# TRACE: OBJECT MOTION EDITING IN VIDEOS WITH FIRST-FRAME TRAJECTORY GUIDANCE

**Anonymous authors**
Paper under double-blind review




Figure 1: **TRACE** enables intuitive object-centric motion editing in videos. **Left:** Given an input video (e.g., a girl flying a kite), the user specifies a desired 2D trajectory in the first-frame view using sparse bounding boxes. Two editing examples (Path Edit I and II) are shown. The solid yellow box denotes the initial bounding box, the dashed yellow box denotes the final bounding box, and the yellow curve represents the user-defined motion path. Our method re-synthesizes the video such that the object follows the new trajectory while preserving the remaining scene content. **Right:** Additional examples demonstrate robust motion editing under diverse camera movements, where TRACE transforms the first-frame path design into temporally consistent object motion in the generated videos. Please find associated videos in Supplementary Material.

## ABSTRACT

We study object motion editing in videos, where the goal is to change a target object's trajectory while preserving the remaining content of the original video. Many existing trajectory-controlled video editing methods require users to specify frame-wise control signals, such as dense point tracks or bounding boxes, making motion design cumbersome and error-prone, particularly in videos with camera motion. We introduce **TRACE**, a framework that allows users to specify the desired motion path in the first frame alone. The key challenge is that a path defined in the first-frame coordinate system does not directly determine the object's image-space trajectory in subsequent frames as the camera moves. To address this mismatch, TRACE employs a two-stage pipeline. First, a cross-view motion transformation module converts the first-frame path into frame-aligned bounding-box trajectories. Second, a motion-conditioned video re-synthesis module removes the object from its original trajectory and regenerates it along the transformed trajectory while preserving non-target video content. Experiments on diverse real-world videos show that TRACE produces more coherent and controllable motion edits than recent image-to-video and video-to-video methods.

## 1 INTRODUCTION

Generative video modeling has facilitated rapid advances in video content creation Singer et al. (2022); Chen et al. (2023; 2024); Polyak et al. (2024); He et al. (2022); Wan et al. (2025); Ge et al. (2023); Yang et al. (2024). Modern video editing systems excel at transforming a video's appearance, like style transfers Wang et al. (2023); Yang et al. (2023); Chung et al. (2024); Bai et al. (2025a) or local edits Bian et al. (2025); Truong et al. (2024); Gu et al. (2024). However, many editing scenarios require changing how a particular object moves while preserving the remainder of



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

the captured video. For example, a creative director may be satisfied with a dynamic scene of a child running with a kite, but wish to change the kite's direction, speed, or timing to achieve a different creative intent (Fig. 1).

Existing object re-synthesis methods typically rely on dense, frame-aligned spatiotemporal control signals, such as mask sequences or per-frame bounding boxes Jiang et al. (2025); Hu et al. (2025); Yang et al. (2025). Although effective as model inputs, these controls require users to specify where the object should appear throughout the video at inference. This process is cumbersome and becomes particularly difficult under dynamic camera motion because the user must manually compensate for viewpoint changes when determining the object's position in each frame.

Inspired by previous video appearance editing works that highlight the user-friendliness of applying edits to an anchor frame and automatically propagating them temporally Liu et al. (2025a); Schneider et al. (2025), we introduce a novel problem setting: *Video Editing with First-Frame Object Motion Design*. The user specifies the desired motion path using sparse bounding boxes in the first frame, describing how the object should move if the camera remained fixed at the initial viewpoint. The first frame therefore acts as a scene-aware canvas, allowing the user to design a complete trajectory relative to visible scene landmarks without annotating subsequent frames.

This interaction introduces a fundamental challenge of cross-view spatial misalignment. A path defined in the first-frame coordinate system does not directly determine the object's image-space trajectory in subsequent frames as the camera moves. We present **TRACE**, a two-stage framework that addresses this mismatch. First, a learnable cross-view motion transformation module converts the first-frame path into a frame-aligned bounding-box trajectory in the input video's dynamic view. Second, a motion-conditioned video re-synthesis module removes the object from its original trajectory and regenerates it along the transformed target trajectory while preserving the rest of the video. A key benefit of this decomposition is that the video re-synthesis module can be trained without paired source and motion-edited target videos.

We validate TRACE's effectiveness on real-world videos with diverse content and motion-editing scenarios, demonstrating high-quality resynthesis and flexible user controls. Our experiments show that TRACE can facilitate flexible manipulation of object motion that was not possible with conventional video editing models. In summary, our work makes the following contributions:

1. We formulate video editing with first-frame object motion design, an interaction paradigm that allows users to specify a selected object's complete target path in a single reference frame rather than through frame-wise annotations.
2. We introduce a learnable cross-view motion transformation module that converts a fixed-view motion design into frame-aligned object placements under camera motion.
3. We develop a motion-conditioned video re-synthesis model that jointly removes the object from its original trajectory and regenerates it along a target trajectory. Our training strategy does not require paired source and motion-edited target videos.

## 2 RELATED WORK

**Motion-controlled video synthesis.** Recent video generation methods provide object-motion control through point trajectories Xiao et al. (2024); Wang et al. (2025); Qiu et al. (2024); Wang et al. (2024b); Xing et al. (2025) or bounding boxes Li et al. (2023); Huang et al. (2025); Wang et al. (2024a). These methods are primarily designed for text-to-video or image-to-video synthesis and generally assume that the control signals are already aligned with each output frame. They are therefore not directly suited to editing a complete input video while preserving its camera motion, non-target dynamics, and scene content. In contrast, TRACE accepts a complete target path specified in the first-frame coordinate system and transforms it into frame-aligned control for video re-synthesis.

**Video editing and re-synthesis.** Video editing methods support global appearance transformations Wang et al. (2023); Yang et al. (2023); Chung et al. (2024); Bai et al. (2025a) and localized content or attribute edits Truong et al. (2024); Chen et al. (2025); Qi et al. (2023); Xu et al. (2025); Zhang et al. (2026). Most of them modify appearance while preserving the motion of the source video. General conditional video models and compositing methods, such as VACE, HunyuanCustom, and GenCompositor Jiang et al. (2025); Hu et al. (2025); Yang et al. (2025), support masks or spatial

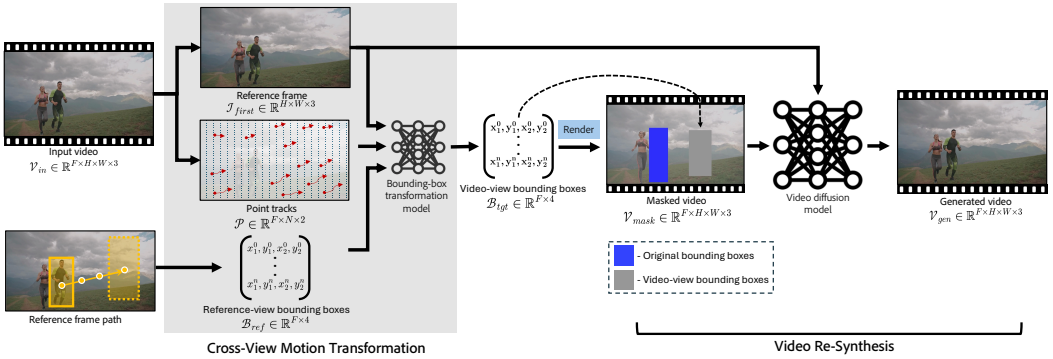



Figure 2: **TRACE Overview.** Our method consists of two modules. First, the **Cross-View Motion Transformation Module** converts a user-drawn path in the first frame into a scene- and camera-aware bounding-box trajectory in the video view. Second, the **Video Re-Synthesis Module** uses this transformed box sequence to guide generation so the object follows the desired path while inpainting the object's original path. Both models in the figure are diffusion models.

controls, but typically require frame-wise conditions or focus on inserting external content. Our work instead changes the trajectory of an object already present in the source video. Moreover, our video re-synthesis module is trained by reconstructing from automatically derived trajectory and masking conditions, without requiring paired source and motion-edited target videos.

**Video motion editing and first-frame interaction.** Recent methods including ReVideo, Shape-for-Motion, and Edit-by-Track Mou et al. (2024); Liu et al. (2025b); Lee et al. (2025) also edit object motion through trajectory-conditioned video reconstruction. However, these approaches require temporally distributed trajectory controls, dense point tracks, or edits derived from the object's observed motion. Separately, first-frame video editing methods propagate appearance edits from a single anchor frame Liu et al. (2025a); Schneider et al. (2025), reducing the need for temporally consistent user annotations. TRACE combines this single-frame interaction with motion editing: the user designs the complete target path in the first frame, and a learned cross-view transformation converts it into frame-aligned object placements under camera motion.

## 3 METHOD

Our system re-synthesizes an input video using a video diffusion model to move an object along a new motion path, while preserving the other aspects of the input video. The model generates the final video conditioned on two inputs: *(i)* the original video, and *(ii)* the user-specified motion path, parameterized as a sparse bounding box sequence defined on the first frame of the video. Our method TRACE (Fig. 2) consists of two main components: a *Cross-View Motion Transformation* module and a *Video Re-Synthesis* module. The former translates the first-frame-based motion path, which depicts the user's high-level motion intent, into the corresponding boxes for each video frame, accounting for the original video's camera motion. The latter, the re-synthesis model, faithfully renders the object along this path.

