MotionV2V: Editing Motion in a Video

Ryan Burgert[1,2] Charles Herrmann[1] Forrester Cole[1] Michael S Ryoo[2]
Neal Wadhwa[1] Andrey Voynov[1] Nataniel Ruiz[1]

[1]Google [2]Stony Brook University

Abstract

While generative video models have achieved remarkable fidelity and consistency, applying these capabilities to video editing remains a complex challenge. Recent research has extensively explored motion controllability as a means to enhance text-to-video generation or image animation; however, we identify precise motion control as a promising, yet under-explored, paradigm for editing ***existing videos****. In this work, we propose modifying video motion by directly editing sparse trajectories extracted from the input. We term the deviation between input and output trajectories a 'motion edit' and demonstrate that this representation, when coupled with a generative backbone, enables many powerful video editing capabilities. To achieve this, we introduce a novel pipeline for generating 'motion counterfactuals' — video pairs that share identical content but distinct motion — and fine-tune a motion-conditioned video diffusion architecture on this dataset. Our approach allows for edits that start at any timestamp and propagate naturally. In a 4-way head-to-head user study, our model achieves over 65% preference against prior work. Please see our project page: ryanndagreat.github.io/MotionV2V*

1. Introduction

Consider filming a climactic race between two dogs: your Corgi and a friend's Bichon. The original video sees the Bichon take the win. After countless recent advances in generative models, does the technology exist to modify this video such that your Corgi is victorious? We propose a method for generalized motion editing in existing user-provided videos that successfully tackles this unsolved problem.

Historically, tackling this problem in the VFX industry has been notably hard. A reshoot for scenes that need substantial changes is usually the necessary option. VFX pipelines can use tricks like retiming and plate stitching, isolated retimes with rotoscoping, or even full-dog CGI replacements. These typically require a high level of skill and large amounts of human hours.

Modern generative models, with their impressive priors, show promise in tackling traditional VFX tasks. In this subfield, current methods for motion editing fall into different categories, each exhibiting significant constraints. Image-to-video (I2V) based approaches like Re-Video [24] and Go-with-the-Flow [3] can only generate new video with specified motion conditioned on a single image. Using these on the first frame of a video can give the illusion of video motion control, but have significant drawbacks. For example, content generated in regions that do not appear in that initial frame will be entirely hallucinated, whereas for true video motion editing these regions are known and should remain identical. Re-Video attempts to address this problem by inpainting information from the original video into the edited video, a technique which fundamentally breaks down when the video includes camera movement. Human-specific methods like MotionFollower and MotionEditor can edit motion but are limited to full-body human movements and cannot handle general objects or scenes. Likewise there are also works that allow editing camera trajectories in videos such as ReCapture [44] and ReCamMaster [1]. These are not able to edit subject motion.

In this work we introduce motion edits, a new approach for editing videos by controlling the change in motion from the original to the edited video using video diffusion models. While there has been a large amount of successful recent work on appearance-based video editing (i.e. transforming visual style while preserving motion structure), motion editing presents a fundamentally different challenge. When editing how objects move within a scene (e.g. making a person walk in a different direction), the structural correspondence between input and output videos is broken. This makes the problem harder than appearance-based video editing and renders standard video editing techniques like DDIM inversion ineffective.

Our method addresses this problem, and the limitations of prior work, by acting on the complete video and its motion representation. Users provide an input video along with some sparse tracking points on objects they wish to control; these objects are then automatically tracked throughout the

