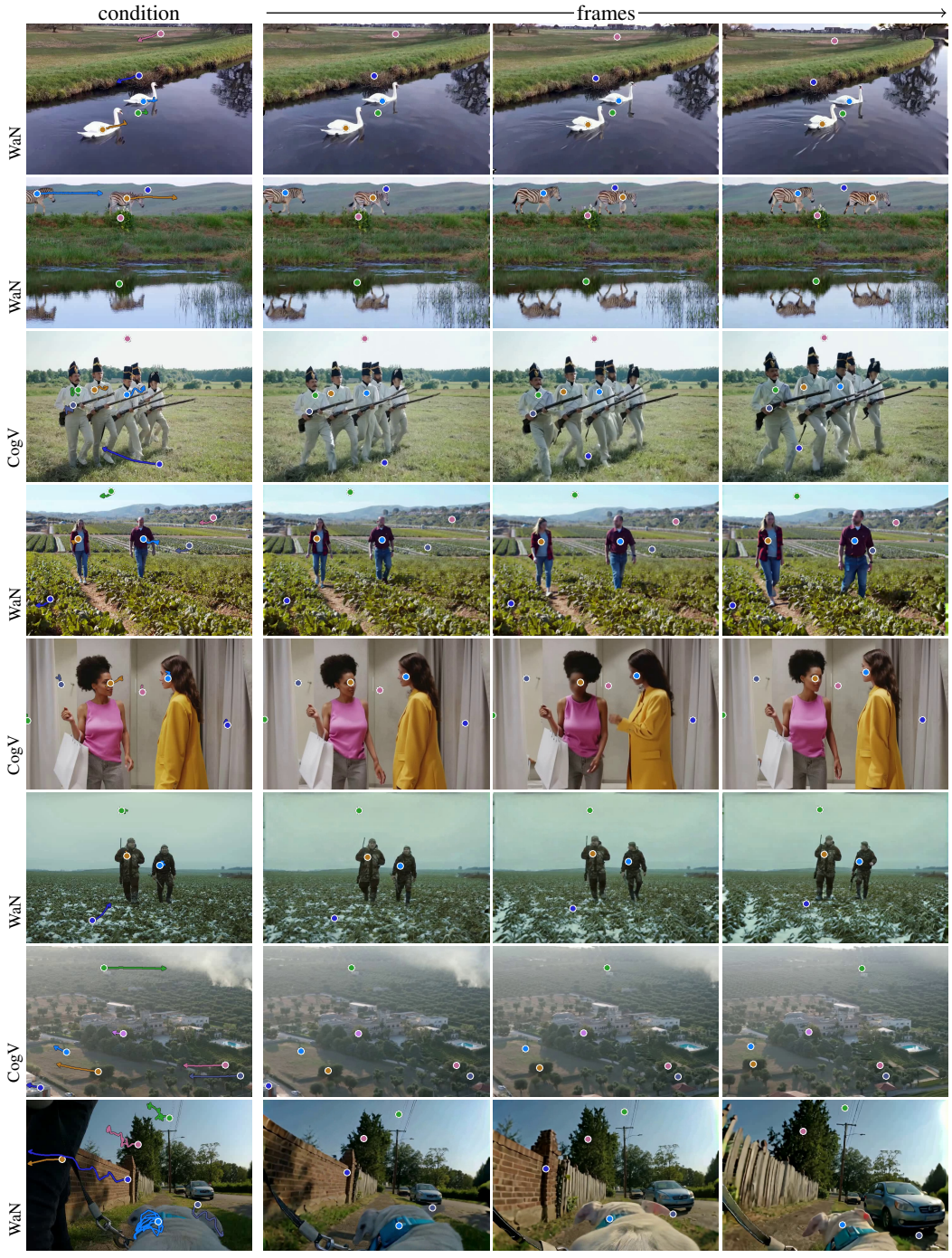

Figure S3: **Results of our real-data models on real, moving-camera footage.** Rows 1–6 are from MoveBench-Multi and rows 7–8 from MagicBench; the backbone is indicated on the left. The first column shows the input first frame with every prescribed trajectory, including the background regions that convey camera motion; the remaining columns show generated frames 9, 29 and 49 with the target position of each trajectory as a colored dot. Full videos are on the project page.

B Additional Qualitative Comparisons

We provide additional qualitative comparisons extending the main paper (Figure 4), with one scene per backbone on every evaluation dataset. In all figures below, each row shows three uniformly spaced frames (9, 29, 49). GT rows show full trajectory paths, and other rows show only the per-object ground-truth location as a dot. Across the synthetic ViN datasets (MoVi, Figure S4; Pool, Figure S5), the synthetic pedestrian dataset (MOTSynth, Figure S6), and the two real-world scene datasets (MOT17, Figure S8; DAVIS, Figure S7), our method keeps each object aligned with its prescribed trajectory while the baselines drift, merge trajectories, or fail to move objects at all. The real-world scenes are out-of-distribution for our synthetically trained model but within the training domain of the baselines, which are all trained on large-scale real-world video data.

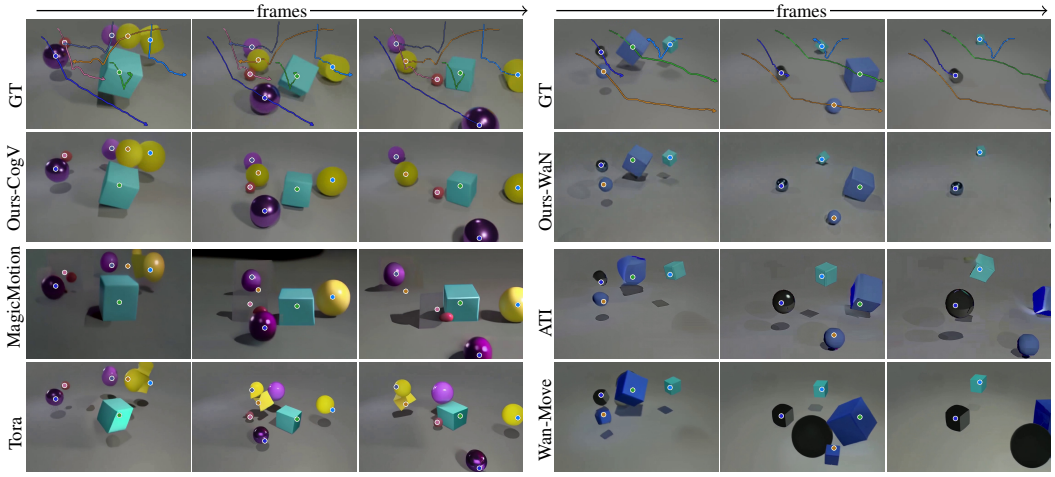

Figure S4: **Additional qualitative comparison on MoVi.** Left: CogVideoX-5B against MagicMotion and Tora. Right: Wan2.1-14B against ATI and Wan-Move. MagicMotion fails to move the object at the orange dot, and Tora removes the blue-dot object from the scene entirely. ATI and Wan-Move retain the green and orange objects even after their trajectories have left the frame. Our method follows all trajectories with consistent identity in both backbones.

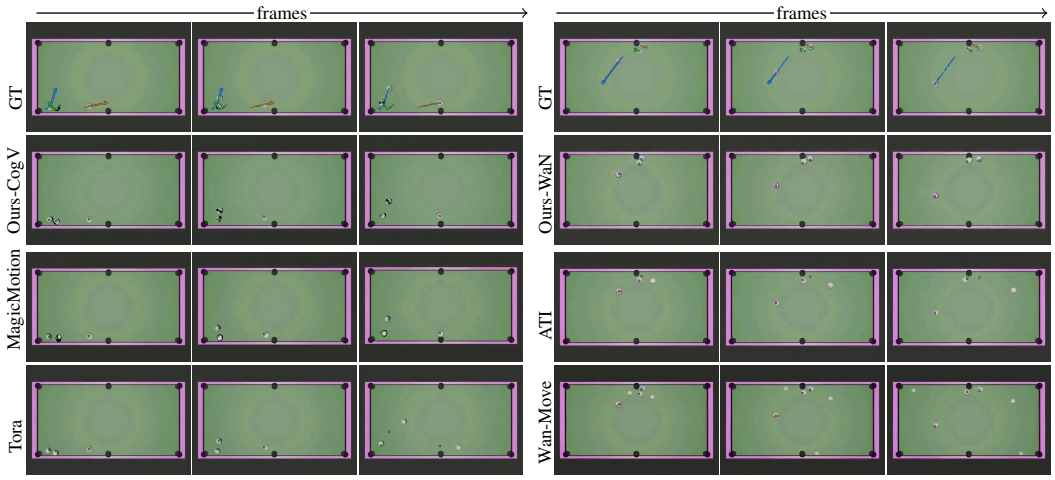

Figure S5: **Additional qualitative comparison on Pool.** Left: CogVideoX-5B against MagicMotion and Tora. Right: Wan2.1-14B against ATI and Wan-Move. Our method follows the ball locations exactly throughout the sequence. Tora and ATI maintain the correct number of balls but drift from the prescribed positions, as visible in the final frame. Wan-Move follows the trajectories but hallucinates additional objects, and MagicMotion shows slight positional errors.