### 3.1 CROSS-VIEW MOTION TRANSFORMATION MODULE

The objective of this module is to map a user-defined motion path from the first-frame view to a temporally consistent target path in the video's per-frame view that aligns with the video's dynamic camera perspective. As illustrated in Fig. 4, naive linear interpolation fails to model dynamic motion and camera changes, often producing trajectories that drift away from the intended target (e.g., last box on the right of the red coral anchor). State-of-the-art depth and pose estimators, such as Depth Anything 3 (DA3) Lin et al. (2025) and MegaSAM Li et al. (2025b), attempt to reconstruct scene geometry, which can be used to warp the boxes, but errors in camera pose and depth estimation result in erratic, "jumpy" trajectories and inconsistent object scales due to accumulated noise. Thus, we propose bypassing explicit 3D reconstruction and treating the task as a sequence-to-sequence mapping problem, in which our model learns to implicitly model camera dynamics.

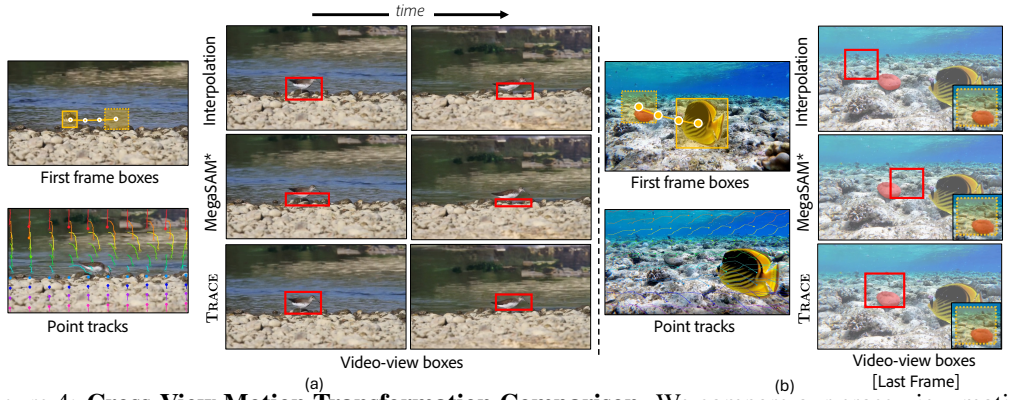



Figure 4: **Cross-View Motion Transformation Comparison.** We compare our cross-view motion transformation against three baselines: (1) simple **Interpolation** of bounding boxes, (2) 3D warping using MegaSAM Li et al. (2025b) to estimate depth and camera pose. **Left:** Shows the first-frame path and generated video-view bounding boxes. Only TRACE accurately translates the first-frame view bounding boxes to the video view. **Right:** The user's intended path moves the fish towards the static red coral (used as an anchor). Interpolation produces an off-track path to the right. Existing 3D warping methods yield incorrect paths due to noisy depth and camera estimation. TRACE generates an accurate path that moves the fish box to the coral.

**Model architecture and training.** Our model $\phi_C$ predicts the frame-aligned target bounding-box sequence $\mathcal{B}_{tgt}$ from three conditions: *(i)* the reference trajectory $\mathcal{B}_{ref}$ specified in the first-frame coordinate system, *(ii)* the first-frame image $\mathcal{J}_{first}$, and *(iii)* sparse point trajectories $\mathcal{P}$ extracted from the input video, which provide cues about camera motion. We represent each point trajectory compactly using discrete cosine transform coefficients and treat the resulting representations as conditioning tokens. To support flexible user inputs, we temporally interpolate sparse key bounding boxes into the dense reference sequence $\mathcal{B}_{ref}$ required by the model. The model is trained on paired reference-view and dynamic-view box sequences to generate $\mathcal{B}_{tgt}$ from noisy box tokens. More details are provided in the Appendix A.3. $\phi_C$ is trained with flow-matching *v-pred* Lipman et al. (2022):

$$\min_{\phi_C} \mathbb{E}_{t,x_t} \left[ \| v - \phi_C(x_t, t \mid \mathcal{J}_{first}, \mathcal{P}, \mathcal{B}_{ref}, \mathcal{J}_{first}) \|_2^2 \right].$$

**Data curation**. Collecting paired data of object boxes in the first-frame view and a dynamic video sequence in the real world is challenging because it requires capturing the same video with both a static camera (to obtain first-frame boxes) and a dynamic camera (to obtain corresponding video boxes). Instead, we design a synthetic data pipeline that produces paired static and dynamic videos, enabling automatic construction of aligned bounding boxes for training. We collect 7,500 videos captured with a static camera and extract the first-frame view bounding boxes, $\mathcal{B}_{ref}$. Next, we use ReCamMaster Bai et al. (2025b) to re-render each video with 10 different dynamic camera paths and extract video-view bounding boxes $\mathcal{B}_{tgt}$, resulting in 75k synthesized videos and over 150K paired bounding box sequences. Then, we use CoTracker Karaev et al. (2024) to extract a grid of $25 \times 25$ point trajectories. Finally, we filter the data based on motion dynamics (e.g., disproportionate number of static objects in static videos) and object size, yielding approximately 110k high-quality pairs. (see the Appendix A.6 for details

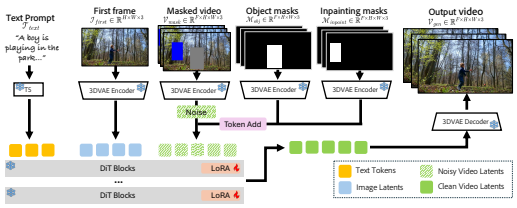



Figure 3: **Video Re-Synthesis Module.** A DiT backbone is conditioned on the first frame, masked video, object and inpainting masks.

## 3.2 VIDEO RE-SYNTHESIS MODULE

The goal of the Video Re-Synthesis Module is to edit an input video $\mathcal{V}_{in}$ so that a selected object follows the target motion boxes $\mathcal{B}_{tgt}$ predicted by the Cross-View Motion Transformation Module. Let $\mathcal{B}_{src}$ denote the object's original boxes in input video $\mathcal{V}_{in}$. To make the conditioning unambiguous in both training and inference, we denote the two color-coded binary masks by $\mathcal{M}_{gray}$ and $\mathcal{M}_{blue}$, corresponding to pixels rendered gray and blue, respectively, in the masked video $\mathcal{V}_{mask}$. Both masks are also supplied to the model as separate binary conditions. **During training**, $\mathcal{M}_{gray}$ cov-

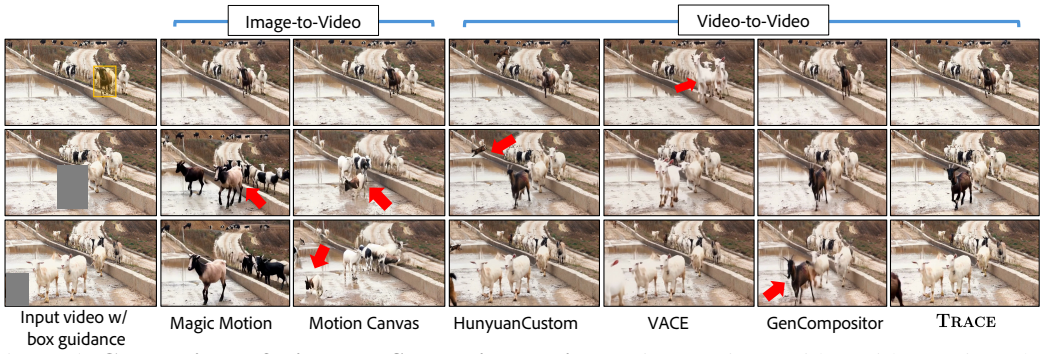



Figure 5: **Comparison of Video Re-Synthesis Baselines** .Given an input video with a region (the original object) masked out and conditioned on the appearance of a reference object, the model must regenerate the object within the masked region. The goal is to produce a high-fidelity video that accurately restores the object while maintaining its original identity and temporal consistency.

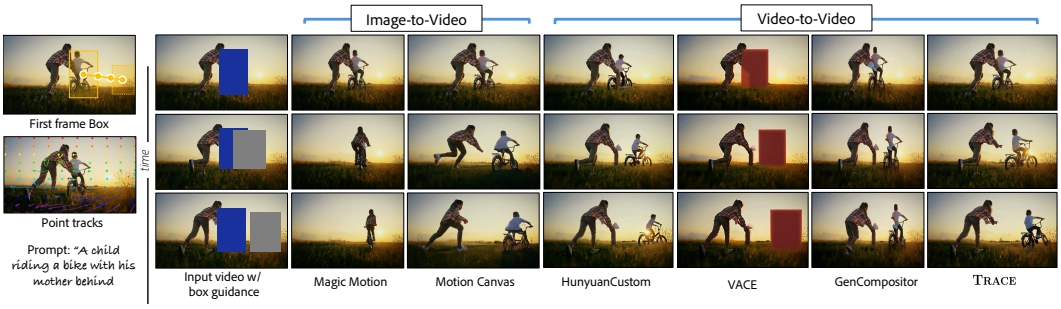



Figure 6: **Full pipeline comparison.** Our Cross-View Motion Transformation module converts a user's path (defined by the first and last boxes in the first frame) into a sequence of video boxes. This sequence is then used to guide our re-synthesis model and three baselines (which first require a separate inpainting step). For Image-to-Video Baselines, we use only a sequence of video boxes and background point tracks (for Motion Canvas only).

ers the tracked object region in $\mathcal{V}_{in}$, while $\mathcal{M}_{blue}$ is an augmented region sampled for background synthesis (see the Appendix A.4 for details). We construct $\mathcal{V}_{mask}$ by painting these regions with their corresponding colors and train the model to reconstruct the original video $\mathcal{V}_{in}$. This teaches the gray condition to synthesize the object and the blue condition to restore background content. If the two regions overlap, the gray region takes precedence so that object synthesis is not overwritten by background restoration. **At inference**, the regions are assigned according to the editing task: $\mathcal{M}_{blue}$ is constructed from the object's original boxes $\mathcal{B}_{src}$ to remove the original object and restore the revealed background, whereas $\mathcal{M}_{gray}$ is constructed from target motion boxes $\mathcal{B}_{tgt}$ and serves as box guidance for generating the object at its new location. The resulting masked video $\mathcal{V}_{mask}$, both binary masks, and the first frame $\mathcal{F}_{first}$ are provided to the model to generate $\mathcal{V}_{gen}$. Fig. **??** shows the data processing for both training and inference.