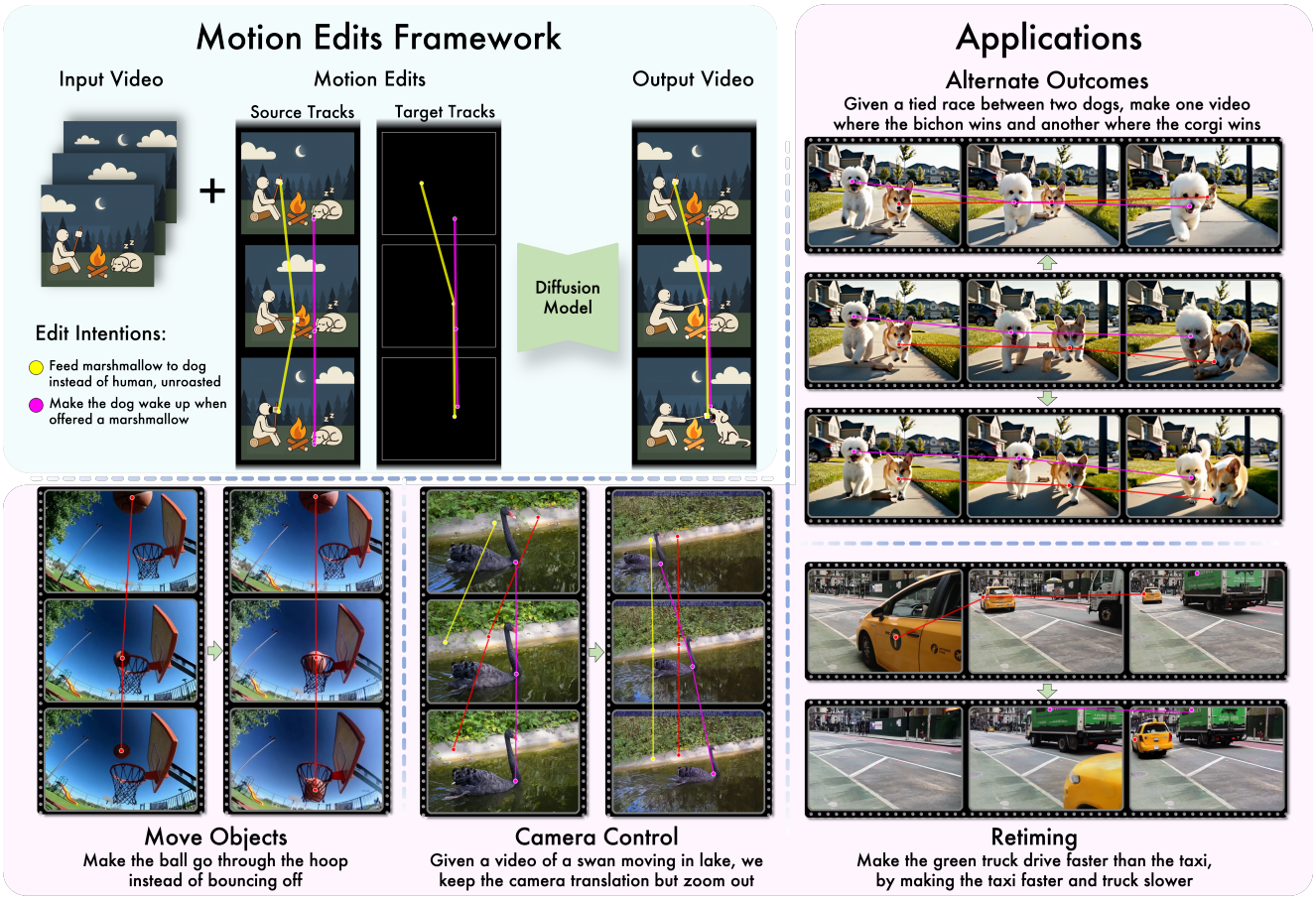

Figure 1. **Motion Edits Framework:** Users provide an input video along with source motion tracks (colored dots connected by lines, extracted from the input) and target motion tracks (user-specified desired motion). Lines indicate point trajectories while dot presence/absence indicates visibility. Our diffusion model generates an output video matching the target motion. **Applications:** Our method can edit videos in a true sense, where content is preserved but motion is changed.

video. Users can then choose to either anchor these points (to preserve original motion) or modify them (to edit the trajectory). For example, in a video of a person walking into a crowd, the system tracks both entities; the user can specify a new direction for the person by altering their tracks while strictly preserving the crowd's original motion. In a more complex edit, the user can change the camera by editing all the points.

Our approach enables diverse video editing capabilities (Fig. 1): object motion editing, camera motion editing, control over the timing of content, and successive edits. The motion edits are naturally introduced into the output video and the video model handles plausibility correctly - e.g. when dragging a person's tracking point through an image the model will make the person traverse the image by walking. Our approach requires no manual masking and can handle any type of object, while also maintaining scene consistency even when both object motion and camera movement are edited simultaneously. And in contrast to prior methods, our approach also allows the ability to change when an object appears in the frame. Finally, our model generalizes to vastly different scenes and objects and achieves state-of-the-art performance in quantitative comparisons and human evaluations.

In summary, we propose the following contributions:

- We identify motion as a powerful control signal for video editing and propose directly editing sparse trajectories extracted from the input video to change the motion of the output video. We define the change between the input and target trajectories as a "motion edit" and show that motion edits, coupled with a powerful generative video model, can address several challenging video editing tasks.
- We present a methodology to train a video diffusion model to generate high quality "motion counterfactual" video pairs which have the scene appearance but different motion. As part of this, we also identify sources of data that work well in this training.
- We propose a new model architecture with careful conditioning on both user-specified video and motion trajectories that generates a motion-edited output.

2. Related Works

Diffusion models have fundamentally reshaped media generation, evolving from foundational image synthesis frameworks [13, 31] to complex video dynamics [2, 14, 15, 33]. Recent text-conditioned video models [36, 42, 49] have further advanced the field by adopting transformer-based architectures [27] for scalable denoising.

2.1. Conditional Video Generation

Conditional video diffusion extends base text-to-video architectures by incorporating auxiliary control signals. Inspired by the spatial conditioning of ControlNet [46], recent works have adapted similar mechanisms to the temporal domain [12, 16, 32], enabling guidance through depth maps, motion vectors, and camera parameters. Concurrently, video-to-video (V2V) editing methods focus on propagating edits across frames while preserving the features of the source video [4, 11, 18, 25, 29, 38, 40]. Many such approaches leverage DDIM inversion to facilitate appearance modifications [5, 22, 26]. However, these methods are fundamentally designed for local appearance changes; they struggle with non-local motion edits where the structural correspondence between frames is disrupted. When motion patterns are altered, the temporal alignment assumptions underlying these inversion-based approaches are violated.

2.2. Motion-Guided Video Generation

Motion control has emerged as a critical research direction, broadly categorized into trajectory-based and optical-flow-based methods. Trajectory-based approaches condition generation on point trajectories [8, 9, 21, 23, 30, 35, 39, 41, 43, 47, 48], granting precise control over object paths, camera movement, and complex interactions. Conversely, optical flow-based methods [19, 20] utilize dense correspondence priors derived from optical flow estimators and point trackers [6, 7, 17, 34] to achieve fine-grained motion transfer.

Despite their impressive capabilities, these methods operate primarily as *generators* rather than editors. Instead of modifying an input video directly, they extract attributes (e.g., optical flow) to condition the synthesis of an entirely new video. Recent trajectory-based methods [3, 10, 37] attempt to bridge this by conditioning on single images and motion trajectories. However, while powerful for content creation, they fail to preserve the unrevealed visual context of existing videos when motion is modified. First-frame preserving methods like ReVideo [24] attempt to address this via inpainting but degrade when camera motion reveals content absent from the initial frame.

Our method addresses these limitations to enable true video-to-video motion editing. Specifically, we allow for flexible modification of object and camera trajectories while rigorously preserving the remaining video content. This approach generalizes effectively to arbitrary objects, diverse camera motions, and complex multi-element scenes.

3. Our Approach

In this section, we present a video-to-video motion editing framework that integrates a motion description mechanism with a video diffusion model. Our approach enables four core capabilities: object motion (altering movement while preserving static backgrounds, e.g., moving a dog but not the scene); camera control (simultaneously manipulating object and camera perspective, e.g., panning while an object moves); temporal control (adjusting trajectory timing, e.g., delaying an action to the 5th second); and arbitrary frame specification (applying edits across any frame span).

We demonstrate that explicitly defining 'motion edits', which explicitly describe the desired change in motion, enables our system to robustly support these tasks. We first outline these capabilities in detail, followed by our key technical contributions: the motion counterfactual video generation method and our specialized video-to-video architecture.

3.1. Editing Video through Motion Edits

Moving Objects By identifying an object's trajectory and editing it, we can change the motion of the object as the video progresses. As shown in Figure 1 and 2, this can have high level effects such as changing the ultimate outcome of a scenario and is a flexible tool for many applications, such as re-timing subjects, improve video aesthetics by moving occluders, or recomposing a video and its parts in motion.

