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?"

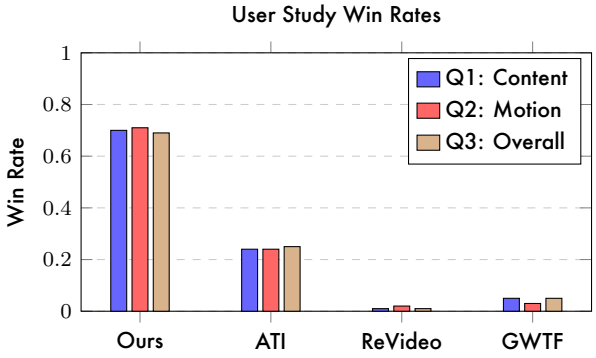

Figure 9. User study win rates per question (see Table 1 for values).

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.

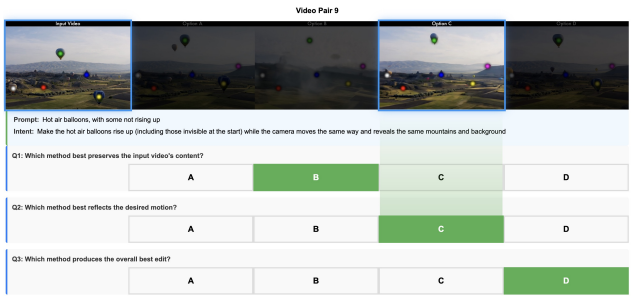

Figure 10. **User study evaluation interface.** Side-by-side video comparisons with visualized motion edits.

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.

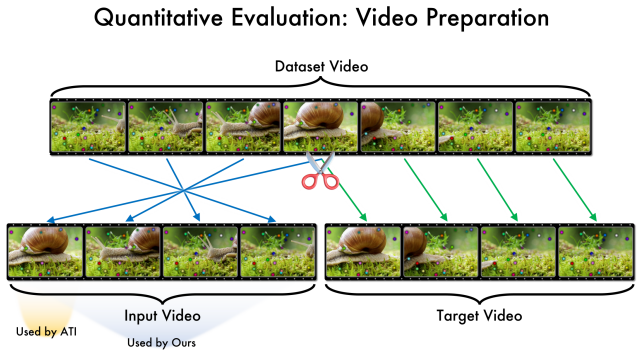

Figure 11. **Test Data Generation.** A video is separated at the middle, and then one half is reversed. This results in two videos with a common starting frame.

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

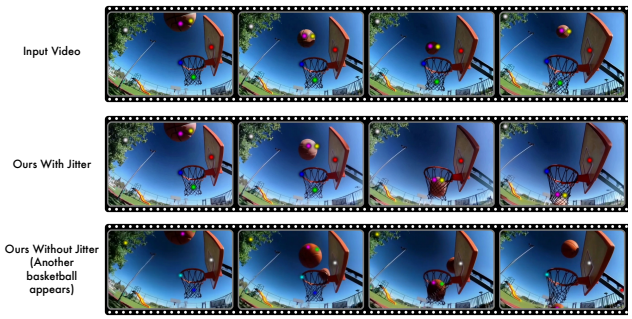

Figure 12. **The effects of trajectory jitter on motion editing.** Top: without jitter, a second basketball appears. Bottom: with 1-2 pixel jitter, the edit follows correctly.

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.