**Model architecture and training**. We adapt the Wan 2.1 14B model Wan et al. (2025), $\phi_V$, to predict $\mathcal{V}_{gen}$ using $\mathbf{C} = \{\mathcal{T}_{text}, \mathcal{F}_{first}, \mathcal{V}_{mask}, \mathcal{M}_{gray}, \mathcal{M}_{blue}\}$ as conditioning inputs. During training, the model reconstructs $\mathcal{V}_{in}$; at inference, the same interface produces the edited video from the inference-time masks described above. We fine-tune the model with LoRA using the flow-matching *v-pred* objective Lipman et al. (2022):

$$\min_{\phi_V} \mathbb{E}_{t,x_t}\left[\|v - \phi_V(x_t, t \mid \mathbf{C})\|_2^2\right],$$

where $t$ is the flow time, $x_t$ is the noisy interpolated latent, and $v$ is the target velocity. More details are provided in the supplementary material.

**Dataset curation**. We use approximately 1.1M videos from an internal dataset for training, of which 80% are long-shot videos better suited to the motion editing task. For each video, we extract object bounding boxes using DEVA Cheng et al. (2023). To construct the inpainting condition, we create a perturbed copy of the object-mask trajectory by applying random spatial offsets and additional frame-wise jitter to its bounding boxes. This teaches the model to restore the underlying scene content, which is exactly the inpainting task. Thus, both object generation and background inpainting

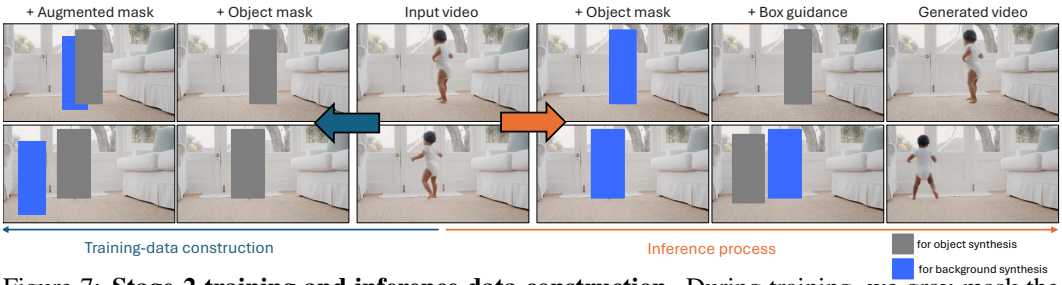



Figure 7: **Stage-2 training and inference data construction.** During training, we gray-mask the object, add an augmented blue mask for background synthesis, and feed the masked video and both masks to the model to reconstruct the original video. At inference, these colors switch roles: blue marks the original object to remove, whereas gray box guidance specifies where to generate it. (The same video is shown only for illustration; the training and test sets are disjoint.)

are learned from individual unedited videos, without separately collected paired supervision. The training-data construction process is illustrated in Fig. 7. We also smooth the box guidance and augment the training data to improve robustness; details are provided in Appendix A.4

### 3.3 Applications

We demonstrate that TRACE supports a wide range of versatile video editing tasks during inference: *(a) Multi-object editing*. Users can specify new trajectories for multiple objects simultaneously on the first frame (Fig. 9). *(b) Object insertion*. TRACE enables object insertion with precise motion control: the user defines a target layout and path using bounding boxes in the first frame. We utilize Qwen Wu et al. (2025) to inpaint the new object only in the first frame, and propagate the synthesized object along the designed trajectory (Fig. 8 top). *(c) Object editing*. Our model can propagate localized appearance edits by modifying the object's attributes (e.g., color, textures) in the first frame. (Fig. 8 bottom-left). *(d) Object replacement*. Users can replace a selected object with an entirely different object in the first frame, and then TRACE can re-synthesize the video to ensure the new object maintains the original motion path (Fig. 8 bottom-right).(Refer to Appendix A.1 for full application details)

## 4 Experiments

In this section, we evaluate the effectiveness of our method on: 1) cross-view motion transformation, 2) video re-synthesis, and 3) joint end-to-end performance.

### 4.1 Cross-View Motion Transformation Module

**Training details.** For the Cross-View Motion Transformation Module, we train an 8-layer DiT conditioned on the interpolated first-frame box sequence, the 3D-VAE-encoded first frame, and a $25 \times 25$ grid of CoTracker point trajectories represented by their first $K = 20$ DCT coefficients. The model is optimized for 180K steps using a flow-matching $v$-prediction objective, with a learning rate of $1.2 \times 10^{-4}$ and weight decay of $0.01$. For Stage I, we split the 7,500 curated videos into 7,400 training videos and 100 evaluation videos.

Table 1: **Cross-view motion transformation.** Results for first-frame-to-video (f2v) and video-to-first-frame (v2f) mapping.

| Model | $\text{IoU}_{\text{f2v}} \uparrow$ | $\text{mAP}_{\text{f2v}} \uparrow$ | $\text{IoU}_{\text{v2f}} \uparrow$ | $\text{mAP}_{\text{v2f}} \uparrow$ |
|---|---|---|---|---|
| Interpolation | 0.67 | 0.72 | 0.67 | 0.72 |
| MegaSAM-based warp | 0.63 | 0.56 | 0.67 | 0.48 |
| DA3-based warp | 0.73 | 0.67 | 0.70 | 0.57 |
| TRACE (Ours, Stage I) | 0.80 | 0.91 | 0.77 | 0.85 |

**Evaluation.** We conduct two experiments using separate models for the two mapping directions: one predicts video-view boxes from first-frame boxes (f2v), and the other maps video-view boxes to the first frame (v2f). We evaluate both models against ground-truth boxes using IoU Rezatofighi et al. (2019) and mAP at an IoU threshold of 0.5.

**Baselines.** We adopt off-the-shelf 3D estimation models, MegaSAM Li et al. (2025b) and DA3 Lin et al. (2025), to estimate camera motion and per-frame depth from the input video, and use the estimated 3D information to perform depth-based warping, projecting the user's first-frame boxes

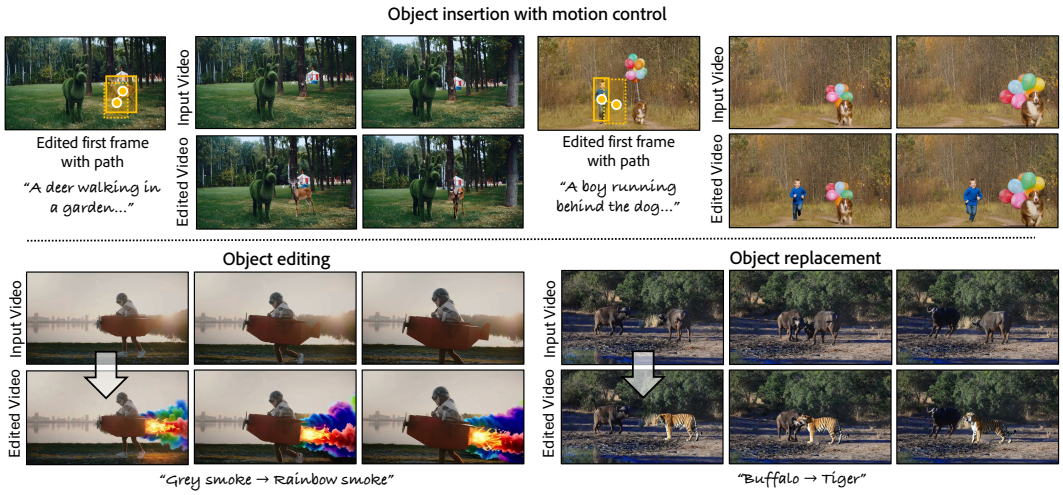



Figure 8: **Additional Application.** Applications for diverse video editing tasks. We leverage motion flexibility for various inference-time edits: (1) **Object insertion**: defining a first-frame layout and path, then using Qwen Wu et al. (2025) to inpaint the first frame and leveraging our model to propagate a new object; (2) **Object editing**: modifying first-frame attributes using Qwen and consistently propagating them along the trajectory; (3) **Object replacement**: substituting an object category in the initial frame using Qwen and re-synthesizing using the original object bounding boxes.

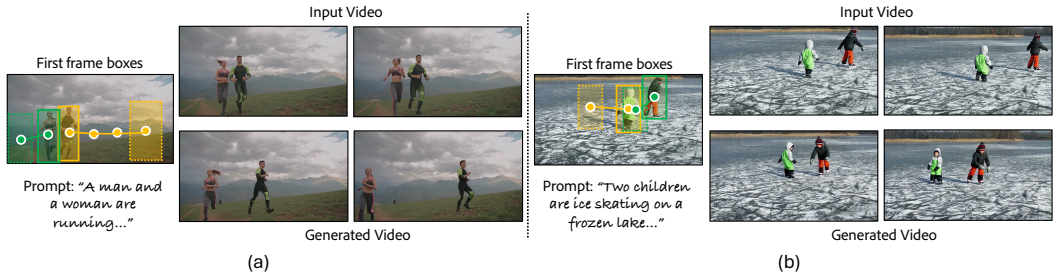



Figure 9: **Multiple object editing.** Even though our method is trained to edit one object, it enables simultaneous zero-shot motion editing for multiple objects by processing parallel trajectories while maintaining scene consistency.

onto subsequent frames. We also compare against the interpolation baseline that directly applies the original first-frame bounding box to all frames. (See the Appendix A.6 for details)

**Quantitative results.** As shown in Table 1, our method significantly outperforms both 2D interpolation and 3D-warping baselines (MegaSAM and DA3). Our model can implicitly learn camera motion from the input video and adapt to dynamic scene changes better than the baselines.