Camera Control With our motion editing scheme we can control camera pose and motion in the video relative to the scene. We estimate a dynamic pointmap [45], reproject it into each frame using user-specified camera extrinsics and intrinsics, and then solve for deviations in the pointwise trajectories. This allows us to dynamically change the position and focal length of the camera in any frame while also preserving the video content. We show this in the Swan example in Figure 1 where the Swan is swimming and the ripples in the water are preserved despite changes in the camera position and in 2 where each frame has a different zoom level with same scene content.

Time Control Our method allows users to control trajectories of specific elements in a video independent of the global timeline. This enables delaying or accelerating an object's trajectory, such as making a subject appear on second 5 instead of second 2, while preserving the background's original motion. As shown in Figure 2, we can delay the appearance of a duck until later frames, effectively decoupling the

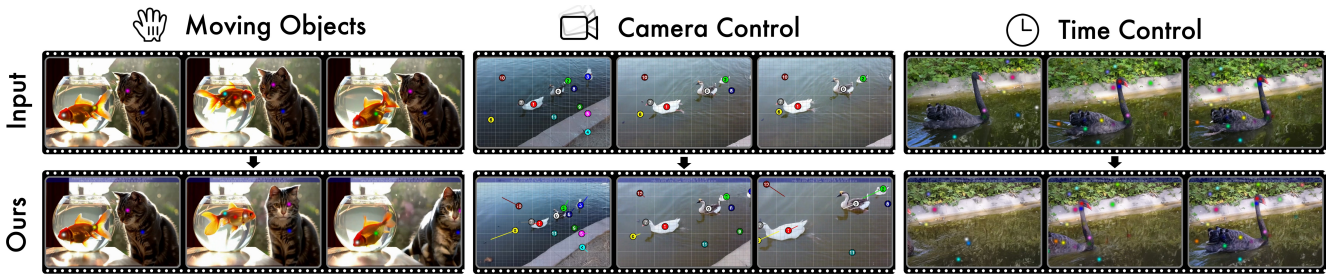

Figure 2. From left to right respectively, **Cat Fish.** In the edited video, the cat moves away from the bowl. **Camera control.** In the edited video, the first frame is zoomed out, middle frame is identical, the last frame is zoomed in. **Duck Zoom.** The edited video exhibits different content for a given frame (time) than the original, e.g. in the edited video, the duck is not visible in the first frame whereas it is visible in the original.

subject's timeline from that of the scene.

Arbitrary Frame Specification Unlike image-to-video approaches that rely on the first frame for content generation [3, 10, 24], our framework supports editing objects that appear at any point in the video. Relying on the initial frame severely restricts possible edits and fails to account for elements that emerge later. Additionally, the motion of the rest of the video is entirely hallucinated whereas ours can conserve it in part or entirely. By conditioning on the full video, we enable precise control over mid-stream objects, such as the stop sign in Figure 3.

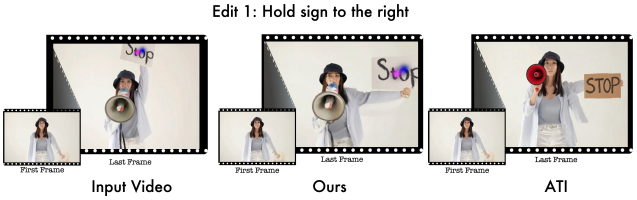

Figure 3. **Controlling Content on Any Frame.** By conditioning on the full video, we can move and preserve content appearing on any frame. Methods like ATI rely on the first frame, failing to control objects, like the sign, that emerge mid-sequence.

3.2. Motion Counterfactual Video Generation

Our approach requires training data consisting of video pairs with identical visual content but different motion patterns. We generate these *motion counterfactual videos* V_{cf} and corresponding *target videos* V_{target} from raw videos using a systematic process that ensures trackable point correspondences between video pairs (Figure 4).

Given a source video V_{full} of length F_{full} frames, we generate video pairs as follows. First, we extract the target video V_{target} by selecting a contiguous frame chunk of length F with random starting frame $f_{\text{start}} \sim \text{Uniform}(0, F_{\text{full}} - F)$. We keep real video as targets to ensure the model trains toward realistic motion and appearance.

For the counterfactual video V_{cf}, we randomly select start and end frame indices $f^{\text{cf}}_{\text{start}}, f^{\text{cf}}_{\text{end}} \sim \text{Uniform}(0, F_{\text{full}} - 1)$ and choose one of two generation strategies:

Frame Interpolation: We use a video diffusion model

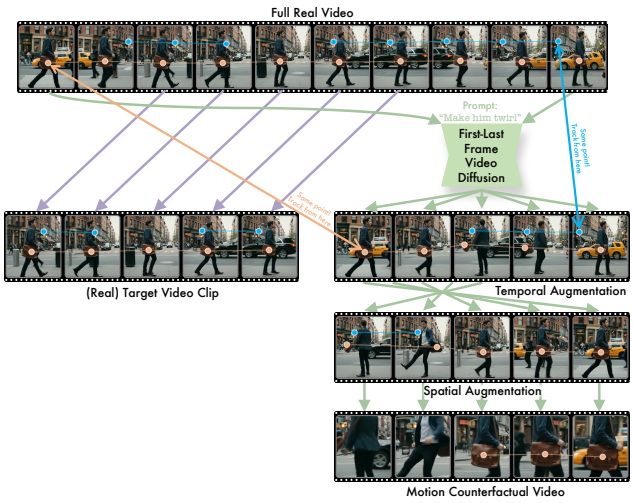

Figure 4. **Counterfactual data generation process.** In order to generate a real / counterfactual video pair and its corresponding trajectories, we take a full real video, extract a video clip, then create a counterfactual video. The counterfactual has new motion from the video generator, as well as temporal and spatial augmentations. In order to ensure we have two corresponding set of tracks, we specifically use the first and last frames, which directly match the original video, to anchor the tracks for the counterfactual.

conditioned on frames $f^{\text{cf}}_{\text{start}}$ and $f^{\text{cf}}_{\text{end}}$ to generate a F-frame video. This adds new content via LLM-generated prompts—e.g., instructing a walking person to "twirl" (Figure 4). This provides more data than first-frame-only methods, allowing the model to use more of the input video.

Temporal Resampling: We extract F frames evenly spaced between $f^{\text{cf}}_{\text{start}}$ and $f^{\text{cf}}_{\text{end}}$ from V_{full}. This creates natural speed variations, temporal shifts, and sometimes reversed motion when $f^{\text{cf}}_{\text{start}} > f^{\text{cf}}_{\text{end}}$.

