DOI: 10.1111/cgf.70362

EUROGRAPHICS 2026 / B. Masia and J. Thies
(Guest Editors)

COMPUTER GRAPHICS *forum*
*Volume 45 (2026), Number 2*

# MultiCOIN: Multi-Modal COntrollable INbetweening

M. Tanveer[1,2], Y. Zhou[2], S. Niklaus[2], A. Mahdavi Amiri[1], H. Zhang[1], K. K. Singh[2] and N. Zhao[2]

[1]Simon Fraser University
[2]Adobe Research

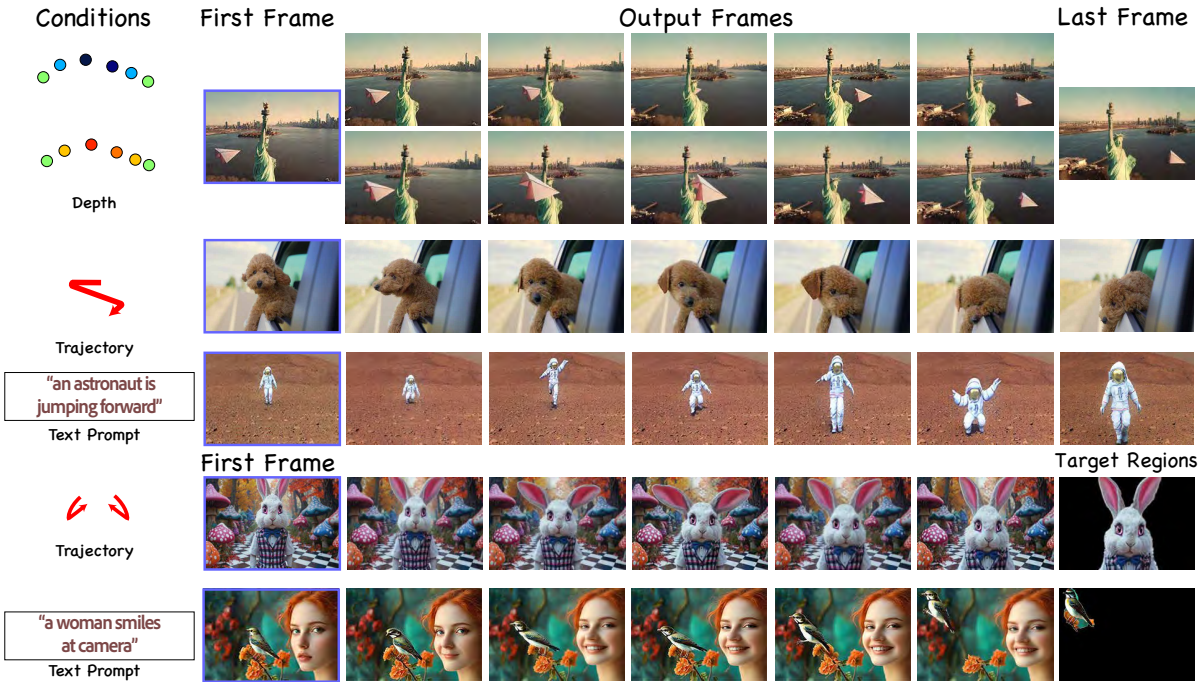


**Figure 1:** *MultiCOIN takes a start and end image frame to generate an interpolative video inbetweening. It supports multi-modal controls, including depth change and layering, motion trajectories, text prompts, and target regions for movement localization, to generate smooth and plausible transitions. The controls can be used individually (top four rows) to create diverse results even with the same input pair (e.g., two depth layering results in top two rows). The controls can also be organized in a complementary way to ease the user's interactions. For example, target regions may be used for content control, while trajectory provides motion information. Also, while specifying the general movement of the woman by text, the user can exert accurate spatial control for the bird with target region.*

**Abstract**

*Video inbetweening creates smooth transitions between two frames making it an indispensable tool for video editing and long-form video synthesis. Existing methods struggle with large or complex motion and offer limited control over intermediate frames, often misaligning with user intent. We introduce MultiCOIN, a video inbetweening framework supporting multi-modal controls, including depth transitions and layering, motion trajectories, text prompts, and target regions for movement localization. It balances flexibility, usability, and fine-grained precision. Built on a Diffusion Transformer (DiT), due to its proven capability to generate high-quality long video, our model maps all motion controls into a unified sparse point-based representation compatible with the denoising process. Further, to respect the variety of controls which operate at varying levels of granularity and influence, we separate content and motion into two branches, enabling dedicated generators for each. A stage-wise training strategy ensures stable learning of multi-modal controls. Extensive experiments show improved motion complexity, controllability, and narrative consistency. Project Page: MultiCOIN.*