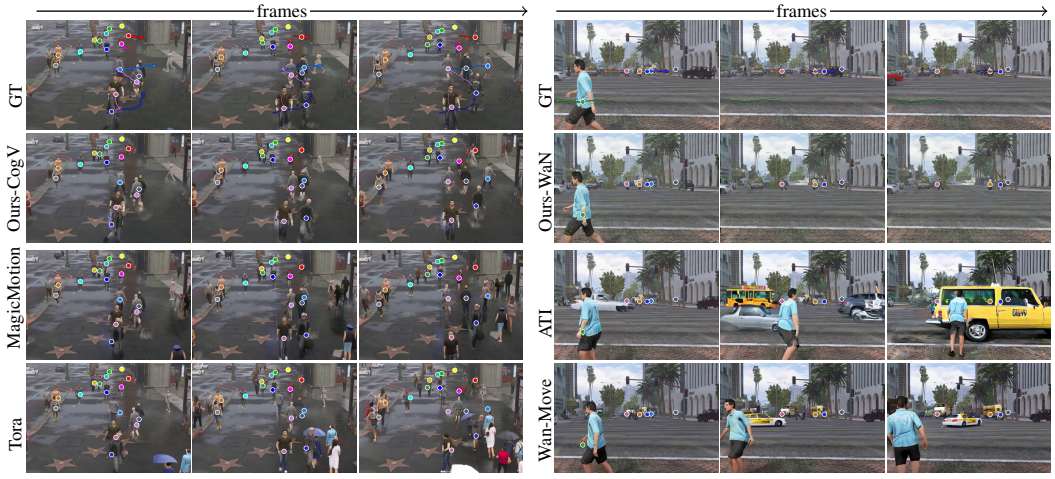

Figure S6: **Additional qualitative comparison on MOTSynth.** Left: CogVideoX-5B against MagicMotion and Tora. Right: Wan2.1-14B against ATI and Wan-Move. MagicMotion fails to follow several pedestrians after trajectory intersections, as seen at the brown and pink dots where objects drift from their prescribed positions. Tora also struggles with occlusion, notably the gray-dot pedestrian on the left side of the scene. ATI and Wan-Move also fail, for example the green-dot pedestrian should exit the scene but remains visible.

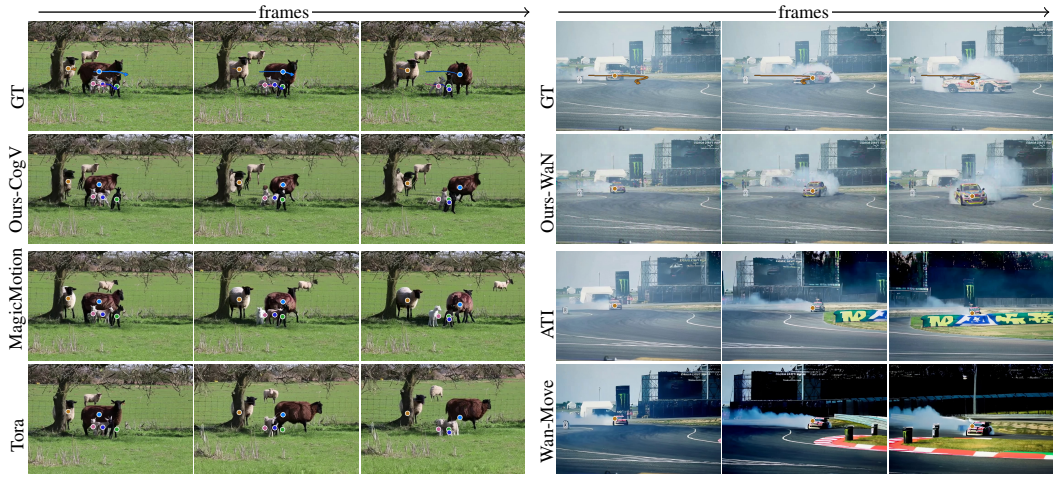

Figure S7: **Additional qualitative comparison on DAVIS.** Left: CogVideoX-5B against MagicMotion and Tora on `sheep`. Right: Wan2.1-14B against ATI and Wan-Move on `drift-chicane`. On the `sheep` scene, Tora struggles with the blue-dot object while MagicMotion performs reasonably well. On `drift-chicane`, a single-object scene with no interactions, ATI and Wan-Move move the camera rather than the car to follow the prescribed trajectory. Our method correctly moves the object itself in both cases.

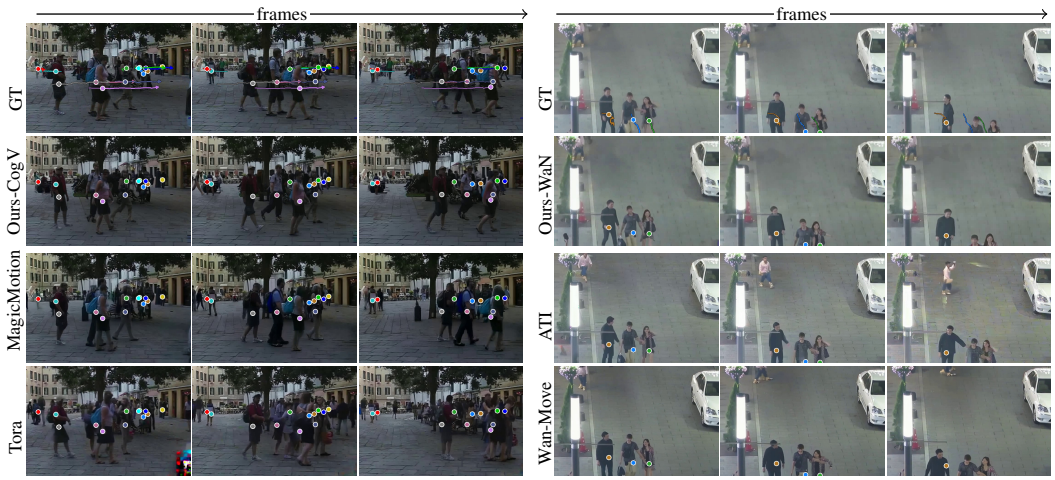

Figure S8: **Additional qualitative comparison on MOT17.** Left: CogVideoX-5B against MagicMotion and Tora. Right: WaN 2.1-14B against ATI and Wan-Move. This real-world scene features many pedestrians with frequent occlusions (video results on our project page provide a clearer view). MagicMotion changes the appearance of the pink-dot pedestrian (white shirt), and Tora causes him to disappear entirely, in addition to inaccurate motion on other trajectories. ATI and Wan-Move succeed in moving objects but with less precision, while our method places each pedestrian's center exactly on its target dot.

C Failure Cases

C.1 GTA-V Training-Distribution Leakage on Out-of-Distribution Scenes

Our CogVideoX model exhibits a failure mode on out-of-distribution scenes. When trajectory guidance forces an object through large displacements, the model occasionally morphs the object's appearance into a GTA-V pedestrian sprite mid-generation. We attribute this to the dominance of MOTSynth [8] in the training mixture ($\sim$44% of clips, all rendered from GTA V). When the object category is underrepresented in training, the appearance encoder's signal becomes insufficient to override the strong GTA-V prior learned by the backbone, causing the model to default to the most frequently seen appearance. The same effect occurs on MOT17, though it is less visually pronounced since the scenes already contain pedestrians and the GTA-V style is closer to the target domain.

Figure S9 illustrates this on the DAVIS `rallye` scene. The rally car is generated correctly at the start but visibly morphs into a GTA-V pedestrian sprite mid-clip. To support our hypothesis that this stems from overfitting to the synthetic data, we generate with an earlier checkpoint (5,500 steps) of the same model, which produces a rally car with stable appearance throughout. Quantitatively, the earlier checkpoint improves PSNR by 3%, EPE by 5%, and FVD by 12% on DAVIS. We report the later checkpoint in the main tables because it yields the best performance across all six datasets.

This observation reflects a broader limitation of the current training setup. There is no publicly available large-scale static-camera dataset of real-world multi-object trajectories at the scale required for video diffusion fine-tuning, so our training mixture is heavily weighted toward synthetic sources. When the evaluation distribution drifts far from those sources, style priors from the dominant source leak into the output. Scaling this line of work to broader real-world scenes will require collecting or curating a comparably large multi-object trajectory dataset outside the synthetic domain.



Figure S9: **GTA-V training-distribution leakage on the DAVIS `rallye` scene.** Checkpoint 8,500 (middle row) generates the rally car correctly at the start but morphs it into a GTA-V pedestrian sprite mid-clip. Checkpoint 5,500 (bottom row), from earlier in training, retains stable car appearance throughout.



Figure S10: **First-frame rendering artifacts on the WaN backbone.** Frame 1 of three DAVIS scenes. With the reported guidance scale gs=5.0 (middle column), the first few frames of some scenes show visible rendering artifacts that do not appear in the GT. Lowering to gs=3.0 (right column) mitigates the effect while leaving object appearance intact.



Figure S11: **Unnatural motion of untracked objects.** A MOTSynth scene shown at four uniformly spaced frames (1, 17, 33, 49). The GT row shows the full trajectory paths provided as input. The Ours rows show only the dots at the current frame, so any residual motion visible in the scene comes from the model rather than the conditioning. Both backbones follow the tracked trajectories but move *untracked* objects implausibly. The woman in a white tank top in the bottom-left of frame 1. On Ours-WaN she merges with a tracked pedestrian under occlusion. On Ours-CogV she drifts backwards out of frame.