Next, we establish point correspondences between the video pair. We initialize $N \sim \text{Uniform}(1, 64)$ tracking points with coordinates (t_i, x_i, y_i) where x_i and y_i sampled uniformly from frame dimensions $W_{\text{rgb}}, H_{\text{rgb}}$ respectively. For temporal coordinates t_i:

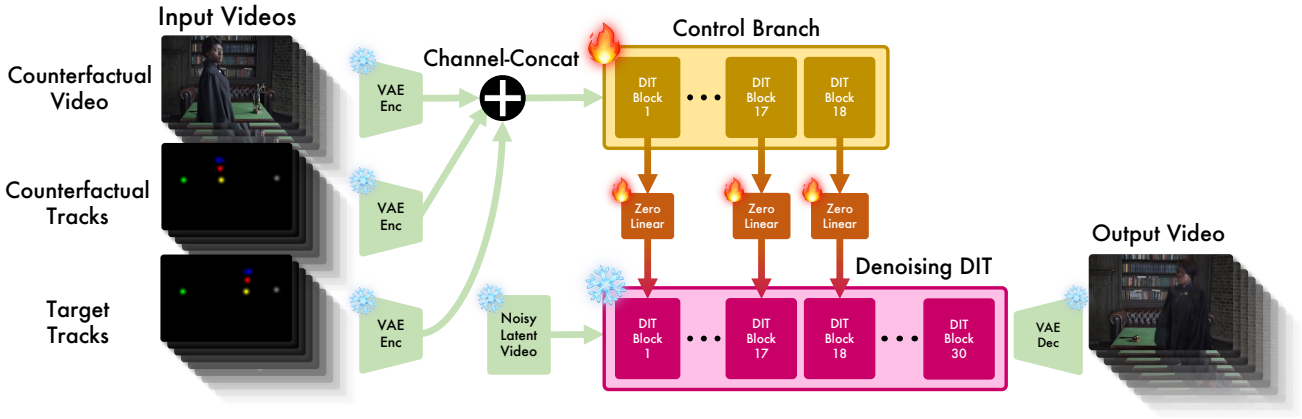

Figure 5. **Our motion-conditioned video diffusion architecture.** We extend a T2V DiT model with a control branch that processes three additional video conditioning channels: the counterfactual video, counterfactual motion tracks, and target motion tracks. The control branch duplicates the first 18 transformer blocks and integrates with the main branch through zero-initialized MLPs, similar to ControlNet.

- **Temporal resampling:** Frame indices are sampled from frames present in both V_{target} and V_{cf}
- **Frame interpolation:** Frame indices are restricted to $\{f^{\text{cf}}_{\text{start}}, f^{\text{cf}}_{\text{end}}\}$ to ensure correspondence

We use TAPNext [50], a bidirectional point tracker, on V_{full} with these initial points to obtain target tracks T_{target}. For counterfactual tracks T_{cf}: in temporal resampling cases, we use the same tracker output; for interpolation cases, we first replace the corresponding frames in V_{full} with the interpolated V_{cf} frames before running TAPNext [50].

Finally, we apply geometric augmentations to the counterfactual videos including random sliding crops, rotations, and scale changes, with the same transformations applied to the corresponding tracking points to maintain correspondence. These artificial moving crops approximate multi-view videos and ensure perfect temporal synchronization—giving the model a bias toward synchronizing appearance when otherwise unspecified.

Trajectory Representation A key part of our method is our representation of motion throughout videos. Our model is conditioned on three videos: the counterfactual video V_{cf}, the rendered counterfactual motion tracks B_{cf}, and the rendered target motion tracks B_{target}, each of dimension $\mathbb{R}^{F\times 3\times H_{\text{rgb}}\times W_{\text{rgb}}}$. Additionally, like our base model, text prompts provide semantic conditioning, which we will leave out of equations in this section for brevity.

We rasterize the tracking information as colored Gaussian blobs on black backgrounds to create motion conditioning channels. For each training sample, we randomly select N distinct random colors. Each tracking point is rendered as a Gaussian blob with standard deviation of 10 pixels in its assigned color in both the counterfactual tracks video B_{cf} and target tracks video B_{target}, with blobs only drawn when the corresponding point is visible (not occluded) as reported by the point tracker. We also tried representations similar to [12], but found that both large number of points and the lack of distinct colors made it a weaker control signal.

The tracks are subject to dropout during training, with target motion blobs T_{target} experiencing higher dropout rates than conditioning tracks T_{cf} to improve robustness and prevent overfitting to specific motion patterns. During inference, we limit the number of point correspondences to approximately 20, as the model fails to follow all correspondences when given too many points.

3.3. Model Architecture

We use a pre-trained T2V DiT as our base model [42]. In order to condition on motion and input videos we incorporate a control branch duplicating the first 18 transformer blocks of the DiT that feeds into the main branch using zero-initialized MLPs. Conceptually the control branch is similar to a ControlNet [46] applied to a DiT architecture. We are inspired by the architecture proposed in DiffusionAsShader [12], but our implementation has the key difference of allowing conditioning on three video tracks and using a control branch patchifier that handles $48 = 3 \times 16$ input channels for the three conditioning videos in latent space.

The control branch tokens are fed through zero-init [46], channel-wise MLPs and then added to the main branch token values in their respective transformer blocks. The base model processes the noisy video being denoised along with text conditioning, while our control branch handles the three additional video conditioning channels $V_{\text{cf}}, B_{\text{cf}}, B_{\text{target}}$. All video inputs are encoded using a 3D Causal VAE [42], which compresses RGB videos of shape $F\times 3\times H_{\text{rgb}}\times W_{\text{rgb}}$ to latent representations of shape $F_{latent}\times C_{\text{latent}}\times H_{\text{latent}}\times W_{\text{latent}}$ where $C_{\text{latent}} = 16, F_{latent} = \left(\frac{F-1}{4}+1\right), W_{\text{latent}} =$

$\frac{W_{\text{rgb}}}{8}, H_{\text{latent}} = \frac{H_{\text{rgb}}}{8}$. The main branch is frozen while the control branch is trained.

During training, the model learns to generate target videos V_{target} that follow specified motion patterns and satisfy the given correspondences between counterfactual and target tracks. The training objective is conditioned on the counterfactual video V_{cf}, its tracks T_{cf}, the target tracks T_{target}, and a text prompt describing the scene. This formulation successfully teaches the model to transfer motion patterns from the target tracks while maintaining the visual realism of target video content.