**Qualitative result.** Fig. 4 shows a visual example of box-transformation results from different methods. As shown in our qualitative analysis, simple Interpolation of bounding boxes fails to account for perspective shifts, resulting in an off-track trajectory that drifts to the right of the intended path. While 3D warping methods, with MegaSAM Li et al. (2025b) for depth and camera pose estimation, attempt to model the scene geometry, they remain highly susceptible to noisy depth and imprecise pose estimates. This leads to unstable, "jittery" boxes. In contrast, our proposed cross-view transformation generates a smooth path.

### 4.2 Video Re-Synthesis Module

**Training details.** For the Video Re-Synthesis Module, we fine-tune Wan 2.1 14B Wan et al. (2025) with LoRA Hu et al. (2022) on 81-frame clips at 24 fps and $480 \times 832$ resolution, conditioning on the text prompt, first frame, role-coded masked video, and separate object-synthesis and inpainting masks for 8K steps using AdamW with a learning rate of $1.2 \times 10^{-5}$, weight decay of $0.01$, and batch size 32 under a flow-matching $v$-pred objective Lipman et al. (2022).

**Evaluation.** We evaluate Stage II reconstruction on the DAVIS benchmark Perazzi et al. (2016),

Table 2: **Quantitative evaluation of Video Re-Synthesis Module.** We evaluate the video re-synthesis module on the DAVIS Perazzi et al. (2016) benchmark, comparing generated video quality, similarity to the input video, and box alignment against state-of-the-art baselines.

| Model | Type | Base Model | Similarity and Quality | | | Box Alignment |
|---|---|---|---|---|---|---|
| | | | PSNR ↑ | SSIM ↑ | LPIPS ↓ | Region Containment ↑ |
| MagicMotion Li et al. (2025a) | I2V | CogVideoX | 14.39 | 0.34 | 0.50 | 0.9824 |
| Wan-Mover Chu et al. (2025) | I2V | Wan2.1 (14B) | 13.07 | 0.29 | 0.52 | 0.9875 |
| Motion Canvas Xing et al. (2025) | I2V + track | Wan2.1 (14B) | 15.27 | 0.36 | 0.45 | 0.9893 |
| ReVideo Mou et al. (2024) | V2V | SVD | 13.22 | 0.23 | 0.51 | 0.9805 |
| HunyuanCustom Hu et al. (2025) | V2V + inpaint | Hunyuan | 21.50 | 0.67 | 0.35 | 0.9862 |
| VACE Jiang et al. (2025) | V2V + inpaint | Wan2.1 (14B) | 19.89 | 0.65 | 0.24 | 0.9874 |
| GenCompositor Yang et al. (2025) | V2V + inpaint | CogVideoX | 18.69 | 0.62 | 0.31 | 0.9801 |
| PISCO Gao et al. (2026) | V2V + inpaint | CogVideoX | 19.33 | 0.57 | 0.42 | 0.9823 |
| TRACE (Ours) | V2V | Wan2.1 (14B) | 20.48 | 0.71 | 0.19 | 0.9884 |

Table 3: **Quantitative evaluation of the Full Pipeline**. We evaluate the video quality, similarity to the input, and box alignment of our TRACE, and compare it with other baselines.

| Model | Type | Base Model | Similarity and Quality | | | Box Alignment |
|---|---|---|---|---|---|---|
| | | | FVD ↓ | FID ↓ | KID ↓ | Region Containment ↑ |
| MagicMotion Li et al. (2025a) | I2V | CogVideoX | 1492.56 | 46.91 | 0.0065 | 0.9903 |
| Wan-Move Chu et al. (2025) | I2V | Wan2.1 (14B) | 977.80 | 41.22 | 0.0029 | 0.9892 |
| MotionCanvas Xing et al. (2025) | I2V + track | Wan 2.1 (14B) | 957.10 | 39.79 | 0.0034 | 0.9890 |
| ReVideo Mou et al. (2024) | V2V | SVD | 1525.56 | 52.23 | 0.0072 | 0.9835 |
| HunyuanCustom Hu et al. (2025) | V2V + inpaint | Hunyuan | 780.13 | 41.81 | 0.0035 | 0.9822 |
| VACE Jiang et al. (2025) | V2V + inpaint | Wan 2.1(14B) | 916.37 | 60.86 | 0.0084 | 0.9900 |
| GenCompositor Yang et al. (2025) | V2V + inpaint | CogVideoX | 967.11 | 178.69 | 0.0916 | 0.9834 |
| PISCO Gao et al. (2026) | V2V + inpaint | CogVideoX | 772.09 | 67.97 | 0.0136 | 0.9834 |
| TRACE (Ours) | V2V | Wan2.1 (14B) | 614.94 | 34.13 | 0.0028 | 0.9906 |

using the same input-construction procedure as during training (Sec. 3.2). The guidance follows the object's original trajectory, so the input video provides a reconstruction reference. We evaluate changed trajectories in the complete-system experiment (Sec. 4.3). We measure reconstruction quality against the original video using SSIM Wang et al. (2004), PSNR, and LPIPS Zhang et al. (2018). To evaluate trajectory adherence, we use *Region Containment*, which measures the overlap between the generated-object mask and the box-guidance region, normalized by the area of the generated-object mask. This metric matches our goal of following the intended trajectory while preserving natural motion, rather than forcing the object to fit every guidance box exactly;(see the Appendix A.5 for details)

**Baselines.** We compare our method with two groups: (1) image-to-video (I2V) methods, including MagicMotion Li et al. (2025a), Wan-Move Chu et al. (2025), ReVideo Mou et al. (2024), and Motion Canvas Xing et al. (2025); and (2) state-of-the-art video-to-video (V2V) models: VACE Jiang et al. (2025) , HunyuanCustom Hu et al. (2025), and GenCompositor Yang et al. (2025).

**Quantitative results.** As shown in Table 2, TRACE achieves the best LPIPS and SSIM among V2V baselines and the second-best PSNR and Region Containment. I2V methods exhibit lower reconstruction similarity because they do not condition on the complete source video. Motion Canvas attains higher Region Containment by following box guidance more rigidly, whereas our smoothed-box training favors realistic object motion over exact box matching. Overall, TRACE provides a strong balance between source-video fidelity and trajectory adherence.

**Qualitative result**. Fig. 5 shows example video re-synthesis results from our experiment. The I2V baselines frequently hallucinate background content and fail to preserve the goat's appearance and color. HunyuanCustom introduces small hallucinated goats in the background, while VACE fails to maintain the brown goat's identity, generating a white goat and altering the background. GenCompositor produces comparatively lower video quality, which further degrades object identity. Our outputs remain highly faithful to the input video, correctly generating the brown goat while maintaining a consistent background.

### 4.3 COMPLETE SYSTEM EVALUATION

**Evaluation.** We evaluate the full framework (box transformation and video re-synthesis) on a manually curated test set of 70 videos. For each video, we annotate the target object location in the first

Table 4: **Ablation study of Cross-View Motion Transformation**. We demonstrate how different conditions and training objectives affect the model's performance.

| | Keep dynamic points | w/o points | w/o first frame | Concat boxes | Combine 2 tasks | Ours |
|---|---|---|---|---|---|---|
| IoU ↑ | 0.66 | 0.41 | 0.32 | 0.67 | 0.67 | 0.80 |
| mAP ↑ | 0.76 | 0.33 | 0.18 | 0.73 | 0.73 | 0.91 |

Table 5: **Ablation study of Video Re-synthesis Module**. We evaluate the impact of different training configurations on the model using our full integrated pipeline and custom dataset.

| Smooth Boxes | Data Aug | FVD ↓ | FID ↓ | KID ↓ | Region Containment ↑ |
|---|---|---|---|---|---|
| ✗ | ✓ | 785.00 | 39.14 | 0.0033 | 0.9856 |
| ✓ | ✗ | 643.62 | 37.91 | 0.0030 | 0.9878 |
| ✓ | ✓ | 614.94 | 34.13 | 0.0028 | 0.9906 |

and last frames on the first view to define a new motion path. These sparse annotations are interpolated into dense bounding boxes and used by Stage I to generate the full trajectory. The predicted trajectory is then used to re-synthesize the video. We compare against video re-synthesis baselines, providing them with a shared inpainted video via Omnimatte Lee et al. (2025) since they lack native object removal capabilities. We quantitatively assess video quality, control-following, and object identity preservation using FVD, FID, KID and Region Containment. We also conduct a **user study** on the same 70 videos to complement automatic metrics with human judgments of box alignment, video quality(motion quality, temporal coherence), object identity preservation, properties that are difficult to capture automatically. Participants compare our method with baselines along these dimensions.(see the Appendix A.8 for details)

**Quantitative result**. As shown in Table 3, our method achieves the best results across FVD, FID, KID, and Region Containment. Our user study (Fig. 10) confirms these practical advantages. Users consistently preferred our method over all baselines in terms of video quality, object identity preservation, and motion controllability.

**Qualitative result**. Fig. 6 compares our full pipeline against baselines. MagicMotion fails to preserve the camera path, replacing the child with the mother and altering the background. While MotionCanvas preserves camera motion, it exhibits background drift and inconsistent identities for both subjects. HunyuanCustom and VACE suffer from identity loss and hallucinations, while GenCompositor shows poor video quality and perspective mismatches. In contrast, our method maintains high identity fidelity and background consistency.