C.2 First-Frame Rendering Artifacts on the WaN Backbone

On some scenes, the first few frames of our WaN model show visible rendering artifacts that are absent from the ground truth and do not persist beyond those opening frames. We hypothesize that this artifact stems from the interaction between our method and classifier-free guidance (CFG). During CFG, the model performs both a conditional and an unconditional forward pass, and our method is applied identically to both branches. Since the conditional branch contains the learned subject tokens while the unconditional branch does not, applying the same intervention to both may introduce inconsistencies. We find that simply lowering the CFG scale from gs=5.0 to gs=3.0, while keeping our method unchanged, visibly reduces the artifacts. This supports our hypothesis, as a lower scale reduces the influence of the unconditional branch. We note that all metrics reported in the paper are computed with the default parameters (gs=5.0). Figure S10 shows the first frame of three DAVIS examples. First-frame PSNR (gs=5.0 $\rightarrow$ gs=3.0) improves from 24.82 to 27.94 on gold-fish, 22.71 to 24.17 on schoolgirls, and 22.61 to 25.15 on soccerball. Designing branch-specific interventions that handle the conditional and unconditional passes differently is left for future work.

C.3 Unnatural Motion of Untracked Objects

Our trajectory inputs specify the motion of a subset of the objects in the scene, and we rely on the base I2V model's generative prior to animate everything else (background, crowds, occluders). On crowded MOTSynth scenes, we occasionally observe implausible motion of these untracked objects. Figure S11 shows a representative example: the woman in a white tank top in the bottom-left of the first frame is not part of the trajectory inputs. On Ours-WaN she merges with a tracked pedestrian under occlusion, and on Ours-CogV she drifts backwards out of frame in a way inconsistent with natural walking. The tracked objects follow their prescribed trajectories correctly in both cases. The failure reflects the limits of what trajectory conditioning alone can enforce: the model has no signal for what untracked objects should do, and nothing in our attention replacement constrains them.

D Joint Self-Attention: Derivations and Feasibility

D.1 Attention-Weight Matrix Sizes: Cross-Attention vs. Joint Self-Attention

Our column replacement (Eq. 3) overwrites entries of the attention-weight matrix $\mathbf{A}$ before it is multiplied by $\mathbf{V}$. Whether this can be applied exactly depends on whether $\mathbf{A}$ is small enough to store in memory at the scale of the target backbone. The two backbones we consider sit on opposite sides of this boundary, and this appendix makes the distinction quantitative.

We instantiate both cases at 720×480 resolution with 49 video frames, which yields a latent spatiotemporal grid of $F\times h\times w = 13\times30\times45$ and therefore $M = F\cdot h\cdot w = 17{,}550$ video tokens.

Explicit cross-attention (WaN 2.1-14B) WaN 2.1 uses separate cross-attention: video tokens $\mathbf{Q}_{\text{vid}} \in \mathbb{R}^{M\times d}$ attend to text tokens $\mathbf{K}_{\text{txt}}, \mathbf{V}_{\text{txt}} \in \mathbb{R}^{L\times d}$, with $L = 512$ and $d = 128$ over 40 heads. The per-head attention matrix is therefore

$$\mathbf{A} \in \mathbb{R}^{M\times L} = \mathbb{R}^{17{,}550\times512}, \tag{6}$$

which is approximately $17{,}550 \times 512 \times 2$ bytes ≈ 18 MB per head in bfloat16. At 40 heads, the full per-layer footprint is $\sim$720 MB, which fits comfortably alongside the model activations. $\mathbf{A}$ can therefore be stored in memory, the supervised columns at $\{q_i\}$ overwritten, and the row-wise renormalization of Eq. 3 applied directly.

Joint self-attention (CogVideoX-5B) CogVideoX concatenates text and video tokens into a single sequence of length $L_{\text{tot}} = L + M = 226 + 17{,}550 = 17{,}776$ and performs a single self-attention pass over the joint sequence. The per-head attention matrix is

$$\mathbf{A} \in \mathbb{R}^{L_{\text{tot}}\times L_{\text{tot}}} = \mathbb{R}^{17{,}776\times17{,}776}, \tag{7}$$

approximately $17{,}776^2 \times 2$ bytes ≈ 0.6 GB per head in bfloat16. With 48 heads the per-layer footprint is $\sim$30 GB, and the transformer has tens of such layers. Storing $\mathbf{A}$ in memory is therefore infeasible in practice, which is precisely why production implementations of joint self-attention rely on FlashAttention-style kernels that never store $\mathbf{A}$ explicitly.

D.2 Weight Replacement Approximation in Self-Attention

In joint self-attention architectures such as CogVideoX [41], text and video tokens form a single sequence. The full attention matrix is then too large to store in memory, reaching roughly 30 GB per layer for CogVideoX (Appendix D.1). Standard efficient kernels (e.g., FlashAttention) therefore never build $\mathbf{A}$ explicitly, and applying Eq. 3 exactly would require a custom kernel that performs the column overwrite inside the tiled attention loop. Instead, we derive two alternatives that operate entirely with standard scaled dot-product attention (SDPA) calls.

Below we first state the exact column-replacement output for joint self-attention, then derive two relaxations of it that use only standard SDPA calls.

Exact formulation We derive the counterpart of the cross-attention output $\mathbf{O}' = \mathbf{A}'\mathbf{V}_{\text{txt}}$ (§3.1) for the joint self-attention setting. Let $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{L_{\text{tot}}\times d}$ denote the joint query, key, and value matrices, with $L_{\text{tot}} = L + M$ and per-head dimension d. We order joint tokens as [text; video], so video position $m \in \{1, \dots, M\}$ corresponds to joint index $L + m$, and we write $\mathbf{Q}_{\text{vid}}$ for the video-row submatrix of $\mathbf{Q}$. We write $\mathrm{SDPA}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ for the scaled dot-product attention output and $\mathrm{SDPA}(\mathbf{Q}, \mathbf{K}, \mathbf{V})[m]$ for its m-th row. Throughout this section and Appendices D.3 and D.4, we write $\mathbf{A}[m, \cdot]$ as shorthand for the m-th video-query row of the joint attention matrix (i.e., joint row $L + m$), for $m \in \{1, \dots, M\}$.

Let $\{q_i\}_{i=1}^N$ denote the object-token columns introduced in §3.1 (the columns of $\mathbf{A}$ where attention weights are replaced by trajectory heatmaps); we refer to these as *supervised positions*. Let $\mathcal{U} = \{1, \dots, L_{\text{tot}}\} \setminus \{q_i\}_{i=1}^N$ be the complementary set of non-supervised positions, and write $\mathbf{K}_{\mathcal{U}}, \mathbf{V}_{\mathcal{U}}$ for the submatrices restricted to those rows. Define

$$H[m] = \sum_{i=1}^{N} \mathbf{h}_i[m], \qquad S_{\mathcal{U}}[m] = \sum_{j\in\mathcal{U}} \mathbf{A}[m, j], \tag{8}$$

where $H[m]$ is the total trajectory heatmap mass at video position m, and $S_{\mathcal{U}}[m]$ is the native attention mass assigned to the non-supervised positions. The exact counterpart of $\mathbf{O}' = \mathbf{A}'\mathbf{V}_{\text{txt}}$ for joint

self-attention is

$$\mathbf{O}_{\text{vid}}^{\text{exact}}[m] = \frac{S_{\mathcal{U}}[m]}{S_{\mathcal{U}}[m] + H[m]} \,\text{SDPA}(\mathbf{Q}_{\text{vid}}, \mathbf{K}_{\mathcal{U}}, \mathbf{V}_{\mathcal{U}})[m] + \frac{1}{S_{\mathcal{U}}[m] + H[m]} \sum_{i=1}^{N} \mathbf{h}_i[m]\mathbf{V}[q_i]. \quad (9)$$

The derivation of Eq. 9 is provided in Appendix D.3.

First relaxation: additive correction A simple relaxation preserves the native joint self-attention output at the video-token positions and adds the supervised contribution:

$$\begin{aligned} \mathbf{O}_{\text{vid}}^{\text{add}}[m] &= \text{SDPA}(\mathbf{Q}, \mathbf{K}, \mathbf{V})[L+m] + \sum_{i=1}^{N} \mathbf{h}_i[m]\,\mathbf{V}[q_i] \\ &= S_{\mathcal{U}}[m]\,\text{SDPA}(\mathbf{Q}_{\text{vid}}, \mathbf{K}_{\mathcal{U}}, \mathbf{V}_{\mathcal{U}})[m] + \sum_{i=1}^{N} \big(\mathbf{A}[m, q_i] + \mathbf{h}_i[m]\big)\,\mathbf{V}[q_i]. \end{aligned} \quad (10)$$

This form is close to Eq. 9 but weaker: the supervised contribution carries $\big(\mathbf{A}[m, q_i] + \mathbf{h}_i[m]\big)$, so the native attention weight $\mathbf{A}[m, q_i]$ sits *on top of* the heatmap $\mathbf{h}_i[m]$ rather than being replaced by it, and the non-supervised coefficient $S_{\mathcal{U}}[m]$ is not renormalized against the supervised mass. The derivation of the second line is given in Appendix D.4. The second line of Eq. 10 is an equivalent rewrite of the first, not a computational recipe: the additive correction is implemented using the first line, which avoids forming $\mathbf{A}$ and therefore avoids accessing the $S_{\mathcal{U}}[m]$ and $\mathbf{A}[m, q_i]$ quantities that appear only for comparison with Eq. 9.