The task we tackle is harder than a typical ControlNet task where the structure is usually given to the model. For example an edge-to-image ControlNet has a good idea of what the structure of the output should be with edges as input. Surprisingly, our adapter works despite the inputs (video + motion blobs) lacking spatiotemporal synchronization with the output. We hypothesize that transformer blocks do non-trivial work to achieve this capability.

4. Results

4.1. Implementation Details

We use CogVideoX-5B [42] as our base text-to-video model for both the finetuned counterfactual video generation model and the V2V editing model. Training was conducted on 8 H100 GPUs for one week using standard latent diffusion training with L2 loss. We set $F = 49$, with input resolution of 480×720 pixels, corresponding to latent dimensions of 60×90. We use N varying between 1 and 64 during training and set the control branch depth appropriately. We use a learning rate of 10^{-4} and a dataset size of $100,000$ videos, for $15,000$ iterations with an effective batch size of 32. We use an internal video dataset with 500,000 samples.

We evaluate our motion editing approach through user studies and quantitative metrics, comparing against state-of-the-art motion control methods.

4.2. User Study

We conducted a user study comparing our method against three baselines: ATI [37], a trajectory-guided image-to-video method based on WAN 2.1 [36]; ReVideo [24]; and Go-with-the-Flow (GWTF) [3]. We manually created 20 test videos spanning diverse scenarios including object motion editing, camera motion changes, and complex scenes with multiple moving elements. 41 participants compared all four methods using the interface shown in the Supplementals, selecting the best video for each of three questions per test case:

- **Q1:** "Which video better preserves the input video's content?"
- **Q2:** "Which video better reflects the desired motion?"
- **Q3:** "Which video is overall a better edit of the input video?"

Question	Ours	ATI	ReVideo	GWTF
Q1: Content (↑)	**70%**	24%	1%	5%
Q2: Motion (↑)	**71%**	24%	2%	3%
Q3: Overall (↑)	**69%**	25%	1%	5%

Table 1. **User study win rates across all methods.** Participants selected the best video for each question. Our method consistently wins across all evaluation criteria.

Table 1 show that users consistently ranked our method highest across all questions, with win rates around 70% compared to 25% for ATI and less than 5% for ReVideo and GWTF, demonstrating superior content preservation and motion control.

4.3. Quantitative Evaluation

We developed a quantitative evaluation protocol using photometric reconstruction error to assess motion editing quality.

4.3.1. Dataset Construction

We curated a dataset of $N_{\text{test}} = 100$ test videos using the following protocol. Given a source video V_{test} of length F_{full} frames, we split it temporally at the midpoint to obtain $V_0 = V_{\text{test}}[1 : F_{\text{full}}/2]$ and $V_1 = V_{\text{test}}[F_{\text{full}}/2 : F_{\text{full}}]$. We then create the counterfactual input by temporally reversing V_1 to get V_1', ensuring temporal continuity between V_0 and V_1' (i.e., the last frame of V_0 matches the first frame of V_1').

We selected random internet videos not seen during training where significant content appears in middle frames but is not visible in the first frame. To quantify this, we initialize $N_{\text{points}} = 25$ tracking points at the temporal midpoint of each video and track them bidirectionally using TAPNext [50]. We retain only videos where a substantial number of points become occluded when tracked to both the first and last frames.

4.3.2. Evaluation Protocol

For each test case, we use V_0 as input and V_1 as the target video. We provide both our method and ATI with identical motion trajectories extracted from V_1 and measure reconstruction quality using frame-wise L2 loss:

$$L_2 = \frac{1}{F} \sum_{i=1}^{F} \| I_i^{\text{pred}} - I_i^{\text{target}} \|_2^2 \tag{1}$$

where F is the number of frames, I_i^{pred} is the i-th predicted frame, and I_i^{target} is the corresponding target frame.

Our method achieves substantially lower reconstruction error (Table 2), confirming that our full-video approach better preserves content compared to first-frame generation

Figure 6. **Iterative editing.** Outputs can become inputs for subsequent edits, enabling complex sequential motion changes. Yellow dots used for first edit, green/cyan for second. Arrows added from old to new position for ease of visualization.

Method	L_2 (↓)	SSIM (↑)	LPIPS (↓)
Ours	**0.024**	**0.098**	**0.031**
ATI	0.038	0.094	0.072
Go-with-the-Flow	0.067	0.089	0.088
ReVideo	0.096	0.080	0.106

Table 2. Evaluation of photometric reconstruction error for our method and ATR. Our method achieves significantly lower L2 reconstruction error.

methods, particularly in scenarios involving content not visible in initial frames.

4.4. Qualitative Comparisons

Iterative Edits. One of the strengths of our technique is that it can be applied iteratively - taking the output of one run and using it as input for a successive video edit. This allows users to chain multiple simple, intuitive edits together in order to achieve a very complicated edit. This iterative editing also provides more immediate feedback to the user making the process more transparent and easier to control. In Fig. 6, we show that a complex edit (an object motion and a camera change) can be decomposed into its core parts and applied one by one. While this example demonstrates some degree of subject drift, this can be attributed in part to the quality of the base video model. We believe that future versions of our method will be able to be applied infinitely.

Baseline comparisons. Figure 7 compares our method against several baselines in multiple video editing scenarios, each of which demonstrate the capabilities of our motion edits. We primarily compare against ATI [37], a trajectory-guided image-to-video method based on WAN 2.1 [36], our strongest baseline despite using a more powerful base model than our CogVideoX base. Subfigure 4 additionally shows ReVideo [24] and Go-with-the-Flow [3], which rated poorly in user evaluation—ReVideo lacks text conditioning and Go-with-the-Flow was not designed for point control.

Edit #1: Complex Edits on the Boat Scene. This edit moves the boat left and shifts the camera so that mountains from the original's last frame appear in the edit's first. This requires specifying a substantial temporal trajectory change and holistic knowledge of the scene content. Ours is the only method that realistically moves the boat while correctly adjusting the camera to reveal the mountains at the beginning of the video.

Edit #2: Reposing a Cheerleader. This edit raises the cheerleader's arms. The challenge involves preserving the red pom-pom, which is absent from the first frame. Ours successfully modifies the motion while retaining this content. In contrast, ATI and ReVideo rely solely on the first frame, leading to unnatural movements and a failure to preserve the pom-pom.