### 4.4 Ablation Study

We validate the design choices of the box transformation model in Table 4. First, we confirmed the importance of our conditional inputs: Removing key conditional inputs (either the point trajectory or the first-frame reference) causes a significant performance drop. Training with noisy box motion similarly degrades performance, underscoring the need for high-quality data. Second, we explored alternative task formulations ("inpainting-style" setup and "Combine 2 tasks") and found that both alternatives underperformed, confirming the superiority of our chosen architecture. For the Video Re-Synthesis model (Table 5), both box smoothing and data augmentation consistently improve performance. The full configuration achieves the lowest FVD, FID, and KID and the highest Region Containment, showing that these components jointly enhance generative quality and box alignment.

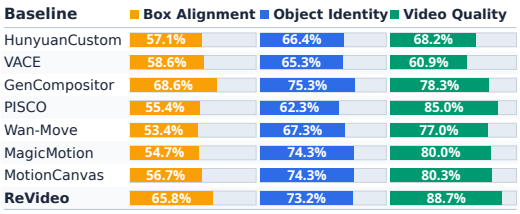



Figure 10: **User study.** Pairwise preference for TRACE over each baseline in box alignment, object identity, and video quality. Colored bars indicate the percentage of users who preferred TRACE over baseline methods.

## 5 Conclusion

We present TRACE, a novel framework for editing object motion in videos. We introduce a video-editing setting with first-frame object motion design, where users intuitively specify complex motion paths from a single anchor frame. To address the key challenge of cross-view spatial misalignment, our two-stage pipeline first employs a transformation module to predict a dense, frame-aligned bounding-box sequence from the user's first-frame path. A conditional re-synthesis model then uses this sequence to seamlessly "erase-and-resynthesize" the object, preserving the original video's content while making the selected object follow the user's new intended motion path. Experiments on real-world videos demonstrate high-quality, temporally consistent results, enabling new applications such as motion re-planning.

## REFERENCES


Jianhong Bai, Tianyu He, Yuchi Wang, Junliang Guo, Haoji Hu, Zuozhu Liu, and Jiang Bian. Uniedit: A unified tuning-free framework for video motion and appearance editing. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 10171–10180, 2025a.

Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, et al. Recammaster: Camera-controlled generative rendering from a single video. *arXiv preprint arXiv:2503.11647*, 2025b.

Yuxuan Bian, Zhaoyang Zhang, Xuan Ju, Mingdeng Cao, Liangbin Xie, Ying Shan, and Qiang Xu. Videopainter: Any-length video inpainting and editing with plug-and-play context control. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–12, 2025.

Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter1: Open diffusion models for high-quality video generation, 2023.

Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models, 2024.

Yuqing Chen, Junjie Wang, Lin Liu, Ruihang Chu, Xiaopeng Zhang, Qi Tian, and Yujiu Yang. O-disco-edit: Object distortion control for unified realistic video editing. *arXiv preprint arXiv:2509.01596*, 2025.

Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee. Tracking anything with decoupled video segmentation. In *ICCV*, 2023.

Ruihang Chu, Yefei He, Zhekai Chen, Shiwei Zhang, Xiaogang Xu, Bin Xia, Dingdong Wang, Hongwei Yi, Xihui Liu, Hengshuang Zhao, et al. Wan-move: Motion-controllable video generation via latent trajectory guidance. *arXiv preprint arXiv:2512.08765*, 2025.

Jiwoo Chung, Sangeek Hyun, and Jae-Pil Heo. Style injection in diffusion: A training-free approach for adapting large-scale diffusion models for style transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8795–8805, 2024.

Xiangbo Gao, Renjie Li, Xinghao Chen, Yuheng Wu, Suofei Feng, Qing Yin, and Zhengzhong Tu. Pisco: Precise video instance insertion with sparse control. *arXiv preprint arXiv:2602.08277*, 2026.

Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve your own correlation: A noise prior for video diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22930–22941, 2023.

Jing Gu, Yuwei Fang, Ivan Skorokhodov, Peter Wonka, Xinya Du, Sergey Tulyakov, and Xin Eric Wang. Via: Unified spatiotemporal video adaptation framework for global and local video editing. *arXiv preprint arXiv:2406.12831*, 2024.

Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation. 2022.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

Teng Hu, Zhentao Yu, Zhengguang Zhou, Sen Liang, Yuan Zhou, Qin Lin, and Qinglin Lu. Hunyuancustom: A multimodal-driven architecture for customized video generation, 2025. URL https://arxiv.org/abs/2505.04512.

Hsin-Ping Huang, Yu-Chuan Su, Deqing Sun, Lu Jiang, Xuhui Jia, Yukun Zhu, and Ming-Hsuan Yang. Fine-grained controllable video generation via object appearance and context. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3698–3708. IEEE, 2025.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17191–17202, 2025.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *Proc. ECCV*, 2024.

Yao-Chih Lee, Erika Lu, Sarah Rumbley, Michal Geyer, Jia-Bin Huang, Tali Dekel, and Forrester Cole. Generative omnimatte: Learning to decompose video into layers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12522–12532, 2025.

Pengxiang Li, Kai Chen, Zhili Liu, Ruiyuan Gao, Lanqing Hong, Guo Zhou, Hua Yao, Dit-Yan Yeung, Huchuan Lu, and Xu Jia. Trackdiffusion: Tracklet-conditioned video generation via diffusion models. *arXiv preprint arXiv:2312.00651*, 2023.

Quanhao Li, Zhen Xing, Rui Wang, Hui Zhang, Qi Dai, and Zuxuan Wu. Magicmotion: Controllable video generation with dense-to-sparse trajectory guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12112–12123, 2025a.

Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. MegaSaM: Accurate, fast and robust structure and motion from casual dynamic videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025b.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

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

Shaoteng Liu, Tianyu Wang, Jui-Hsien Wang, Qing Liu, Zhifei Zhang, Joon-Young Lee, Yijun Li, Bei Yu, Zhe Lin, Soo Ye Kim, and Jiaya Jia. Generative video propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17712–17722, June 2025a.

Yuhao Liu, Tengfei Wang, Fang Liu, Zhenwei Wang, and Rynson WH Lau. Shape-for-motion: Precise and consistent video editing with 3d proxy. *arXiv preprint arXiv:2506.22432*, 2025b.

Chong Mou, Mingdeng Cao, Xintao Wang, Zhaoyang Zhang, Ying Shan, and Jian Zhang. Revideo: Remake a video with motion and content control. *Advances in Neural Information Processing Systems*, 37:18481–18505, 2024.

Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 724–732, 2016.

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

Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. Fatezero: Fusing attentions for zero-shot text-based video editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15932–15942, 2023.

Haonan Qiu, Zhaoxi Chen, Zhouxia Wang, Yingqing He, Menghan Xia, and Ziwei Liu. Freetraj: Tuning-free trajectory control in video diffusion models. *arXiv preprint arXiv:2406.16863*, 2024.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 658–666, 2019.

Jan Philipp Schneider, Pratik Singh Bisht, Ilya Chugunov, Andreas Kolb, Michael Moeller, and Felix Heide. Neural atlas graphs for dynamic scene decomposition and editing. *arXiv preprint arXiv:2509.16336*, 2025.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

Quoc-Truong Truong, Vinh-Tiep Nguyen, Lan-Phuong Nguyen, Hung-Phu Cao, and Duc-Tuan Luu. Local part attention for image stylization with text prompt. *Neural Computing and Applications*, 36(34):21859–21871, 2024.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12490–12500, 2025.

Jiawei Wang, Yuchen Zhang, Jiaxin Zou, Yan Zeng, Guoqiang Wei, Liping Yuan, and Hang Li. Boximator: Generating rich and controllable motions for video synthesis. *arXiv preprint arXiv:2402.01566*, 2024a.

Wen Wang, Yan Jiang, Kangyang Xie, Zide Liu, Hao Chen, Yue Cao, Xinlong Wang, and Chunhua Shen. Zero-shot video editing using off-the-shelf image diffusion models. *arXiv preprint arXiv:2303.17599*, 2023.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

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 *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024b.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025. URL https://arxiv.org/abs/2508.02324.

Zeqi Xiao, Wenqi Ouyang, Yifan Zhou, Shuai Yang, Lei Yang, Jianlou Si, and Xingang Pan. Trajectory attention for fine-grained video motion control. *arXiv preprint arXiv:2411.19324*, 2024.

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 *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–11, 2025.

Jiacong Xu, Yiqun Mei, Ke Zhang, and Vishal M Patel. Freevis: Training-free video stylization with inconsistent references. *arXiv preprint arXiv:2510.01686*, 2025.

Shuai Yang, Yifan Zhou, Ziwei Liu, and Chen Change Loy. Rerender a video: Zero-shot text-guided video-to-video translation. In *SIGGRAPH Asia 2023 Conference Papers*, pp. 1–11, 2023.

Shuzhou Yang, Xiaoyu Li, Xiaodong Cun, Guangzhi Wang, Lingen Li, Ying Shan, and Jian Zhang. Gencompositor: generative video compositing with diffusion transformer. *arXiv preprint arXiv:2509.02460*, 2025.

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

Guofeng Zhang, Angtian Wang, Jacob Zhiyuan Fang, Liming Jiang, Haotian Yang, Alan Yuille, and Chongyang Ma. Hector: Hybrid editable compositional object references for video generation. *arXiv preprint arXiv:2603.08850*, 2026.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

# A Appendix

We provide several visual comparisons and application results in ***index.html***. Please open this webpage to view all videos.

## A.1 Additional applications

Beyond the main application of altering object motion, we also extend our framework to support several additional tasks.

**First-frame object editing and insertion.** In addition to changing an object's motion, users can modify its appearance or insert a new object by editing the first frame, and then specify the desired motion path using the same interface described in the main paper. The edited first frame provides the object's appearance and identity reference, while the transformed trajectory supplies per-frame motion guidance. Stage II propagates the edited or inserted object along this path and re-synthesizes it consistently throughout the video, enabling joint control of object appearance and motion.