**CCS Concepts**

• ***Computing methodologies*** → *Computer vision tasks;*

## 1. Introduction

Video inbetweening or video interpolation seeks to generate intermediate frames between two end keyframes, creating a smooth transition from one scene to another. It is a long-standing problem [CLK00, HLK04a] and an increasingly important building block for video content creators and animators as they perform video editing, storytelling, and short-to-long video synthesis [MCD*18, SZY*21]. Such a frame interpolation is typically carried out in two steps: motion estimation and motion compensation [HLK04b, CLK00, NL20, RKT*22]. As temporal and spatial gaps between the input frames widen, both tasks are faced with significant challenges, since generating realistic intermediate frames requires synthesizing novel content to fill in and bridge the missing information, as well as resolving the inherent ambiguities therein. However, as the emerging generative models [BRL*23] become more powerful, the continuing growth of the space of exploration for the generated frames has opened up new possibilities for inbetweening of distant input scenes. At the same time, this poses a one-to-many problem, where a single output is typically insufficient since users are often not interested in just *any* possible video interpolation, but one which respects their artistic expression or creative mind. As a result, the inbetweening must be *user-controllable*.

Prior attempts on controllable video inbetweening, as exemplified by the recent work Framer [WWZ*24], have focused on respecting motion trajectories. In practice however, user controls are often more versatile and fine-grained. Recent advances in LLMs have popularized the use of text prompts as a means for edit controls. Even when confined to trajectory guidance only, additional constraints such as *depth transitions* (e.g., to specify whether an object moves in front of or behind another object, as shown in the top row of Fig. 1) and *region/object localization* (e.g., see the bottom two rows of Fig. 1 for the use of *target regions* to isolate the motioned object) must be incorporated to avoid ambiguities.

In this paper, we present MultiCOIN, for *MULTI-modal COntrollable INbetweening*, a novel unified video inbetweening framework for DiT, which can accommodate all the edit controls mentioned above, as shown in Fig. 1. Specifically, trajectory-based controls provide precise motion paths. Depth inputs can add 3D structure cues to help disambiguate non-lateral motions and improve occlusion handling. Furthermore, target regions can add motion localization constraints, ensuring consistency over detailed, especially multi-object, regions, while text-based control facilitates high-level semantic guidance. By combining all of these modalities in a unified latent space, our method strives to achieve a balance between flexibility, ease of use, and precision, empowering users to achieve high-quality and fine-grained interpolation with minimal effort.

To generate dynamic, accurate, and customizable motion transitions with multiple controls, we must build on an advanced video generative model. To this end, we resort to the Diffusion Transformer (DiT) architecture [PX23], due to its proven capability to generate high-quality long videos, which our method targets. The first challenge however, lies in making the multi-modal controls compatible with DiT. Unlike UNet, adopted by Framer for trajectory control only, DiT employs a Vision Transformer (ViT)-style 3D Variational Autoencoder (VAE) that divides frames into spatio-temporal patches with positional encodings and compresses them along the temporal dimension. These operations disrupt the spatial correlation of native control signals, e.g., for trajectory and depth, making them incompatible in their raw forms. Likewise, content information provided at different temporal locations must be aligned with DiT's representation space as well. Furthermore, ControlNet-style conditioning, assumes dense, pixel-aligned feature injection into UNet backbones, and is not directly applicable to the DiT architecture used in our method.

To resolve the incompatibility, we map all the controls into the same domain as the video/noise input. First, trajectory and depth information, presented in the form of optical flow and depth maps respectively, are converted into RGB, as the VAE in DiT operates on such a format. Specifically, depth is represented using relative color encoding and applied to both compositional layering (see Fig. 1, top two rows) and for object-specific control (see Fig. 4). Next, we transform dense optical flow and depth maps into *sparse*, point-based representations by extracting trajectories from high-motion regions. Along these trajectories, both optical flow and depth values are sampled, yielding sparse control points, which are more user-friendly. Users may provide one or both of these modalities, trajectory and depth, depending on the desired level of control. The input frames, including those defining target regions (see Fig. 1, bottom), are inserted at designated temporal positions with the remaining slots filled with black frames and corresponding binary masks indicating valid regions. These representations are passed through the DiT-VAE and appended to the DiT input noise.