Second relaxation: Two-SDPA approximation A stronger relaxation follows the same supervised/non-supervised decomposition as Eq. 9, but replaces the full-sequence SDPA call with an SDPA call restricted to $\mathcal{U}$. The supervised term then drops the native weight $\mathbf{A}[m, q_i]$ and carries only the heatmap; the non-supervised term is rescaled by $1 - H[m]$, the mass complement of the supervised contribution, yielding

$$\mathbf{O}_{\text{vid}}^{\text{2sdpa}}[m] = (1 - H[m])\,\text{SDPA}(\mathbf{Q}_{\text{vid}}, \mathbf{K}_{\mathcal{U}}, \mathbf{V}_{\mathcal{U}})[m] + \sum_{i=1}^{N} \mathbf{h}_i[m]\mathbf{V}[q_i]. \quad (11)$$

The approximation becomes exact whenever the base model's native attention at the supervised columns already matches the target heatmap mass, $\sum_{i=1}^{N} \mathbf{A}[m, q_i] = H[m]$ (equivalently, $S_{\mathcal{U}}[m] = 1 - H[m]$). Both formulations also coincide when $H[m] = 0$, i.e., far from all trajectories, where the supervised contribution vanishes and the non-supervised coefficient evaluates to 1 in both forms, reducing to $\text{SDPA}(\mathbf{Q}_{\text{vid}}, \mathbf{K}_{\mathcal{U}}, \mathbf{V}_{\mathcal{U}})[m]$. When $H[m] > 0$, both formulations combine the same untouched-column term with the same supervised contribution $\sum_i \mathbf{h}_i[m]\mathbf{V}[q_i]$, differing only in their relative weighting. This formulation is implemented with two standard SDPA calls, incurring approximately a $2\times$ attention overhead per layer.

D.3 Exact Form of Joint-Attention Replacement

In this section, we derive the exact analogue of the column replacement in Eq. 3 for joint self-attention architectures (Eq. 9).

We use the notation established in Appendix D.2: the joint matrices $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{L_{\text{tot}} \times d}$, the supervised positions $\{q_i\}_{i=1}^{N}$ and their complement $\mathcal{U}$, and the per-position masses $H[m]$ and $S_{\mathcal{U}}[m]$. Let $\mathbf{A} = \text{softmax}(\mathbf{Q}\mathbf{K}^\top/\sqrt{d}) \in \mathbb{R}^{L_{\text{tot}} \times L_{\text{tot}}}$ denote the hypothetical full joint attention matrix (which cannot be stored in practice; Appendix D.1).

Apply the column replacement (Eq. 3) to $\mathbf{A}$: the modified matrix $\mathbf{A}'$ satisfies $\mathbf{A}'[m, j] = \mathbf{A}[m, j]$ for $j \in \mathcal{U}$ (unsupervised columns untouched) and $\mathbf{A}'[m, q_i] = \mathbf{h}_i[m]$ (supervised columns overwritten). The row-m sum after replacement is

$$\sum_{k=1}^{L_{\text{tot}}} \mathbf{A}'[m, k] = \underbrace{\sum_{j \in \mathcal{U}} \mathbf{A}[m, j]}_{S_{\mathcal{U}}[m]} + \underbrace{\sum_{i=1}^{N} \mathbf{h}_i[m]}_{H[m]}. \quad (12)$$

The renormalized output at video position m is then

$$\mathbf{O}_{\text{vid}}^{\text{exact}}[m] = \frac{1}{S_{\mathcal{U}}[m] + H[m]} \Big(\sum_{j \in \mathcal{U}} \mathbf{A}[m, j]\, \mathbf{V}[j] + \sum_{i=1}^{N} \mathbf{h}_i[m]\, \mathbf{V}[q_i] \Big). \tag{13}$$

For the unsupervised sum, factor out $S_{\mathcal{U}}[m]$:

$$\sum_{j \in \mathcal{U}} \mathbf{A}[m, j]\, \mathbf{V}[j] = S_{\mathcal{U}}[m] \sum_{j \in \mathcal{U}} \frac{\mathbf{A}[m, j]}{S_{\mathcal{U}}[m]}\, \mathbf{V}[j]. \tag{14}$$

The ratio $\mathbf{A}[m, j]/S_{\mathcal{U}}[m]$ is the softmax over only the unsupervised keys, since normalizing the original softmax weights by $S_{\mathcal{U}}[m]$ cancels the full-sequence normalization:

$$\frac{\mathbf{A}[m, k]}{\sum_{j \in \mathcal{U}} \mathbf{A}[m, j]} = \frac{\exp(\mathbf{Q}[m]\mathbf{K}[k]^\top / \sqrt{d})}{\sum_{j \in \mathcal{U}} \exp(\mathbf{Q}[m]\mathbf{K}[j]^\top / \sqrt{d})}, \qquad k \in \mathcal{U}. \tag{15}$$

The unsupervised sum therefore equals $S_{\mathcal{U}}[m]\, \mathrm{SDPA}(\mathbf{Q}_{\text{vid}}, \mathbf{K}_{\mathcal{U}}, \mathbf{V}_{\mathcal{U}})[m]$. Substituting back yields

$$\mathbf{O}_{\text{vid}}^{\text{exact}}[m] = \frac{S_{\mathcal{U}}[m]}{S_{\mathcal{U}}[m] + H[m]}\, \mathrm{SDPA}(\mathbf{Q}_{\text{vid}}, \mathbf{K}_{\mathcal{U}}, \mathbf{V}_{\mathcal{U}})[m] + \frac{1}{S_{\mathcal{U}}[m] + H[m]} \sum_{i=1}^{N} \mathbf{h}_i[m] \mathbf{V}[q_i], \tag{16}$$

which is Eq. 9. □

D.4 Additive Correction: Supervised/Non-Supervised Form

We derive the second line of Eq. 10, which rewrites the additive correction $\mathbf{O}_{\text{vid}}^{\text{add}}[m]$ in the same non-supervised/supervised decomposition used by the exact form in Eq. 16. Concretely, we show that

$$\mathrm{SDPA}(\mathbf{Q}, \mathbf{K}, \mathbf{V})[L + m] = S_{\mathcal{U}}[m]\, \mathrm{SDPA}(\mathbf{Q}_{\text{vid}}, \mathbf{K}_{\mathcal{U}}, \mathbf{V}_{\mathcal{U}})[m] + \sum_{i=1}^{N} \mathbf{A}[m, q_i]\, \mathbf{V}[q_i], \tag{17}$$

and then plug this identity into Eq. 10 to obtain its reformulated right-hand side.

Step 1: decompose the joint-SDPA row $L + m$. By definition of attention, the m-th video-query row of the joint SDPA output is

$$\mathrm{SDPA}(\mathbf{Q}, \mathbf{K}, \mathbf{V})[L + m] = \sum_{j=1}^{L_{\text{tot}}} \mathbf{A}[m, j]\, \mathbf{V}[j] = \sum_{j \in \mathcal{U}} \mathbf{A}[m, j]\, \mathbf{V}[j] + \sum_{i=1}^{N} \mathbf{A}[m, q_i]\, \mathbf{V}[q_i]. \tag{18}$$

Step 2: rewrite the non-supervised sum. Factoring out $S_{\mathcal{U}}[m]$,

$$\sum_{j \in \mathcal{U}} \mathbf{A}[m, j]\, \mathbf{V}[j] = S_{\mathcal{U}}[m] \sum_{j \in \mathcal{U}} \frac{\mathbf{A}[m, j]}{S_{\mathcal{U}}[m]}\, \mathbf{V}[j]. \tag{19}$$

The ratio $\mathbf{A}[m, j]/S_{\mathcal{U}}[m]$ is exactly the softmax over only the non-supervised keys, by the same cancellation used in Eq. 15:

$$\frac{\mathbf{A}[m, j]}{S_{\mathcal{U}}[m]} = \mathrm{softmax}\big(\mathbf{Q}_{\text{vid}}[m]\mathbf{K}_{\mathcal{U}}^\top / \sqrt{d}\big)[j], \qquad j \in \mathcal{U}. \tag{20}$$

This is the softmax used by $\mathrm{SDPA}(\mathbf{Q}_{\text{vid}}, \mathbf{K}_{\mathcal{U}}, \mathbf{V}_{\mathcal{U}})$ at query row m, so

$$\sum_{j \in \mathcal{U}} \mathbf{A}[m, j]\, \mathbf{V}[j] = S_{\mathcal{U}}[m]\, \mathrm{SDPA}(\mathbf{Q}_{\text{vid}}, \mathbf{K}_{\mathcal{U}}, \mathbf{V}_{\mathcal{U}})[m]. \tag{21}$$

Step 3: combine. Substituting back into Step 1 yields Eq. 17.