**Multiple-object editing.** Although Stage II is trained with only one object per video, it generalizes at inference to simultaneous control of multiple objects without retraining. For two objects, we construct one blue–gray mask pair per object and combine both pairs into the role-coded conditioning masks: the blue regions mark the source locations to remove and inpaint, while the gray regions specify the target trajectories for object synthesis. Stage 2 then re-synthesizes both objects along their respective guidance paths. This zero-shot capability demonstrates the robustness and generalization of the model beyond its single-object training setting; examples are provided in the main paper.

## A.2 Limitations and Future Work

**Stage I failure.** i) Stage I relies on point tracks to estimate camera motion, so its predictions depend on the quality of the off-the-shelf tracker. When the input video contains little texture, the tracks can be noisy, causing the predicted boxes to deviate slightly from the user-defined trajectory and the resulting motion to differ from the user's intent. For example, in Fig. 11, the user intends the car to move in front of the tree, but the predicted boxes overshoot the desired trajectory and misalign the car with respect to the tree. To correct such deviations, the user can provide an ending box in the last frame to explicitly specify the desired final location. More generally, Stage I supports flexible box guidance, including boxes specified in the first frame or only in sparse frames.

**Stage II failure.**: i) In some cases, Stage II fails to remove the original object from its source region (the blue-mask region) before synthesizing it at the target location, resulting in two copies of the object. This is an inpainting failure: when the object is visually similar to the background, the model may interpret it as background and leave it unchanged instead of reconstructing the background. For

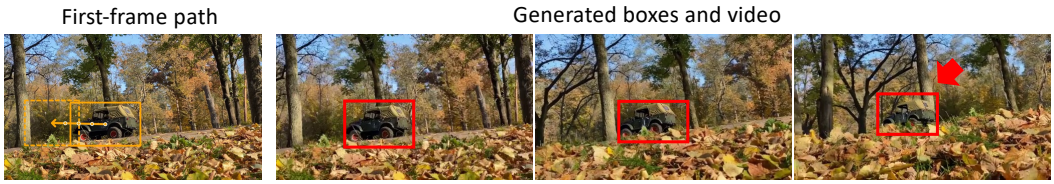



Figure 11: **Limitation of Stage I.** Noisy point tracks can cause the predicted bounding boxes to deviate from the user-defined trajectory, resulting in motion that does not fully match the user's intent.

example, in the first case in Fig. 12, the green fake deer blends into the grass, so Stage 2 retains the deer in its original location while also generating it along the target trajectory. A similar failure can occur when objects in the input video are difficult to distinguish from their background.

ii) When the user-provided inpainting region (the blue-mask region) does not cover effects associated with the object, such as shadows, reflections, or water ripples, Stage 2 edits the object's trajectory but does not move or regenerate those effects. Consequently, the object may be synthesized at its new location while its shadow or other effect remains at the original location, producing an implausible, disconnected result, as shown in the second example in Fig. 12. This limitation is partly caused by the current input design: the user explicitly specifies the object's trajectory, whereas the desired motion of its associated effects is only implicit. Similar issues are common in inpainting and object-removal methods when such effects are not included in the editing region. In future work, inspired by prior work on effect-aware inpainting, we plan to extend Stage 2 to jointly model and propagate an object's associated effects along its new trajectory.

iii) Stage II uses only the first frame as an appearance reference for the object. If the object is small, blurry, or occluded in that frame, important identity cues may be unavailable, forcing the model to hallucinate the missing appearance in later frames. As shown in Fig. 12(b), the monkey's face is not visible in the first frame; Stage II therefore synthesizes its facial appearance, and the generated monkey's face in subsequent frames may not exactly match the monkey in the input video. Similar identity drift can occur whenever the target object's appearance is occluded or otherwise ambiguous in the first frame. This limitation stems from the model design, which assumes that the first frame provides a clear and unambiguous representation of the object's appearance and identity. To better preserve object identity when the first-frame observation is ambiguous, future work could condition the model on multiple cropped object views sampled from different frames or on a segmented object video. We leave these extensions for future exploration.

**Future Work.** We plan to address each limitation with a targeted extension. For Stage 2 failure (i), we will train with paired object-removal and inpainting data that emphasize low-contrast objects and explicitly supervise removal of the source object and reconstruction of the revealed background. For Stage 2 failure (ii), we will introduce effect-aware training data and automatically expand or predict the editing region to cover object-associated shadows, reflections, and water effects, allowing them to be regenerated and propagated jointly with the object along its target trajectory. For Stage 2 failure (iii), we will augment first-frame conditioning with appearance cues aggregated from multiple input frames and apply stronger identity-consistency supervision to reduce hallucinated details when the object is unclear in the first frame. We currently fine-tune Wan 2.1-14B using LoRA; scaling the training data and updating more of the backbone may further improve these Stage 2 capabilities. Finally, to address the Stage 1 failure under extreme camera motion, we will augment the current ReCamMaster training data with synthetic videos rendered in Unreal Engine using larger camera movements and viewpoint changes.

### A.3 Training Details

**Cross-View Motion Transformation:** We train the model from scratch using 1 NVIDIA H100 node. The training resolution is set to $488 \times 832$ with 73 frames per clip, and we use 28 inference steps during evaluation. To match the 81-frame clips used in Stage 2, we insert eight linearly interpolated frames between consecutive frames in the final portion of each 73-frame clip, increasing its length to 81 frames.

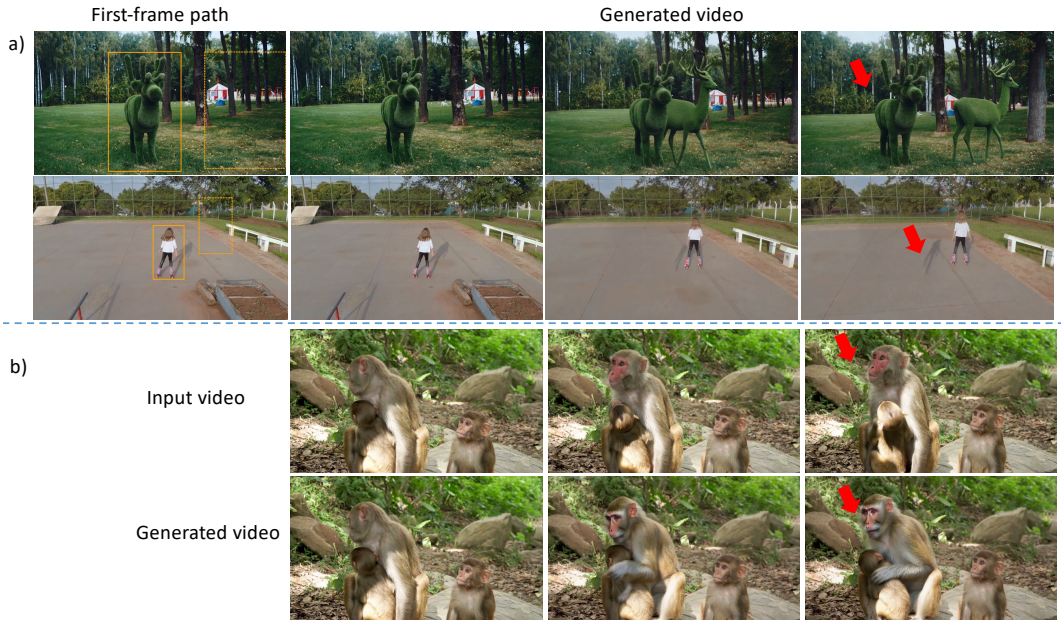



Figure 12: **Limitation stage II.** The figure illustrates three representative failure cases: (1) the original object remains in its source region while another instance is generated along the user-specified trajectory; (2) object-associated effects, such as shadows or reflections, do not follow the edited object; and (3) the object is partially occluded in the first frame, causing identity ambiguity in subsequent frames.

**Video Resynthesis:** We fine-tune the Wan 2.1 14B base model using LoRA on 4 NVIDIA H100 nodes. Training is conducted at a resolution of $488 \times 832$ with 81-frame video clips. During fine-tuning, LoRA fine-tuning are applied only to the DiT blocks, while the rest of the base model remains frozen. The model uses three conditioning inputs: the role-coded masked video (containing blue and gray regions), a binary object-synthesis mask, and a binary inpainting mask. Their embedding networks are initialized from scratch and therefore trained with full-parameter updates. In contrast, the pretrained Wan weights remain frozen, and only LoRA adapters in the DiT blocks are optimized; this reduces training cost while preserving the pretrained video prior.

### A.4 Stage II Data curation details

**Data augmentation for Stage II training.** At inference, Stage II receives box trajectories predicted by Stage 1, which may contain small localization errors or missing boxes. To make the full pipeline robust to these prediction errors, we add small random perturbations to the training box guidance and independently drop the guidance in each frame with probability 0.03. These augmentations simulate imperfect Stage I outputs and prevent Stage II from overfitting to clean ground-truth trajectories. As shown in Table 5, data augmentation improves FVD, FID, KID, and Region Containment, demonstrating better video quality and trajectory alignment for the complete two-stage pipeline.

**Augmented mask creation.** As illustrated in Fig. 7, we construct the self-supervised training inputs from each video and its automatically extracted object masks. To synthesize the blue background mask, we sample a random motion direction and speed, select a perturbed terminal bounding box, and linearly interpolate the boxes across frames to form an augmented trajectory. We initialize each box size from the corresponding object box and add a small random perturbation. Rasterizing this trajectory yields the blue background mask, while the original object mask is retained as the gray mask. The masked video is used as input, and the original unmasked video serves as the reconstruction target.