The second challenge arises when we must generate intermediate frames coherently with the keyframes in both spatial and temporal domains, while respecting the variety of controls which operate at varying levels of granularity and influence. To this end, we separate content controls, given by the input frames, from motion controls, via optical flow and depth, into *two branches* to encode the required features before guiding the denoising process, resulting in two generators, one for motion and the other for content. Our experiments have demonstrated that such a dual-branch setup provides greater stability and robustness in training, while preserving both motion fidelity and content consistency, despite the multi-modality. Finally, we propose a stage-wise training strategy to ensure that our model learns the multi-modal controls smoothly. Specifically, we feed the model with denser and more concrete controls first, and then gradually move to sparser and higher-level controls.

We evaluate MultiCOIN through extensive quantitative and qualitative experiments. Our method supports multi-object control using trajectory and depth, with content guided by keyframes and target regions at different temporal points. Depth enables layering and object-specific control, while text prompts refine motion or act as standalone signals. By aligning motion controls with the input in the spatio-temporal domain, our approach achieves significantly better trajectory alignment compared to Framer. Moreover, MultiCOIN demonstrates strong multi-modal versatility, highlighting the benefits of complementing trajectories with additional controls for more flexible inbetweening.

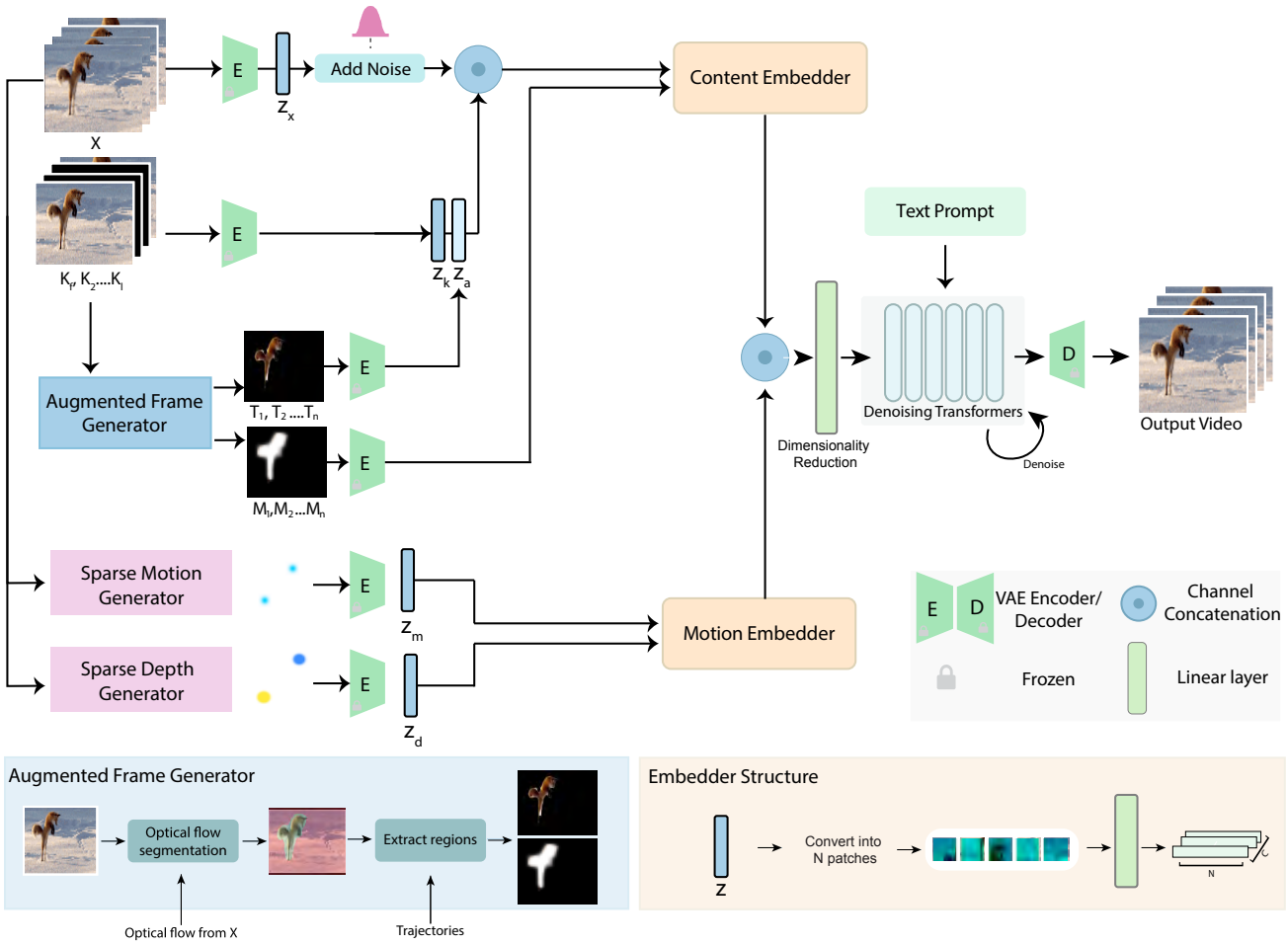


**Figure 2:** *Overview of MultiCOIN pipeline. Given a video X, we extract multi-modal motion controls through two generators: the Sparse Motion Generator via optical flow and the Sparse Depth Generator for depth maps, producing sparse RGB points for trajectory/depth. Augmented Frame Generator computes target regions and masks for fine-grained content control. Control signals are encoded via a dual-branch embedder architecture to separately capture motion and content. In addition, a text prompt condition is processed by a text encoder to provide semantic guidance over the generated content. At inference, the model flexibly integrates these multi-modal controls for interpolation.