Edit #3: Move The Bicyclist. This edit controls a cyclist visible only in the final frame of the original video. Ours correctly propagates the cyclist and tracking dots (cyan, magenta, white) throughout. ATI, lacking full temporal context, misplaces the cyclist and synthesizes wrong buildings (red circles).

Edit #4: Dog Race. Differential timing breaks single-frame-based methods. We decelerate the Corgi (green dot) to reverse the race outcome while keeping the Bichon steady. This requires independent temporal control; ATI fails to decouple the motions, incorrectly copying the Bichon and transforming a light pole into a tree.

Edit #5: Moving Static Balloons. We add upward motion to stationary balloons. The white balloon (white dot), which appears mid-video, challenges partial information methods. While ATI moves visible balloons, it renders the initially hidden white balloon orange due to missing appearance data. Our method uses full video context to maintain correct colors.

Edit #6: Zooming out on the Swan In this DAVIS [28] example, we transform a panning shot into a static, zoomed-out view. The output field of view differs entirely from the input, yet the swan must remain anchored to specific vegetation. Lacking full spatial context, ATI synthesizes a second swan and produces inconsistent motion.

Edit #7: Retiming a taxi. We do a complex isolated retiming of taxi and truck movement. This requires complete temporal understanding; ATI's single-frame generation cannot achieve this reversal. Figure 7 compares our method against ATI (WAN 2.1-based), as well as ReVideo and Go-with-the-Flow (in Subfigure 4), both of which were rated poorly due to their design limitations.

Edit #8: Moving an Offscreen Car. As the camera follows a red car, a motorcyclist enters late. We reposition this initially invisible rider behind the car while maintaining consistent background architecture. Lacking future frames to reference the rider and buildings (red circles), ATI synthesizes incorrect content.

Figure 7. Comparison of our method vs. baselines across eight challenging motion editing scenarios. Each row shows a different editing task with input video, our result, and ATI's result (with additional baselines shown for subfigure 4). **Icon key:** Human Pose (modifying human motion), Move Object (repositioning objects), Move Camera (changing camera motion), Time Control (retiming events), Changed All Frames (no shared frames between input/output—impossible for image-to-video methods). Colored dots track correspondence points throughout the video; dot presence/absence indicates object visibility. Red circles highlight key differences where baselines fail.

Discussion. These scenarios highlight I2V limitations: conditioning only on the first frame prevents leveraging information from the full input. Our V2V formulation enables bidirectional flow, allowing outputs to pull content from *any* input frame. This handles offscreen content, camera changes, and reordering—challenges where I2V methods like ReVideo [24], Go-with-the-Flow [3], and Motion-Prompting [10] fail.

5. Conclusion

We developed a new video-to-video motion editing algorithm that allows us to edit the motion of objects, subjects and camera pose in user-provided videos. To the best of our knowledge it is the first in its class, compared to other work that control motion in the image-to-video setup. Our algorithm has a comfortable user interface, where a user drags sparse point trajectories to control objects or camera motion.

References

[1] Jianhong Bai, Menghan Xia, Xintao Wang, Ziyang Yuan, Xiao Fu, Zuozhu Liu, Haoji Hu, Pengfei Wan, and Di Zhang. ReCamMaster: Camera-controlled generative rendering from a single video, 2025.

[2] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, et al. Lumiere: A space-time diffusion model for video generation. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024.

[3] Ryan Burgert, Yuancheng Xu, Wenqi Xian, Oliver Pilarski, Pascal Clausen, Mingming He, Li Ma, Yitong Deng, Lingxiao Li, Mohsen Mousavi, Michael Ryoo, Paul Debevec, and Ning Yu. Go-with-the-flow: Motion-controllable video diffusion models using real-time warped noise, 2025.

[4] Duygu Ceylan, Chun-Hao Paul Huang, and Niloy J. Mitra. Pix2Video: Video editing using image diffusion. In *ICCV*, 2023.

[5] Wenhao Chai, Xun Guo, Gaoang Wang, and Yan Lu. StableVideo: Text-driven consistency-aware diffusion video editing. In *ICCV*, 2023.

[6] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. TAPIR: Tracking any point with per-frame initialization and temporal refinement. In *ICCV*, 2023.

[7] Carl Doersch, Pauline Luc, Yi Yang, Dilara Gokay, Skanda Koppula, Ankush Gupta, Joseph Heyward, Ignacio Rocco, Ross Goroshin, Joao Carreira, and Andrew Zisserman. BootsTAP: Bootstrapped training for tracking-any-point, 2024.

[8] Wanquan Feng, Tianhao Qi, Jiawei Liu, Mingzhen Sun, Pengqi Tu, Tianxiang Ma, Fei Dai, Songtao Zhao, Siyu Zhou, and Qian He. I2VControl: Disentangled and unified video motion synthesis control, 2024.

[9] Xiao Fu, Xian Liu, Xintao Wang, Sida Peng, Menghan Xia, Xiaoyu Shi, Ziyang Yuan, Pengfei Wan, Di Zhang, and Dahua Lin. 3DTrajMaster: Mastering 3D trajectory for multi-entity motion in video generation, 2024.

[10] Daniel Geng, Charles Herrmann, Junhwa Hur, Forrester Cole, Serena Zhang, Tobias Pfaff, Tatiana Lopez-Guevara, Carl Doersch, Yusuf Aytar, Michael Rubinstein, Chen Sun, Oliver Wang, Andrew Owens, and Deqing Sun. Motion prompting: Controlling video generation with motion trajectories, 2024.

[11] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. TokenFlow: Consistent diffusion features for consistent video editing. In *ICLR*, 2024.

[12] Zekai Gu, Rui Yan, Jiahao Lu, Peng Li, Zhiyang Dou, Chenyang Si, Zhen Dong, Qifeng Liu, Cheng Lin, Ziwei Liu, Wenping Wang, and Yuan Liu. Diffusion as shader: 3D-aware video diffusion for versatile video generation control, 2025.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[14] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models, 2022.

[15] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *NeurIPS*, 2022.

[16] Zhihao Hu and Dong Xu. VideoControlNet: A motion-guided video-to-video translation framework, 2023.

[17] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker3: Simpler and better point tracking by pseudo-labelling real videos, 2024.

[18] Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2Video-Zero: Text-to-image diffusion models are zero-shot video generators. In *ICCV*, 2023.