Step 4: plug into Eq. 10 and collect supervised terms. Adding $\sum_i \mathbf{h}_i[m]\,\mathbf{V}[q_i]$ to Eq. 17 and grouping the two supervised sums gives

$$\mathbf{O}^{\text{add}}_{\text{vid}}[m] = \text{SDPA}(\mathbf{Q},\mathbf{K},\mathbf{V})[L+m] + \sum_{i=1}^{N} \mathbf{h}_i[m]\,\mathbf{V}[q_i] \tag{22}$$

$$= S_{\mathcal{U}}[m]\,\text{SDPA}(\mathbf{Q}_{\text{vid}},\mathbf{K}_{\mathcal{U}},\mathbf{V}_{\mathcal{U}})[m] + \sum_{i=1}^{N} \mathbf{A}[m,q_i]\,\mathbf{V}[q_i] + \sum_{i=1}^{N} \mathbf{h}_i[m]\,\mathbf{V}[q_i] \tag{23}$$

$$= S_{\mathcal{U}}[m]\,\text{SDPA}(\mathbf{Q}_{\text{vid}},\mathbf{K}_{\mathcal{U}},\mathbf{V}_{\mathcal{U}})[m] + \sum_{i=1}^{N} \big(\mathbf{A}[m,q_i] + \mathbf{h}_i[m]\big)\,\mathbf{V}[q_i], \tag{24}$$

which is the second line of Eq. 10. □

Comparing Eq. 16 and this rewritten form of Eq. 10 side by side makes the semantic difference explicit. Both formulas combine the same non-supervised SDPA term and a supervised contribution at the same positions $\{q_i\}$, but the exact form uses renormalized coefficients $S_{\mathcal{U}}[m]/(S_{\mathcal{U}}[m]+H[m])$ and $1/(S_{\mathcal{U}}[m]+H[m])$ and relies entirely on the heatmap $\mathbf{h}_i[m]$ for the supervised contribution. The additive form, in contrast, uses the unnormalized coefficient $S_{\mathcal{U}}[m]$ and carries the native attention weight $\mathbf{A}[m,q_i]$ *in addition to* the heatmap, which is why it is a weaker correction.

D.5 Additive vs. Replacement Correction on CogVideoX

Table S2 compares the additive correction (Eq. 10) with the two-SDPA replacement (Eq. 11) on CogVideoX-5B across all six datasets. Replacement wins clearly on MOTSynth and MOT17 (all four metrics) and on Pool (three of four, EPE near-tied), while MoVi and Football are mixed. The pattern flips on DAVIS, where additive wins all four metrics. DAVIS is the only dataset whose object categories (animals, vehicles, sports) fall outside the training distribution. We report replacement throughout, as it wins on the most datasets.

Table S2: Additive vs. two-SDPA replacement on CogVideoX-5B across all six evaluation datasets. Both modes are ours; 2-SDPA is the variant used in the main-paper tables. Best per cell in **bold**.

	MoVi				**Pool**				**Football**			
Mode	PSNR↑	LPIPS↓	FVD↓	EPE↓	PSNR↑	LPIPS↓	FVD↓	EPE↓	PSNR↑	LPIPS↓	FVD↓	EPE↓
2-SDPA (reported)	**29.77**	**.108**	14.98	0.53	**35.64**	**.025**	**8.79**	0.07	24.79	.090	**13.87**	**0.03**
Additive	29.42	.116	**14.16**	**0.52**	33.42	.029	8.97	**0.06**	**24.98**	**.088**	14.60	**0.03**

	MOTSynth				**MOT17**				**DAVIS**			
Mode	PSNR↑	LPIPS↓	FVD↓	EPE↓	PSNR↑	LPIPS↓	FVD↓	EPE↓	PSNR↑	LPIPS↓	FVD↓	EPE↓
2-SDPA (reported)	**22.34**	**.232**	**39.70**	**2.32**	**21.59**	**.265**	**53.92**	**2.37**	18.39	.267	89.39	2.43
Additive	22.16	.234	42.31	2.42	20.76	.275	60.97	3.16	**18.84**	**.259**	**80.01**	**2.23**

E Encoder Architectures and Training

E.1 Trajectory Encoder Details

This section provides architectural and training details for the trajectory encoder $\mathcal{E}_{\text{traj}}$ introduced in Sec. 3.2.1.

Encoder architecture The encoder Enc_{traj} consists of two stages. First, three 1D convolutions with stride 2 progressively downsample the temporal dimension while increasing the channel count ($4 \to 64 \to 128 \to 256$), each followed by batch normalization and GELU; the output is flattened rather than pooled and linearly projected to an intermediate representation $\mathbf{z}_i$ of dimension $d = 32$. The four input channels are the per-frame coordinates $(x_i(t), y_i(t), d_i(t))$ together with a normalized temporal position channel $p(t) = (t-1)/(T-1) \in [0,1]$ for $t \in \{1,\dots,T\}$, concatenated along

the channel dimension before the first convolution. Second, a two-layer MLP head maps $\mathbf{z}_i$ to the token embedding:

$$\langle traj_i \rangle = W_2\,\mathrm{GELU}(W_1\,\mathbf{z}_i), \qquad W_1 \in \mathbb{R}^{512\times d},\ W_2 \in \mathbb{R}^{4096\times 512}, \tag{25}$$

with the final layer initialized to $\mathcal{N}(0, 0.02^2)$ weights and zero bias. The output is normalized to target std ≈ 0.15 for T5-XXL and std ≈ 0.07 for UMT5-XXL.

Pretraining pipeline Pretraining uses synthetic trajectories generated on-the-fly with statistics matched to real MOTSynth tracks (1,793 trajectories). Each trajectory $(x_t, y_t, d_t) \in [0,1]^3$ is generated as a clamped random walk: we sample a start position from a per-channel Gaussian fit to the real starts (means $(0.48, 0.50, 0.40)$, stds $(0.33, 0.30, 0.23)$), draw a constant per-trajectory drift velocity (v_x, v_y, v_d) from Gaussians matching the real per-trajectory velocity stds $(0.008, 0.004, 0.001)$, and update each frame as

$$c_{t+1} = \mathrm{clamp}(c_t + v + \epsilon_t, 0, 1)\,, \qquad \epsilon_t \sim \mathcal{N}\big(0, \sigma^2_{\text{noise}}\big)\,,$$

with per-channel noise stds $\sigma_{\text{noise}} = (0.011, 0.004, 0.002)$. Each step samples 1–20 trajectories (real mean 4.7) and builds a prompt:

```
"A GTA V street scene with pedestrians.  The following
represent pedestrian trajectories:  [traj_1], [traj_2], ...".
```

The trajectory embeddings $\{\langle traj_i \rangle\}$ are injected at the `[traj_i]` positions and the frozen text encoder processes the full prompt. A symmetric decoder ($\sim$12.8M parameters) reconstructs each trajectory from the text encoder output $\mathbf{c}_i$ at the corresponding position. The decoder consists of a linear expansion followed by three stride-2 transposed 1D convolutions with sigmoid activation, and receives $\mathbf{c}_i$ scaled by a learnable scalar that compensates for the magnitude change introduced by the text encoder's layers.

Although the encoder architecture includes a stochastic bottleneck (the reparameterization $\mathbf{z}_i = \mu_i + \sigma_i \odot \epsilon$ is active during pretraining), we found empirically that removing the KL regularization term yields better trajectory reconstruction, so no KL penalty is applied. The variance σ_i is therefore shaped only by the reconstruction gradient.

After pretraining, two changes are made: (1) the decoder is discarded, and (2) the stochastic sampling is replaced with the deterministic mean, $\mathbf{z}_i = \mu_i$. $\mathrm{Enc}_{\text{traj}}$ is then frozen for all subsequent diffusion training.

Pretraining runs for 500K steps with AdamW (learning rate 3×10^{-4}, cosine decay, weight decay 10^{-2}, gradient accumulation over 8 steps), using the loss $\mathcal{L}_{\text{traj}}$ (Eq. 4) with $\lambda_{\text{vel}} = 1$ and 1–20 trajectories per prompt.

E.2 Appearance Encoder Details

This section provides architectural details for the appearance encoder $\mathrm{Enc}_{\text{app}}$ introduced in Sec. 3.2.2. Unlike the trajectory encoder, the appearance encoder is not pretrained: it is initialized randomly and trained jointly with the LoRA weights during diffusion fine-tuning.

Architecture The encoder takes the full first-frame VAE latent $V_0 \in \mathbb{R}^{16\times60\times90}$ (at 720×480 resolution with VAE stride 8) and produces a dense appearance feature map $\mathbf{z}^{\text{app}}$ through three 2D convolutions: one stride-2 layer ($16 \to 64$ channels, spatial $60\times90 \to 30\times45$) followed by two stride-1 refinement layers ($64 \to 64$), each with batch normalization and GELU. A 1×1 convolution head produces a per-pixel feature map of depth $D_{\text{app}} = 8$ at the compressed spatial resolution 30×45.

Position sampling and projection For each object i, the normalized starting position (x_i^1, y_i^1) is mapped to the compressed grid and the corresponding D_{app}-dimensional feature vector is extracted via indexing. A single linear layer projects this to the text encoder dimension:

$$e_{o_i} = W_{\text{app}}\,\mathbf{z}^{\text{app}}[x_i^1, y_i^1, :] \in \mathbb{R}^{4096}, \quad W_{\text{app}} \in \mathbb{R}^{4096\times8}. \tag{26}$$

The output is globally normalized to match the text encoder output statistics (std ≈ 0.15 for T5-XXL, std ≈ 0.07 for UMT5-XXL). The projection is initialized with $\mathcal{N}(0, 0.02^2)$ weights and zero bias.

E.3 Additional Implementation Details

The diffusion fine-tuning uses AdamW at learning rate 2×10^{-4} in bf16 with gradient checkpointing, for 8,500 steps on CogVideoX (effective batch size 3 across 3 H100 GPUs, $\sim$57 hours) and 12,000 steps on WaN (effective batch size 2 across 2 H100 GPUs, $\sim$50 hours).

F Evaluation Protocol

F.1 Training Data Composition

The training data comprises 11,281 clips from MOTSynth (mean 4.2 pedestrians per clip, filtered to static-camera sequences), 8,236 from MoVi-Extended (mean 5.3 objects per clip), 2,943 from Pool (2–3 balls), and 3,040 from Football (up to 2 players and a ball).