**Smoothing the object boxes.** Stage I produces a temporally smooth box sequence by interpolating sparse user inputs, but real objects can undergo non-rigid motion; for example, a flying chicken's wings may move even when its bounding-box trajectory is smooth. If the model treats each box as




Figure 13: **Stage I data curation.** We extract the reference frame and object boxes from a static-camera video, then use ReCamMaster to render it along a sampled camera path. The rendered video provides the corresponding video-view boxes and CoTracker point tracks.

an exact frame-level constraint, it may overfit to the guidance and generate rigid or unnatural motion. We therefore augment training by smoothing the box sizes over randomly sized consecutive chunks of 8–12 frames: within each chunk, all boxes share the maximum size while their original centers are preserved. These looser, temporally coherent masks teach the model to treat box guidance as an approximate region, allowing natural object motion while remaining controllable. The ablation in Table 5 shows that box smoothing improves FVD, FID, KID, and Region Containment; together with data augmentation, it yields the best result on every reported metric. This confirms that smoothing improves both perceptual quality and trajectory alignment.

**Mask overlap and visibility.** Let $\mathrm{M}_{\mathrm{obj}}$ and $\mathrm{M}_{\mathrm{inp}}$ denote the binary object and inpainting masks, respectively. Pixels marked by the object mask are treated as visible object guidance, whereas pixels marked only by the inpainting mask are regenerated. To avoid conflicting conditions, the object mask has priority wherever the masks overlap. We therefore use the effective inpainting mask $\widetilde{\mathrm{M}}_{\mathrm{inp}} = \mathrm{M}_{\mathrm{inp}} \odot (1 - \mathrm{M}_{\mathrm{obj}})$. Thus, a pixel is assigned to the object condition if $\mathrm{M}_{\mathrm{obj}} = 1$, to inpainting if $\mathrm{M}_{\mathrm{obj}} = 0$ and $\mathrm{M}_{\mathrm{inp}} = 1$, and to unchanged context otherwise. This rule preserves object identity and boundaries while preventing the same pixel from receiving contradictory object and inpainting conditions.

### A.5 Metric explanation

**Region Containment .**We use Region Containment rather than IoU or mAP because the smoothed boxes described above are intentionally approximate guidance regions: they define the desired trajectory, but they should not force the generated object to match the box tightly at every frame. IoU and mAP reward high overlap between two boxes and can therefore penalize natural motion inside a larger guidance region. Region Containment instead measures whether the generated object remains inside the guidance region. Formally, let $\mathrm{M}_{\mathrm{obj}}$ denote the region obtained by rasterizing the Stage 1 predicted trajectory boxes, and let $\mathrm{M}_{\mathrm{generated}}$ denote the generated-object mask. We compute $\mathrm{RC} = |\mathrm{M}_{\mathrm{generated}} \cap \mathrm{M}_{\mathrm{obj}}|/|\mathrm{M}_{\mathrm{generated}}|$, the fraction of generated-object pixels contained within the trajectory boxes. This metric evaluates trajectory adherence while allowing natural non-rigid motion.

### A.6 Additional data curation and baselines of stage I

**Data curation process.** For each Stage I training sample, we select an object in a static-camera video and detect it throughout the clip to obtain its box sequence in the first-frame view. We then re-render the same video with ReCamMaster Bai et al. (2025b), using a camera path sampled at random from its dataset. We detect the corresponding object in the rendered video to obtain its video-view box sequence and use CoTracker to extract point tracks. Each sample contains the reference first frame, the object's first-frame-view box sequence, its corresponding video-view box sequence, and the point tracks from the rendered video (Fig. 13).

**Baselines**: We compare our, TRACE Stage I with MegaSAM Li et al. (2025b) and DA3 (DepthAnything-v3 Lin et al. (2025)) baselines for cross-view motion transformation. These methods are used to estimate the camera parameters and per-frame depth. These are then used to warp

Table 6: **Cross-view motion transformation model compared with baselines.** Comparison of baseline methods from first-frame view to video view and vice versa. We use MegaSAM Li et al. (2025b) and DepthAnything-V3 Lin et al. (2025) to estimate camera information and depth when warping the center path or the four box corners. Higher (↑) is better for all metrics.

| Model | $\mathrm{IoU}_{\mathrm{f2v}}\uparrow$ | $\mathrm{mAP}_{\mathrm{f2v}}\uparrow$ | $\mathrm{IoU}_{\mathrm{v2f}}\uparrow$ | $\mathrm{mAP}_{\mathrm{v2f}}\uparrow$ |
|---|---|---|---|---|
| Interpolation | 0.67 | 0.72 | 0.67 | 0.72 |
| MegaSAM-based warping (center box) | 0.69 | 0.64 | 0.71 | 0.53 |
| MegaSAM-based warping (4 corners) | 0.63 | 0.56 | 0.67 | 0.48 |
| DepthAnything-V3-based warping (center box) | 0.79 | 0.73 | 0.83 | 0.65 |
| DepthAnything-V3-based warping (4 corners) | 0.73 | 0.67 | 0.77 | 0.57 |
| TRACE (Ours, Stage I) | 0.80 | 0.91 | 0.77 | 0.85 |

the boxes from the first-frame view to each subsequent frame, for first-frame path to video-view transformation. Our method still outperforms these warping-based approaches, even when using different camera-calibration estimators. The results can be seen in Table 6. We use 2 different depths for warping the bounding boxes, i) *center box:* the depth taken from the center of the box, ii)*4 corners:* the depth of each of the coordinates of the bounding box.

### A.7 Additional baselines for Stage II video resynthesis

In addition to the baselines evaluated in the main paper, we provide qualitative comparisons with ReVideo Mou et al. (2024) and Shape-for-Motion Liu et al. (2025b) to examine whether existing video-editing methods can support flexible user-specified box guidance. These methods are not designed to follow arbitrary, sparse box trajectories and therefore do not reliably realize the user's intended motion. As shown in Fig. 14, ReVideo struggles when the target object occupies a large image region, resulting in degraded object appearance and overall video quality. Shape-for-Motion relies on a reconstructed 3D mesh and attempts to preserve the object's original geometry, depth, and motion. Consequently, its output often resembles a rigid copy of the original object translated along the guidance path, producing motion that is neither natural nor sufficiently responsive to flexible box guidance (Fig. 15). In contrast, our Stage II treats the boxes as flexible spatial guidance, synthesizing the object along the user-defined trajectory while allowing natural, non-rigid motion and better integration with the surrounding video.

### A.8 User Study

The user study design is illustrated in Fig. 16. We conducted a comprehensive pairwise comparison study to evaluate our method against baseline approaches.

**Evaluation Criteria.** To comprehensively evaluate object-centric video generation while minimizing ambiguity and cognitive bias, we decomposed the assessment into three independent criteria. Each criterion was evaluated through a separate question to ensure focused and reliable judgments.

***Box Alignment.*** Evaluates whether the generated object follows the user-specified trajectory over time. Participants were asked: "Which video more closely follows the displayed trajectory guidance throughout the sequence?" They were instructed to judge the object's motion path rather than exact frame-by-frame box coverage.

***Object Identity.*** Evaluates whether the generated object preserves the identity and appearance of the original object throughout the sequence. Participants were asked: "Compared with the original object in the input video, which generated video better preserves its category, shape, color, texture, and appearance consistency over time?"

***Video Quality.*** Evaluates overall perceptual quality, including visual realism, temporal coherence, visible artifacts, and especially the naturalness of the generated object's motion. Participants were asked: "Which generated video looks more realistic and temporally coherent, contains fewer visible

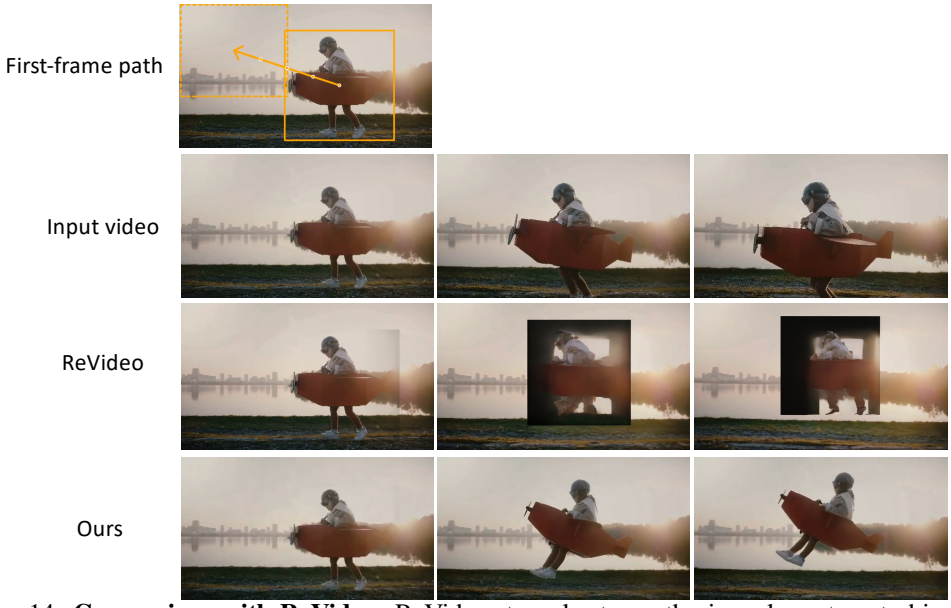



Figure 14: **Comparison with ReVideo.** ReVideo struggles to synthesize a large target object and produces degraded visual quality, whereas our Stage II better preserves object appearance and video quality while following the user-specified box trajectory.