*

## 2. Related Work

### 2.1. Video Generation

Creating realistic and novel videos has long been an interesting research problem [YCS*23, RSB*16]. Earlier studies have employed various generative models including GANs [YCS*23, SMS17, TLYK17, SLE23] and temporally aware networks such as LSTM or autoregressive models [SMS16, YZAS21, HDZ*22]. Recently, inspired by the success of diffusion models in image synthesis, several works have begun to investigate the use of diffusion models for conditional and unconditional video generation [HSG*22, SPH*22, HCS*22, KOZ*24]. Stable Video Diffusion [BDK*23] leverages latent diffusion models [RBL*22, BRL*23] for generating temporally coherent content. Few-shot video generation is facilitated by methods like Tune-a-video [WGW*23], which finetunes [HSS*23] leverage large language models for generative guidance. Another approach to generating videos in a controllable manner is to use keyframes along with text conditions [GSB*23, WBW*24, LCW*23, ZWZ*24], where initial frames are generated to guide subsequent frames, with latent-consistency networks ensuring temporal and visual coherence. VideoJAM [CSZ*25] introduces joint appearance–motion representations to improve motion coherence in diffusion video models. In this work, we target video inbetweening that interpolates between two frames using flexible multi-modal controls in a joint framework.

### 2.2. Conditional Image-to-Video Generation

Conditional image-to-video (I2V) methods generate videos by extrapolating motion from a single input image, typically guided by text prompts or additional control signals. Several recent works ex-

plore explicit motion control in this setting, including trajectory-guided generation such as Tora [ZLL*24], TrackGo [ZWN*25], Image Conductor [LWZ*25] and interaction-based guidance in Motion Prompting [GHH*25]. Other approaches incorporate dense structural conditioning, for example full-body pose sequences or sketches, as in UniAnimate [WZG*25] and Easy-Control [ZYS*25]. Frame Guidance [JKJ*25] further studies conditional video generation using dense inputs such as depth maps, sketches, or color blocks. While these methods demonstrate strong controllability for video generation, in contrast, video inbetweening is conditioned on both the first and final frames, requiring the generated sequence to match the target appearance and geometry at the end timestep. This imposes a stronger structural constraint than I2V generation and changes how control signals such as trajectories, depth, and target regions interact with the model.