F.2 Inference Settings

Table S3 details the inference configuration for each method. All baselines use their released default settings from their official code repositories.

Table S3: Inference settings. All baselines use released defaults. MagicMotion uses bounding-box trajectory maps as ControlNet input. Solver abbreviations: CogV-DPM is the CogVideoX DPM scheduler; FM-Euler is flow-matching Euler; VP-SDE DPM++2M is the VP-SDE variant of DPM++2M from CogVideoX-SAT; FM-UniPC is flow-matching UniPC.

	Ours-Cog	**Ours-Wan**	Tora	MagicMot.	Wan-Move	ATI
Backbone	CogVideoX-5B	Wan2.1-14B	CogVideoX-5B	CogVideoX-5B	Wan2.1-14B	Wan2.1-14B
Mode	I2V	I2V	I2V	I2V	I2V	I2V
Resolution	720×480	720×480	720×480	720×480	720×480	720×480
Frames	49	49	49	49	49	49
Steps	50	50	50	50	40	40
Guidance	6.0	5.0	6.0	6.0	5.0	5.0
Solver	CogV-DPM	FM-Euler	VP-SDE DPM++2M	CogV-DPM	FM-UniPC	FM-UniPC
Traj. input	points	points	points	bboxes	points	points

F.3 DAVIS Static-Camera Evaluation Subset

To evaluate trajectory-conditioned generation on diverse real-world content beyond pedestrians, we evaluate on a subset of DAVIS 2017 [25], a video object segmentation benchmark widely used for trajectory-conditioned evaluation [11, 32, 17, 19, 4]. We use the GT segmentation masks provided by DAVIS to extract trajectories, avoiding tracker noise.

Static-camera filtering Our method controls object motion but provides no signal about camera motion. In scenes with a moving camera, the apparent pixel-space displacement of an object reflects both its own motion and the camera's, so we cannot expect trajectory alignment without explicitly modeling the camera. Starting from the 90 sequences in the DAVIS 2017 train+val splits, we filter for static-camera scenes by computing background optical flow (Farneback [9]) on sampled frame pairs using GT segmentation masks to isolate background pixels; sequences with mean background flow above 1.5 px are rejected. The remaining 13 sequences are processed as follows: (1) video stabilization via ECC alignment [7] on background-only masks with Gaussian-smoothed cumulative transforms, center-cropped from 854×480 to 720×480; (2) trajectory extraction from per-frame median center-of-mass of GT segmentation masks, linearly interpolated through occlusion gaps and Gaussian-smoothed (σ=2); (3) depth estimation with Depth Anything V2 [40]. Sequences shorter than 49 frames are temporally interpolated via nearest-neighbor frame spreading.

Final dataset. The 13 scenes span 6 categories with 37 object trajectories total:

Sequence	Category	Objects	Description
bike-packing	person	2	person working on a bicycle
breakdance	person	1	breakdancer in a city square
crossing	pedestrian	3	person, man, van at crosswalk
disc-jockey	person	2	DJ and turntable in dim room
dogs-jump	animal	3	2 dogs, 1 person
drift-chicane	vehicle	1	car on track
gold-fish	animal	5	goldfish in tank
judo	sports	2	two judokas grappling
planes-water	vehicle	2	planes on a wet runway
rallye	vehicle	1	rally car
schoolgirls	pedestrian	4	4 schoolgirls walking
sheep	animal	5	2 sheep, 3 lambs in field
soccerball	sports	1	ball on grass

All scenes use static cameras and are never seen during training. This dataset complements MOT17 (real-world but pedestrian-only) by testing generalization to animals, vehicles, sports, and other people-centric categories.

F.4 Per-Video Captions

All baselines in Tables 1–2 use per-video captions that describe the scene content. For the three ViN datasets (MoVi, Pool, Football), we use the ShareGPT4Video captions distributed with each dataset. For MOTSynth and MOT17, we generate scene-specific captions using Qwen2.5-VL-7B-Instruct [1], following the captioning methodology of Wan-Move [4].

This setup gives the baselines a textual advantage over our method. The per-video captions are produced by a video VLM (Qwen2.5-VL) that observes all frames of the clip, so they describe backgrounds, lighting, motion patterns, and even objects that are occluded in the first frame but become visible later. Our method follows the template from Eq. 1 ("Scene where o_0 moves [traj$_0$] and o_1 moves [traj$_1$] ..."). The appearance encoder likewise operates on a single patch around each object's first-frame location, so when an object is occluded at the first frame its appearance token may encode the occluder rather than the target. Baselines therefore draw textual context from the entire video, while our conditioning is built exclusively from the first frame.

Qwen2.5-VL prompt We caption clips by sampling 16 frames per video and querying Qwen2.5-VL-7B-Instruct with the following instruction:

```
Describe this video in detail.  Focus on:  (1) the scene setting and
environment, (2) the camera perspective and any camera motion, (3) the
main subjects and their appearance, (4) the motion and actions of each
subject--direction, speed, interactions, (5) lighting conditions and visual
style.  Keep the description concise (2-4 sentences) and factual.  Do not
speculate about what happens outside the frame.
```

Example caption For the MOT17 scene shown in the right panel of Figure S8, Qwen2.5-VL produces the following caption, used by all baselines on this scene:

> *The video captures a scene set in an urban area with a paved walkway and a white car parked nearby. The camera is positioned at a high angle, providing a bird's-eye view of the scene. Three individuals are walking together; one person is dressed in black, another in a dark shirt and light-colored pants, and the third in a patterned outfit. They appear to be engaged in conversation as they walk side by side. The lighting suggests it might be daytime, with natural light illuminating the area. There is minimal camera movement, maintaining a steady focus on the group as they move across the frame.*

This caption conveys properties that require observing the full clip: the "walking together" group behaviour, the per-pedestrian clothing details, and scene elements such as the parked white car. For the same clip, our method's prompt is the template instantiation

> *An outdoor scene with walking figures, where pedestrian walks [traj$_0$], pedestrian strolls [traj$_1$], pedestrian moves [traj$_2$].*

which conveys no scene, appearance, or interaction context beyond the first-frame VAE latent consumed by the appearance encoder.

F.5 WaN Baseline Frame Count

ATI and Wan-Move are trained on 81-frame sequences, while our evaluation uses 49 frames. To reconcile this mismatch, we evaluate both baselines under two protocols: (1) generating 49 frames directly by patching a single hardcoded frame-count constant in each codebase, and (2) generating 81 frames at the native setting with trajectories padded to 81 frames by repeating the last position, then trimming the output to the first 49 frames for metrics. All other settings (checkpoints, seeds, captions, resolution) are identical between the two protocols.

Table S4 reports both variants across all six evaluation datasets. The results show no systematic advantage to either protocol: the 49-frame variant wins on PSNR on 5 of 6 datasets for both baselines, while the 81-frame variant is occasionally better on LPIPS, FVD, or EPE. Differences are generally small (PSNR within 1 dB, LPIPS within 0.03). We report the 49-frame results in the main tables.

Table S4: WaN-backbone baselines: 49-frame generation (reported in main tables) vs. native 81-frame generation trimmed to 49 frames. Trajectories for 81-frame runs are padded by repeating the last position. All other settings identical. Best per cell in **bold**.

			MoVi				**Pool**				**Football**			
	Method	Frames	PSNR↑	LPIPS↓	FVD↓	EPE↓	PSNR↑	LPIPS↓	FVD↓	EPE↓	PSNR↑	LPIPS↓	FVD↓	EPE↓
ATI		49	**20.33**	.355	37.50	**2.01**	**21.13**	.303	57.78	**0.37**	**19.68**	**.250**	46.32	**1.33**
		81	20.02	**.313**	**36.20**	2.41	20.78	**.284**	**51.62**	0.46	19.57	.259	**44.47**	1.73
WM		49	**19.73**	**.331**	40.38	**2.18**	**22.94**	.191	**33.98**	0.29	**18.36**	**.296**	69.84	**3.81**
		81	19.17	.334	**39.82**	2.65	22.05	**.186**	38.10	**0.27**	17.94	.311	**68.80**	4.57

			MOTSynth				**MOT17**				**DAVIS**			
	Method	Frames	PSNR↑	LPIPS↓	FVD↓	EPE↓	PSNR↑	LPIPS↓	FVD↓	EPE↓	PSNR↑	LPIPS↓	FVD↓	EPE↓
ATI		49	**17.76**	**.395**	**62.32**	**8.61**	**17.05**	.416	**65.35**	**4.28**	14.68	.443	**104.90**	14.02
		81	17.32	.413	67.13	9.88	16.70	**.406**	70.33	4.73	**14.96**	**.425**	105.71	**11.15**
WM		49	**18.29**	**.354**	**54.29**	**3.60**	16.27	.405	60.20	3.41	**15.86**	**.352**	**85.51**	4.96
		81	17.78	.383	61.28	3.82	**16.55**	**.379**	**58.70**	**3.33**	15.45	.355	88.15	**4.22**

G MOTSynth-Multi

Construction MOTSynth-Multi (Sec. 4.3) uses the synthetically trained models of Sec. 4 and is drawn exclusively from held-out MOTSynth test scenes, none of which is used for training. Each bin contains 20 clips from 20 distinct scenes, with exactly 5, 10, or 15 controlled pedestrians, or 17–20 controlled pedestrians in the densest bin, where the number of eligible held-out scenes is the limiting factor. Pedestrians are counted within the 720×480 crop, and every controlled pedestrian is visible in the first frame. Clips are prepared and captioned exactly as the MOTSynth evaluation set of Sec. 4.1.