[19] Mathis Koroglu, Hugo Caselles-Dupré, Guillaume Jeanneret, and Matthieu Cord. OnlyFlow: Optical flow based motion conditioning for video diffusion models, 2024.

[20] Guojun Lei, Chi Wang, Hong Li, Rong Zhang, Yikai Wang, and Weiwei Xu. AnimateAnything: Consistent and controllable animation for video generation, 2024.

[21] Yaowei Li, Xintao Wang, Zhaoyang Zhang, Zhouxia Wang, Ziyang Yuan, Liangbin Xie, Yuexian Zou, and Ying Shan. Image conductor: Precision control for interactive video synthesis, 2024.

[22] Jun Hao Liew, Hanshu Yan, Jianfeng Zhang, Zhongcong Xu, and Jiashi Feng. MagicEdit: High-fidelity and temporally coherent video editing, 2023.

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

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

[25] Hao Ouyang, Qiuyu Wang, Yuxi Xiao, Qingyan Bai, Juntao Zhang, Kecheng Zheng, Xiaowei Zhou, Qifeng Chen, and Yujun Shen. CoDeF: Content deformation fields for temporally consistent video processing. In *CVPR*, 2024.

[26] Wenqi Ouyang, Yi Dong, Lei Yang, Jianlou Si, and Xingang Pan. I2VEdit: First-frame-guided video editing via image-to-video diffusion models, 2024.

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

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

[29] 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 *ICCV*, 2023.

[30] Haonan Qiu, Zhaoxi Chen, Zhouxia Wang, Yingqing He, Menghan Xia, and Ziwei Liu. FreeTraj: Tuning-free trajectory control in video diffusion models, 2024.

[31] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[32] Xiaoyu Shi, Zhaoyang Huang, Fu-Yun Wang, Weikang Bian, Dasong Li, Yi Zhang, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Qifeng Chen. Motion-I2V: Consistent and controllable image-to-video generation with explicit motion modeling, 2024.

[33] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data, 2022.

[34] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020.

[35] Yao Teng, Enze Xie, Yue Wu, Haoyu Han, Zhenguo Li, and Xihui Liu. Drag-a-video: Non-rigid video editing with point-based interaction, 2023.

[36] 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, 2025.

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

[38] Jiangshan Wang, Yue Ma, Jiayi Guo, Yicheng Xiao, Gao Huang, and Xiu Li. COVE: Unleashing the diffusion feature correspondence for consistent video editing. In *NeurIPS*, 2024.

[39] 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.

[40] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *ICCV*, 2023.

[41] 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, 2024.

[42] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Xiaotao Gu, Yuxuan Zhang, Weihan Wang, Yean Cheng, Xu Bin, Yuxiao Dong, and Jie Tang. CogVideoX: Text-to-video diffusion models with an expert transformer, 2024.

[43] Shengming Yin, Chenfei Wu, Jian Liang, Jie Shi, Houqiang Li, Gong Ming, and Nan Duan. DragNUWA: Fine-grained control in video generation by integrating text, image, and trajectory, 2023.

[44] David Junhao Zhang, Roni Paiss, Shiran Zada, Nikhil Karnad, David E. Jacobs, Yael Pritch, Inbar Mosseri, Neal Wadhwa, Nataniel Ruiz, Mike Zheng Shou, Sagie Benaim, Yijun Li, and Kfir Aberman. ReCapture: Generative video camera controls for user-provided videos, 2024.

[45] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825*, 2024.

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

[47] Zhenghao Zhang, Junchao Liao, Menghao Li, Long Qin, and Weizhi Wang. Tora: Trajectory-oriented diffusion transformer for video generation, 2024.

[48] Zhiyuan Zhang, Can Wang, Dongdong Chen, and Jing Liao. FlexTraj: Image-to-video generation with flexible point trajectory control, 2024.

[49] 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, 2024.

[50] Artem Zholus, Carl Doersch, Yi Yang, Skanda Koppula, Viorica Patraucean, Xu Owen He, Ignacio Rocco, Mehdi S. M. Sajjadi, Sarath Chandar, and Ross Goroshin. Tapnext: Tracking any point (tap) as next token prediction, 2025.

MotionV2V: Editing Motion in a Video

Supplementary Material

6. Human Interaction

Figure 8. **Interface for creating motion edits.** The red arrow shows the transformation from source trajectory (line) to target trajectory (triangle).

Our motion editing interface (Figure 8) provides an intuitive way to specify complex motion changes without requiring any laborious segmentation or rotoscoping. Users simply click points on the video to initialize points which are then tracked bidirectionally to create source trajectories. Then, the user manipulates these trajectories using Bezier splines to define the desired target motion, with arrows indicating the transformation from source to target.

The red arrow in the figure illustrates a typical edit, showing the transformation from the source trajectory (line) to the target trajectory (triangle). The interface allows users to scrub through video frames and place trajectory points as needed. Different tracking points are represented by different colors: red, green, blue, cyan, magenta, yellow, and white.

7. User Study

As described in Section 4, we conducted a user study with 41 participants evaluating 20 test videos to compare our method against state-of-the-art baselines.

Evaluation Protocol Participants used the interface shown in Figure 10 to compare our method against three baselines: ATI [37], ReVideo [24], and Go-with-the-Flow [3]. The interface presents side-by-side comparisons of the original input video alongside results from our method and all baselines, with colored tracking dots visualizing the motion edits so users can clearly see how each method interprets the desired motion changes. For each test case, participants were asked three questions:

- **Q1:** "Which video better preserves the input video's content?"
- **Q2:** "Which video better reflects the desired motion?"
- **Q3:** "Which video is overall a better edit of the input video?"

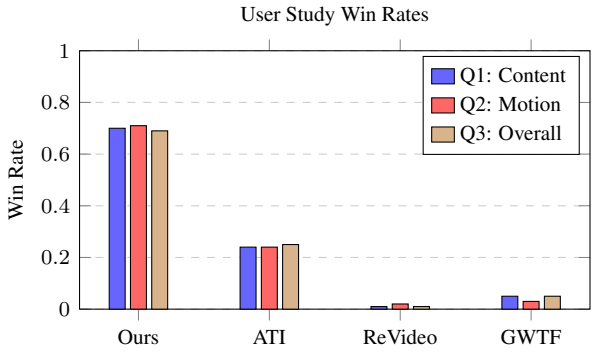

Figure 9. User study win rates per question (see Table 1 for values).