### 2.3. Video Inbetweening

Video inbetweening also known as frame interpolation, frame rate up-conversion, or temporal super-resolution, has a long history, with early approaches operating at a block- instead of a pixel-level due to compute constraints at the time [CLK00, HLK04b]. While we have more compute nowadays, the underlying framework of motion estimation and compensation has largely remained the same throughout the years [NL20, NHC23, RKT*22, NL18]. Flow-based methods use optical flow from the input frames for generating the inbetween frames [JSJ*18, PKLK20, HZH*20]. While approaches like phase- or kernel-based interpolation [NML17a, NML17b,MWZ*15,ZYL*22] use spatially adaptive kernels to synthesize the interpolated pixels. In either case, it is fundamentally still about re-synthesizing an in-between frame from what is in the input frames. However, as the inputs become more distant in time and/or space, the inbetweening will require information that is not present so we need to hallucinate it instead. Nowadays we can utilize foundational video models for generating plausible interpolation results [FDX*24, DZB24, XXZ*25, XLX*24, BDO*25], but users typically aren't interested in just a possible interpolation result but one that follows their artistic expression. This is where motion control comes into the picture, which is the focus of our work. Framer [WWZ*24] is a work that achieves impressive results in controllable inbetweening using motion trajectories. Our method aligns trajectory control more effectively with the input in spatio-temporal domain, resulting in improved motion accuracy. In addition, it introduces a multi-modal framework that combines complementary controls to generate more diverse outputs. Qualitative comparisons in Fig. 8 highlight these improvements.

## 3. Method

Our goal with MultiCOIN is to provide an intuitive and effective control mechanism for inbetweening tasks using motion (e.g. , trajectories, depth) and content (e.g. frames, target regions) guidance for intuitive and fine-grained control as shown in Fig. 1 for generating realistic and coherent outputs. During the training, given the ground truth video clip $X$ with the extracted keyframes $\{K_f, K_2, \ldots, K_l\}$, we represent the motion control as *depth-trajectories* consisting of sparse points $\{P_1, P_2, \ldots, P_m\}$ which have both directional and depth information. We use optical flow and depth maps visualized in RGB format, from which we then extract $\{P_1, P_2, \ldots, P_m\}$ through the proposed *Sparse Motion/Depth Generators*. Along with keyframes we add additional content control via target regions $\{T_1, T_2, \ldots, T_n\}$ and associated guide masks $\{M_1, M_2, \ldots, M_n\}$. These provide regional control in content generation. We extract them through the proposed *Augmented Frame Generator*. Our overall model builds on a DiT-based video diffusion backbone, chosen for its ability to generate long and coherent videos. On top of this backbone, we thus propose two modules (1) Sparse Motion/Depth Generators, which produce the trajectory-based motion controls from RGB flow and depth maps, and (2) Augmented Frame Generator, which provides regional content controls through target regions and masks. These complementary modules are integrated through dual-branch embedders that separately encode motion and content controls. The following sections describe the DiT backbone and each proposed module.

### 3.1. Preliminary

Models like Stable Video Diffusion (SVD) are generative models that extend image diffusion to video by maintaining temporal consistency across frames. Given a noisy video $X_T$, the model utilizes a conditional 3D-UNet to progressively denoise it to a clean video $X_0$ by iteratively applying a denoising function: $X_{t-1} = \epsilon_\theta(X_t, t, c)$, where $\epsilon_\theta$ represents the learned noise, and $c$ represents conditions. Diffusion Transformer (DiT) [PX23] models combine diffusion-based denoising processes with transformer architectures. Compared to traditional UNet-based models, DiT leverages a transformer backbone as its core denoiser to model long-range dependencies and global context, which is critical for capturing fine details and significantly improves the versatility and quality of image and video generation. For training, a diffusion loss is used which measures the mean square error (MSE) between the predicted noise $\hat{\epsilon}$ and the input noise $\epsilon$: $\mathcal{L}_{diff} = ||\hat{\epsilon} - \epsilon||_2^2$.

### 3.2. Control Generation

Large-motion interpolation is challenging due to ambiguity, artifacts, and distortions, requiring precise control. Our method employs two mechanisms: 1) Sparse Motion-Depth Generator, which focuses on key motion paths, and 2) Augmented Frame Generator, which adds extra visual context. Together, these methods enhance the model's ability to produce controlled, natural-looking motion.

#### 3.2.1. Sparse Motion-Depth Generators

The Sparse Motion-Depth Generator (Fig. 3) produces motion outputs aligned with both the model architecture and the input video $X$. A key challenge is generating motion inputs that are not only valid for our target scenario but also structurally compatible with the DiT framework. DiT uses a ViT-based 3D VAE to encode input videos by dividing frames into patches with learned positional encodings and compresses them along temporal dimension. Because our motion inputs are physically grounded (e.g. optical flow and depth maps that directly control spatial displacements), ensuring their compatibility with this patch-based and temporally compressed representation is non-trivial.