Results Table 3 reports all four metrics. EPE decreases for every method as the count grows, because pedestrians in dense crowds move less, which likely also contributes to the rising PSNR of the baselines. With CogVideoX, MagicMotion attains a lower FVD in the densest bin and a lower EPE in the sparsest. Figures S12 and S13 show scenes from the two densest bins.

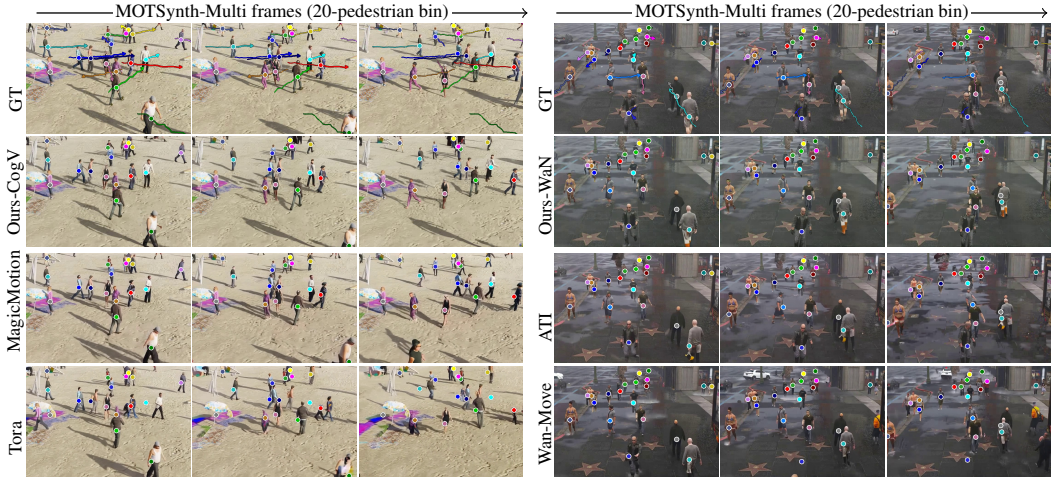

Figure S12: **Qualitative comparison on the 20-pedestrian bin of MOTSynth-Multi with all baselines** (18 and 20 controlled pedestrians). Left: CogVideoX-5B against MagicMotion and Tora. Right: WaN 2.1-14B against ATI and Wan-Move. MagicMotion inserts a pedestrian that is not in the scene at the bottom of the last frame, and Tora loses several pedestrians, for example at the gray, teal and cyan dots. Wan-Move drops the blue-dot pedestrian in the foreground and inserts a pedestrian in an orange jacket on the right, and ATI produces a corrupted region at the top of the last frame. Our method keeps every pedestrian on its prescribed trajectory with both backbones.

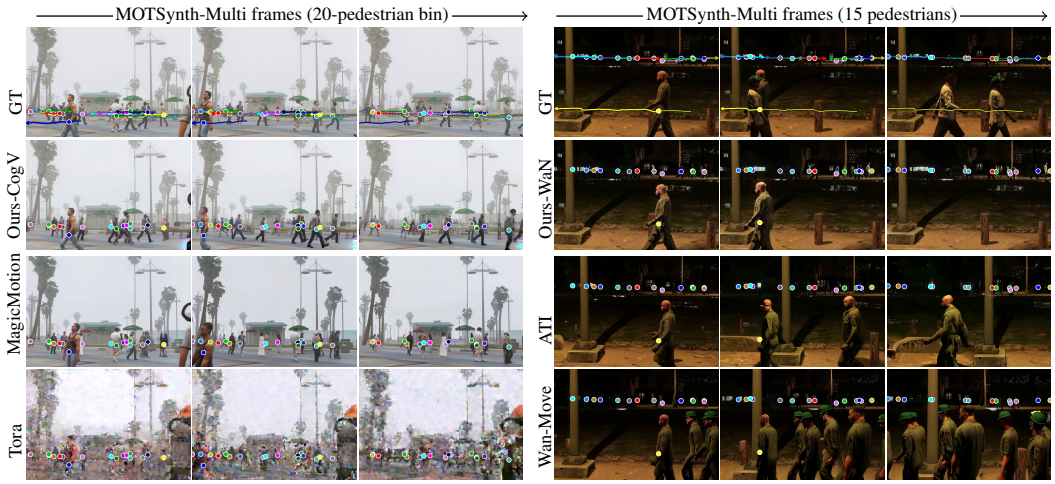

Figure S13: **Additional qualitative comparison on MOTSynth-Multi.** Left: CogVideoX-5B on a scene from the 20-pedestrian bin (19 pedestrians). Right: WaN 2.1-14B on a 15-pedestrian scene. Tora degrades the image quality of the crowded promenade, while MagicMotion and our method preserve it. On the right, the tracked pedestrian in the foreground walks out of the frame; ATI keeps him in view, and Wan-Move fills the scene with copies of him that do not correspond to any trajectory, while our method lets him exit on his prescribed path.

H Extended Ablations

H.1 Additive and Leave-One-Out Ablation on Pool and Football

Table S5 completes Table 5 with the two remaining in-distribution datasets. On Pool, the full model again achieves the best PSNR and EPE, and the trajectory token is unusually informative: on its own it already reaches 31.60 dB, since billiard motion is fully described by the image-plane trajectories of a few balls. On Football, every configuration with attention localization lies within 1.5 dB PSNR of the others and the bare backbone already reaches an EPE of 0.05, so this dataset discriminates little between the components, and attention localization alone gives the best LPIPS and FVD.

Table S5: **Additive and leave-one-out ablation on Pool and Football** (WaN 2.1-14B, configurations as in Table 5). Best in bold.

						Pool				**Football**			
#	Configuration	A	T	P	d	PSNR↑	LPIPS↓	FVD↓	EPE↓	PSNR↑	LPIPS↓	FVD↓	EPE↓
1	Backbone only					27.01	.158	48.66	0.66	23.55	.145	36.22	0.05
2	Attention only	✓				29.69	.065	43.92	0.09	24.66	**.133**	**22.12**	**0.04**
3	w/o trajectory token	✓		✓		29.20	.101	45.91	0.11	23.33	.143	31.03	0.05
4	Trajectory token only		✓		✓	31.60	.075	**27.75**	0.10	23.78	.159	31.64	0.06
5	w/o attention localization		✓	✓	✓	29.46	.113	37.47	0.17	23.02	.192	58.19	0.15
6	w/o appearance token	✓	✓		✓	32.45	.057	38.36	**0.08**	**24.76**	.143	26.02	**0.04**
7	w/o depth channel	✓	✓	✓		32.37	.054	38.25	0.09	24.32	.146	27.67	0.06
8	**Ours**	✓	✓	✓	✓	**33.07**	**.049**	31.34	**0.08**	24.27	.141	24.24	0.05

H.2 Design Choices

Table S6 ablates two design choices of the full model. First, we drop the bounding-box loss of Eq. 5 by setting α=0. This lowers PSNR by 0.2–1.0 dB and increases FVD and EPE on MoVi, MOTSynth, and Pool. On Football, where all configurations perform similarly (Appendix H.1), PSNR changes by less than 0.2 dB. Second, we insert the trajectory token after the text encoder, where the appearance token is inserted, instead of before it. This degrades every dataset sharply, for example by 4.6 dB PSNR on MoVi and 9.8 dB on Pool, with FVD two to four times higher. The trajectory encoder is pretrained so that its embeddings remain informative after passing through the text encoder (Sec. 3.2.1), so the token must be inserted before it.

Table S6: **Design choices** (WaN 2.1-14B) on the four in-distribution datasets. Best per dataset in bold.

	MoVi				**MOTSynth**			
Configuration	PSNR↑	LPIPS↓	FVD↓	EPE↓	PSNR↑	LPIPS↓	FVD↓	EPE↓
Bounding-box loss α=0	28.21	.147	29.23	0.63	21.06	.252	45.54	2.34
Trajectory token after text encoder	24.60	.242	76.88	1.69	17.39	.534	132.4	3.37
Ours	**29.24**	**.131**	**23.39**	**0.57**	**21.39**	**.244**	**41.43**	**2.11**
	Pool				**Football**			
Configuration	PSNR↑	LPIPS↓	FVD↓	EPE↓	PSNR↑	LPIPS↓	FVD↓	EPE↓
Bounding-box loss α=0	32.86	**.046**	32.70	0.09	**24.44**	**.136**	29.04	**0.04**
Trajectory token after text encoder	23.27	.159	91.23	0.22	17.86	.358	97.82	0.33
Ours	**33.07**	.049	**31.34**	**0.08**	24.27	.141	**24.24**	0.05

H.3 Seed Variance of the Leave-One-Out Ablation

Each cell of Table 5 is a single-seed estimate. To verify that the component effects exceed sampling noise, we repeat the leave-one-out configurations on MoVi and MOTSynth with nine seeds, the reported seed and eight new ones, changing only the initial noise (Table S7). The reported single-seed

values lie within 2.1 standard deviations of the nine-seed means. On MoVi, every component effect exceeds the seed standard deviation at least fivefold on every metric. Even the smallest PSNR gap, from removing the appearance token, is 0.75 dB against a standard deviation of at most 0.10 dB. On MOTSynth, removing attention localization or the appearance token lowers PSNR well beyond seed noise, by 2.1 and 0.7 dB. Removing the trajectory token or the depth channel shows mainly in EPE, which rises from 2.15 to 2.49 and 2.32, consistent with their role of encoding motion and depth. The PSNR effect of the trajectory token on MOTSynth (0.08 dB) is within seed noise.