The results, visualized in Figure 9 and detailed in Table 1, show users greatly preferred our method, with rates around 70% compared to approximately 25% for ATI and less than 5% for ReVideo and Go-with-the-Flow.

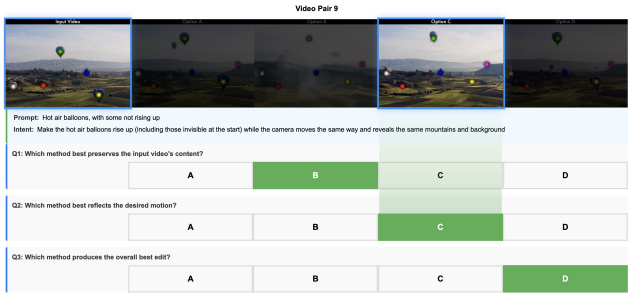

Figure 10. **User study evaluation interface.** Side-by-side video comparisons with visualized motion edits.

8. Quantitative Evaluation Dataset

Our quantitative evaluation dataset construction is fully described in Section 4 (Dataset Construction subsection). We created $N_{\text{test}} = 100$ test videos by splitting videos at their temporal midpoint and reversing one half to create video pairs with common starting frames (Figure 11). This protocol ensures that image-to-video baselines, which require

first-frame correspondences, receive inputs they can properly handle since the tracking points match the first frame. The dataset specifically includes videos where significant content appears in middle frames but not in the first frame, quantified by tracking $N_{\text{points}} = 25$ points bidirectionally from the midpoint.

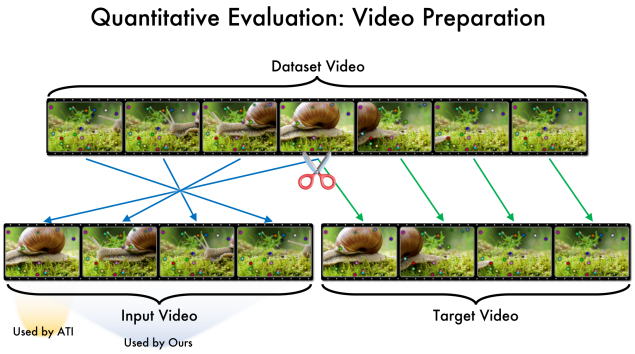

Figure 11. **Test Data Generation.** A video is separated at the middle, and then one half is reversed. This results in two videos with a common starting frame.

9. Baseline Implementation Details

All baseline methods are image-to-video (I2V) algorithms with motion control, fundamentally different from our video-to-video (V2V) approach:

ATI [37] ATI is based on Wan2.1. It is a point-based image-to-video algorithm with controllable motion. For our baseline, we take the target tracks and apply it to the first frame of our counterfactual videos, using the target prompts for text guidance. We use the default number of diffusion steps and CFG as provided by their public code repository.

ReVideo [24] ReVideo is based on Stable Video Diffusion. It takes no prompt as an input. It is an image-to-video point-based motion-controllable algorithm, with the ability to specify editable regions. Since we are avoiding manual labor such as rotoscoping we designate the entire video as an editable region.

Go-with-the-Flow [3] Go-with-the-Flow is not a point-based motion control algorithm, but is instead an image-to-video motion-controllable algorithm driven by warped noise, which often comes from optical flow. To make this baseline work, we run the raterized target tracks through RAFT to get optical flows, and from that create warped noise that is used to generate the output videos. We use a more recent Wan2.2-based version of Go-with-the-Flow to test with, as it is a tougher baseline than their original CogVideoX-5B model.

The fundamental limitation of all these baselines is their I2V formulation: they can only access information from the first frame, preventing them from handling content that appears later in the video, camera viewpoint changes, or complex temporal reordering. Our V2V approach, in contrast, can leverage information from any frame of the input video.

10. Ablations

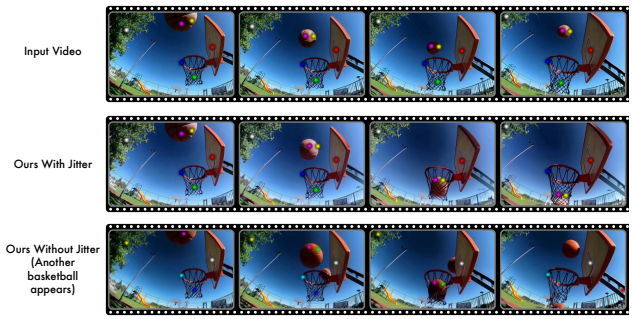

Figure 12. **The effects of trajectory jitter on motion editing.** Top: without jitter, a second basketball appears. Bottom: with 1-2 pixel jitter, the edit follows correctly.

We discovered an interesting phenomenon during inference: when tracking points are pixel-perfectly aligned with the input video trajectories across multiple frames, the model exhibits a strong bias toward reproducing the original video's semantics rather than following the edited motion.

Figure 12 illustrates this effect. In the top row (without jitter), when tracking points are pixel-perfectly aligned with the input video, the model exhibits a bias toward reproducing the original video's semantics. Although the basketball successfully goes through the hoop following the edited trajectory, a second basketball mysteriously appears behind the hoop to match the original video where the basketball passes in front. This occurs because the pixel-perfect alignment of other tracking points signals to the model that it should preserve the content of the entire input video, which is often the only case where the points are aligned that perfectly during training. In the bottom row (with jitter), this identity-copying behavior is eliminated and the edit follows the intended motion correctly.

To address this, we introduce a simple but effective inference-time technique: "Jitter". We add small random noise $\epsilon \sim \mathcal{U}(-2, 2)$ pixels to the (x, y) positions of all tracking points at each frame. Importantly, this is an inference-time modification only—the model is not trained with this jitter. This minimal perturbation (1-2 pixels) is imperceptible in the rendered tracks but sufficient to break the model's tendency to copy the input video's identity, allowing it to follow the edited trajectories more faithfully.

11. Future Work

In future work, we consider creating large-scale synthetic datasets with precise motion counterfactuals made with 3d software. While our current approach leverages real videos paired with diffusion-generated conuterfactuals, synthetic 3d data would provide perfect ground truth motion-edit pairs, enabling exact control over individual object trajectories, physical interactions, and the resulting lighting and shading changes. This would improve the precision of our training dataset, possibly allowing even less points to be used for control.