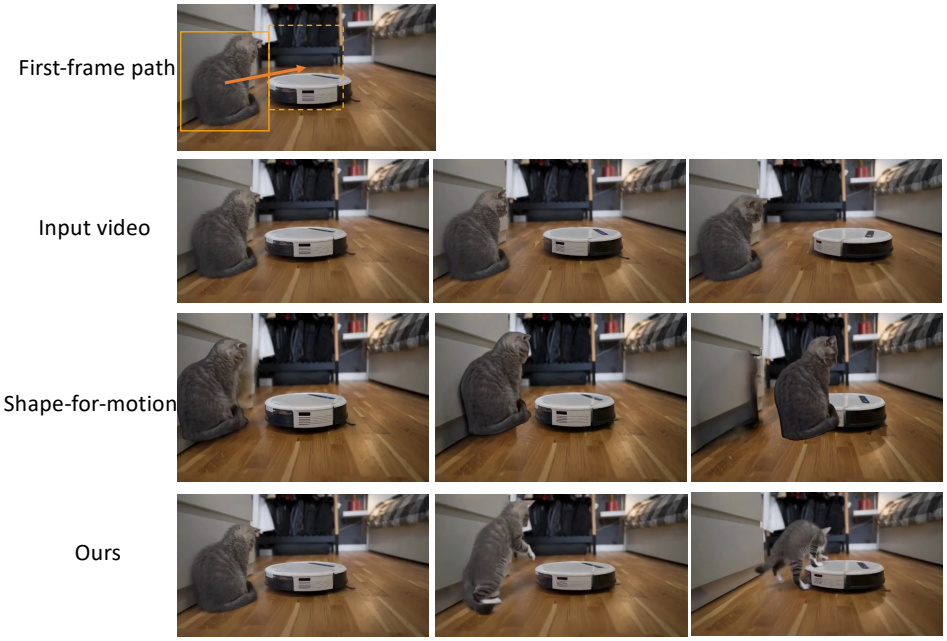



Figure 15: **Comparison with Shape-for-Motion.** Shape-for-Motion preserves the object's original 3D geometry and motion, causing the edited object to resemble a rigid copy translated along the guidance trajectory. Our Stage II follows the user-specified boxes while producing more natural and flexible object motion.

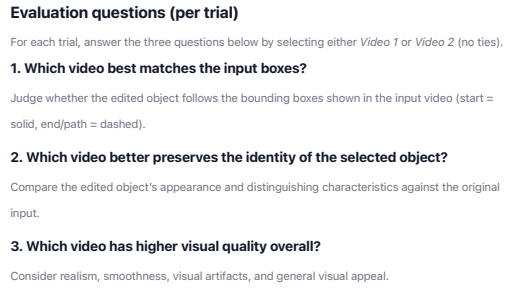
artifacts, and shows smoother, more natural object motion?"

**Setup.**

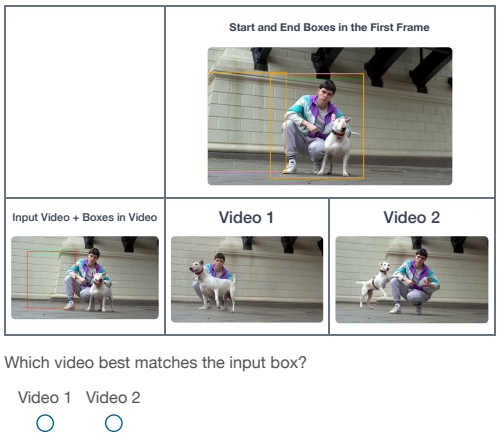
***Inputs:*** The evaluation was conducted on a full set of 70 generated videos for each methos.

***Comparison Protocol:*** We adopted a pairwise comparison paradigm. In each trial, participants were presented with two videos: one generated by our method and one generated by a single baseline model.

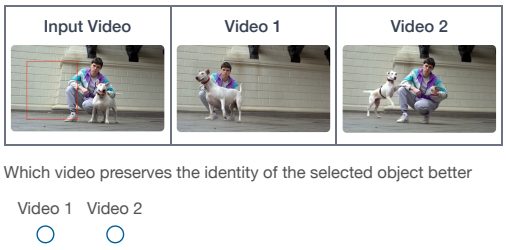
***Interface Design:*** The interface displayed the two videos side-by-side to facilitate direct comparison




(a) Explanation for questions




(b) Question and layout for Controllability




(c) Question and layout for Object Identity




(d) Question and layout for Video quality

Figure 16: **User study design.** Figure shows questions and comparison page shown to annotators to comapre the quality of our results with baselines.

while avoiding visual redundancy or distracting elements. No additional cues were provided that could bias participants.

***Participants and Trials:*** We recruited 52 participants. Each evaluated 40 video pairs comparing our method with one of eight baselines, yielding 2,080 pairwise trials in total.

**Results Reporting.** The results of the pairwise comparisons are summarized in the main paper, Fig. 10 . We report performance using the preference percentage, defined as the proportion of responses in which participants selected our method over the baseline for a given criterion.

The preference ratio ($\mathcal{P}$) is computed as:

$$\mathcal{P} = \frac{\#\text{Responses Preferring Our Method}}{\text{Total \#Responses}} \times 100\%.$$

**Overall Findings.** Across all three evaluation criteria—**Box Alignment**, **Object Identity**, and **Video Quality**—participants demonstrate a clear and consistent preference for our method over the corresponding baselines. These results validate the effectiveness of our approach in improving motion controllability, identity preservation, and overall perceptual quality in object-centric video generation.

### A.9 External point tracker

To evaluate the sensitivity of Stage 1 to point-tracking errors, we perturb each tracked point with zero-mean Gaussian noise whose standard deviation is $\sigma \in \{0, 4, 10\}$ pixels. Table 7 reports the resulting cross-view box-transformation accuracy. Under moderate noise ($\sigma = 4$ px), the mean of the four metrics decreases from $0.83$ to $0.80$, an absolute drop of $0.03$ (approximately $3.6\%$), indicating that Stage 1 remains relatively robust to small tracking errors. With larger noise ($\sigma = 10$ px),

| Model | $\text{IoU}_{\text{f2v}}\uparrow$ | $\text{mAP}_{\text{f2v}}\uparrow$ | $\text{IoU}_{\text{v2f}}\uparrow$ | $\text{mAP}_{\text{v2f}}\uparrow$ |
|---|---|---|---|---|
| STD=0 px | 0.80 | 0.91 | 0.77 | 0.85 |
| STD=4 px | 0.77 | 0.88 | 0.75 | 0.81 |
| STD=10 px | 0.70 | 0.81 | 0.68 | 0.72 |

Table 7: Stage 1 cross-view box-transformation accuracy under different levels of Gaussian noise added to the point tracks.

| Method | Time (Stage 1) | Time (Stage 2) | GPU Memory (Stage 1) | GPU Memory (Stage 2) |
|---|---|---|---|---|
| Ours | 4.15 s | 1525.7 s | 17.3 GB | 65.5 GB |
| Wan2.1-14B | – | 1371.7 s | – | 58.5 GB |
| VACE (Wan2.1-14B) | – | 2113.1 s | – | 79.2 GB |

Table 8: **Inference efficiency.** Runtime and peak GPU memory for our two-stage model compared with Wan2.1-14B and VACE.

the mean decreases to $0.73$, corresponding to an absolute drop of $0.11$ (approximately $12.6\%$). The degradation is consistent in both directions—from the first-frame view to the video view (f2v) and vice versa (v2f)—with the largest drop occurring in v2f mAP, which decreases from $0.85$ to $0.72$. These results show that Stage 1 tolerates moderate point-track noise but becomes increasingly sensitive to larger perturbations, supporting the limitation discussed above: inaccurate tracks can lead to misaligned predicted box trajectories.

### A.10 Inference efficiency

We benchmark inference on a single NVIDIA A100 GPU using 81-frame videos at a resolution of $480 \times 720$. As shown in the table Table 8, Stage I introduces little computational overhead, requiring 4.15 s and 17.3 GB of peak GPU memory per video. Stage II accounts for most of the runtime, requiring 1525.7 s and 65.5 GB. Its computational cost is comparable to the Wan2.1-14B backbone (1371.7 s and 58.5 GB) and lower than VACE (2113.1 s and 79.2 GB). These results show that the cross-view motion transformation in Stage I is computationally lightweight, while the overall inference cost is dominated by the 14B video diffusion backbone.

### A.11 Qualitative inpainting analysis

To analyze local inpainting quality, Figure 17 presents high-resolution crops of regions exposed after object motion. We examine three aspects: completion of the occluded background, continuity of fine textures and lighting, and seamless blending at the inpainting boundary. In these successful examples, the model reconstructs detailed scene content and preserves complex appearance cues, including refracted lighting, without obvious seams. This crop-level analysis complements full-frame FID, FVD, and KID, since local inpainting artifacts may occupy only a small portion of a frame and can be diluted by global metrics.

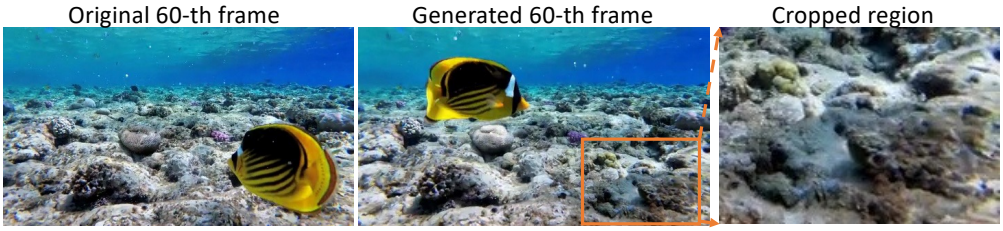



Figure 17: **Qualitative analysis of inpainting quality.** High-resolution crops compare the original and generated regions, highlighting background completion, texture continuity, and boundary blending after object motion.