Table S7: **Leave-one-out ablation over nine seeds** (WaN 2.1-14B), mean ± standard deviation. Best in bold.

	MoVi				**MOTSynth**			
Configuration	PSNR↑	LPIPS↓	FVD↓	EPE↓	PSNR↑	LPIPS↓	FVD↓	EPE↓
Ours	**29.31**±.10	**.130**±.001	**23.47**±.26	**0.56**±.01	**21.46**±.08	**.244**±.002	**41.03**±.69	**2.15**±.05
w/o trajectory token	27.92±.07	.144±.002	26.65±.42	0.66±.01	21.38±.06	.254±.002	42.58±.71	2.49±.10
w/o appearance token	28.56±.09	.144±.001	27.87±.40	0.66±.01	20.78±.09	.258±.002	45.34±.52	2.19±.05
w/o depth channel	27.52±.06	.145±.001	27.84±.46	0.63±.01	21.34±.05	.250±.002	43.09±.46	2.32±.10
w/o attention localization	22.42±.07	.318±.003	40.55±1.1	1.95±.05	19.35±.09	.317±.004	56.01±.80	3.27±.14

H.4 Depth as an Optional Input

In our main evaluation, the depth channel $d_i(t)$ is derived from the ground-truth clip, as during training (Sec. 4.1). To quantify how much the method relies on it, Table S8 compares each model with the same model whose trajectory tokens are re-encoded with depth set to zero, on both backbones and all six datasets. Removing depth costs 1–1.7 dB PSNR on MoVi and MOT17. On Football, MOTSynth, and DAVIS the two variants are within 0.2 dB PSNR, and with CogVideoX the depth-free variant even attains a slightly lower EPE on several datasets. Depth is therefore beneficial but optional, and our real-data models of Appendix A use no depth at all.

Table S8: **Effect of the depth channel** at the checkpoints of the main tables. Better value per dataset and backbone in bold.

		WaN 14B				**CogV**			
Dataset	Depth	PSNR↑	LPIPS↓	FVD↓	EPE↓	PSNR↑	LPIPS↓	FVD↓	EPE↓
MoVi-Extended	with	**29.24**	**.131**	**23.39**	**0.57**	**29.77**	**.108**	**14.98**	0.53
	without	27.56	.145	27.66	0.62	28.84	.118	15.07	**0.52**
Pool	with	**33.07**	**.049**	**31.34**	**0.08**	**35.64**	**.025**	**8.79**	0.07
	without	32.37	.054	38.25	0.09	30.43	.035	11.14	**0.06**
Football	with	24.27	**.141**	**24.24**	**0.05**	24.79	**.090**	13.87	**0.03**
	without	**24.32**	.146	27.67	0.06	**24.85**	.095	**12.83**	**0.03**
MOTSynth	with	**21.39**	**.244**	**41.43**	**2.11**	22.34	.232	**39.70**	2.32
	without	21.32	.250	43.11	2.28	**22.50**	**.230**	40.51	**2.21**
MOT17	with	**19.94**	**.277**	**51.46**	**2.17**	**21.59**	**.265**	**53.92**	2.37
	without	18.96	.299	56.21	2.61	20.58	.270	54.73	**2.16**
DAVIS	with	**18.30**	**.301**	88.37	2.31	18.39	.267	**89.39**	2.43
	without	18.25	**.301**	**88.27**	**2.30**	**18.42**	**.266**	92.35	**2.40**

I Attention Localization Step by Step

Algorithm 1 details the trajectory heatmap construction and the column replacement of Sec. 3.1 for explicit cross-attention. Its only inputs beyond the attention tensors are the prescribed trajectories and the position of each object token in the prompt: the column index q_i is the position of the category token o_i in the tokenized prompt of Eq. 1, and the initial location (x_i^1, y_i^1) used by the appearance encoder is the first entry of τ_i. Because σ is defined in pixels, each Gaussian is rendered at video resolution and then resampled to the latent grid. The heatmaps are computed once per sample and shared by all heads, layers, and denoising steps. Each object occupies a single, independent column, so the cost of the replacement grows with the number of objects rather than with the video resolution. For joint self-attention, the column replacement is computed with the two-SDPA formulation of Eq. 11.

Algorithm 1 Attention localization at one cross-attention layer

Require: trajectories $\{(x_i(t), y_i(t))\}_{t=1}^{T}$ with per-frame visibility, for $i = 1, \dots, N$
Require: object-token columns $\{q_i\}_{i=1}^{N}$; standard deviation σ (pixels)
Require: queries $\mathbf{Q}_{\text{vid}} \in \mathbb{R}^{M \times d}$; text keys and values $\mathbf{K}_{\text{txt}}, \mathbf{V}_{\text{txt}} \in \mathbb{R}^{L \times d}$
Ensure: localized attention output $\mathbf{O}' \in \mathbb{R}^{M \times d}$

Heatmap construction ▷ once per sample

for $i = 1$ **to** N **do**
  for each frame t in which object i is visible **do**
    render g_i^t, an isotropic Gaussian of std. σ at $\big(x_i(t)\, W_p,\ y_i(t)\, H_p\big)$ on the $H_p \times W_p$ grid
    resize g_i^t bilinearly to the $h \times w$ latent grid and normalize it to sum to one
  end for
  interpolate $\{g_i^t\}$ linearly from T video frames to F latent frames; renormalize each frame
  set the latent frames in which object i is absent to zero
  $\mathbf{h}_i \leftarrow \mathrm{vec}\big(\{g_i^f\}_{f=1}^{F}\big) \in \mathbb{R}^{M}, \quad M = Fhw$
end for

Column replacement ▷ every layer, head, and denoising step

$\mathbf{A} \leftarrow \mathrm{softmax}\big(d^{-1/2}\, \mathbf{Q}_{\text{vid}} \mathbf{K}_{\text{txt}}^{\top}\big) \in \mathbb{R}^{M \times L}$
for $i = 1$ **to** N **do**
  $\mathbf{A}[:, q_i] \leftarrow \mathbf{h}_i$
end for
$\mathbf{A}[m, :] \leftarrow \mathbf{A}[m, :] \,/\, \sum_{k=1}^{L} \mathbf{A}[m, k]$ for $m = 1, \dots, M$ ▷ Eq. 3
return $\mathbf{O}' \leftarrow \mathbf{A}\mathbf{V}_{\text{txt}}$

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 describe the task, limitations of existing methods, and our contributions; these claims are supported by the quantitative and qualitative results reported in Section 4.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss limitations in Section 5 (Limitations and Broader Impacts) and failure cases in Appendix C.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: We disclose all experimental information in Section 4 (Implementation details, Training data, Evaluation data) and in the supplementary material (encoder architectures and pretraining in Appendices E.1 and E.2; diffusion fine-tuning hyperparameters and training budget in Appendix E.3; training data composition in Appendix F.1; baseline inference settings in Appendix F.2; DAVIS preprocessing in Appendix F.3; and per-video captioning in Appendix F.4).

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: Code is available at `https://github.com/Sela-Omer/traj-loc`, together with the checkpoints of the real-data models of Appendix A. The checkpoints of the synthetically trained models of Section 4 are not released; these models can be retrained from the publicly available datasets with the procedure and hyperparameters of Section 4 and Appendices E–F. All datasets used are publicly available and their preprocessing is described in Section 4 and the appendix.

   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: We specify training and test details in the Implementation details paragraph of Section 4 and in the supplementary material (trajectory and appearance encoder details in Appendices E.1 and E.2; diffusion fine-tuning hyperparameters and training budget in Appendix E.3; baseline inference settings in Appendix F.2; evaluation protocol in Appendix F; and the WaN frame-count reconciliation in Appendix F.5).

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

Justification: The main comparison tables report single runs, as is common for motion-controllable video generation. We quantify sampling variation for the ablation study, reporting the mean and standard deviation over nine seeds in Appendix H.3.

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: We provide compute details (GPU type, effective batch size, and wall-clock training time for both backbones) in Appendix E.3, and for the real-data models in Appendix A.1.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have carefully reviewed the NeurIPS Code of Ethics to ensure that our submission complies with all regulations.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss broader impacts in Section 5 (Limitations and Broader Impacts), covering positive uses (synthetic data, simulation, professional video editing) and dual-use risks (misleading content), with watermarking and detection tools suggested as mitigations.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release a new high-risk asset: we fine-tune existing pretrained I2V backbones (CogVideoX-5B, WaN 2.1-14B) via LoRA and small encoder modules, and do not introduce new datasets or scraped data. The dual-use risks of realistic video generation are inherited from the underlying I2V models rather than introduced by our method; we discuss these risks and recommended mitigations (watermarking, detection tools for AI-generated videos) in Section 5. We therefore do not propose additional release-time safeguards beyond those already applicable to the base I2V models.

    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: The models and data used in this paper are open-sourced and cited, and we use them in accordance with their respective licenses.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We release code and LoRA and encoder checkpoints for the publicly available CogVideoX-5B and WaN 2.1-14B backbones. Their usage, training data, and limitations are documented in the repository and in Appendix A, and their use is subject to the licenses of the underlying backbones and datasets.

    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: 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.

- 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: Our core method does not involve LLMs as an important or non-standard component.

    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.