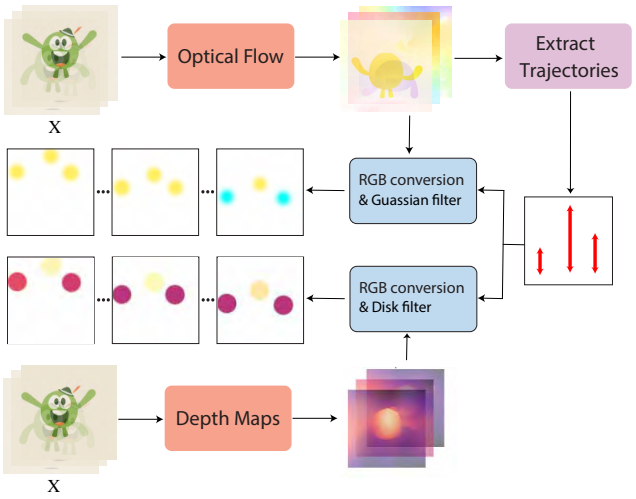


**Figure 3:** *Sparse Motion and Depth Generator. Given video X, dense optical flow and depth maps are computed. Trajectories are selected from high-motion regions along which flow/depth points are sampled and expanded with 2D filters to get sparse RGB inputs.*

First, in the absence of ground-truth motion trajectories, we generate them by extracting dense optical flow from the input video $X$ and tracking points with high motion magnitude using a coarse grid. The coarse grid prevents oversampling a single area, while motion magnitude highlights regions with strong trajectories. Along these tracked paths, we must extract both optical flow and depth information. To make these motion inputs compatible with DiT, we project all controls into the model's latent space. We begin by converting both optical flow and depth trajectories into RGB representations. For optical flow, this involves mapping the direction and magnitude of motion to color space creating a visual representation. For depth, relative distance values are mapped to a red–blue colormap, where hue encodes whether points lie inward or outward in the scene. These visualizations allow optical flow and depth to be processed like RGB video frames.

We extract 2-25 sparse trajectories, as discussed above, and along these trajectories extract the optical flow and depth RGB values. Since each trajectory originates from a single pixel, the resulting motion signals are too sparse to be meaningful. To improve spatial coverage and interpretability, we expand each trajectory's influence using 2D filters. For optical flow, we adopt a Gaussian filter, similar to [WYW*24], which spreads motion smoothly while preserving directionality and gradually reducing magnitude. For depth, we instead use a disk filter that copies depth values over a circular region. Gaussian spreading is suited for optical flow, where hue encodes direction and gradual falloff naturally models motion attenuation, whereas depth hue directly encodes distance, so even small color changes correspond to different depth values. A uniform disk is therefore preferable to avoid introducing unintended depths. The sparse trajectories are thus converted into sparse RGB point controls $P_1, P_2, \ldots, P_n \in \mathbb{R}^{H \times W \times 3}$, following the same format as the input frames. These controls are then passed through DiT's existing 3D VAE, allowing the model to effectively embed motion information, yielding promising results.

Another challenge with depth being a relative measure arises during inference, specifically for single-point inputs, where a depth reference is critical to anchor the model's understanding. To address this, we compute the mean of the sparse depth values provided by the user along the trajectory and generate anchor points at three multiples above and below the mean. These anchors are placed at the corners of the depth input to supply global depth context, as illustrated in Fig. 4.

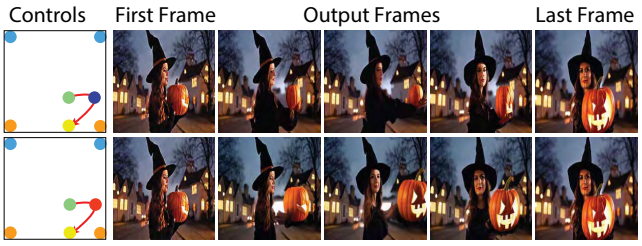


**Figure 4:** *Example of a witch moving the Jack-o'-Lantern along the same path, with motion inward (top) or outward (bottom), depending on midpoint depth (blue vs. red dot).*

### 3.2.2. Augmented Frame Generator

While motion paths provide effective control over inbetweening, we discovered that the inherent ambiguity of diffusion models, combined with the challenges of interpolating large motions, makes regional control an important enhancement. This approach refines the output, reducing the number of motion paths needed. At the same time, we want to avoid overly rigid control, allowing for more natural results. To achieve this, we introduce Augmented Frames. The core concept is to provide the model with a subtle "nudge" in the right direction using content which we call "Target Regions". This may be used alongside trajectories or on their own.

To implement this, we extract a region of interest from $K_f$ and translate it across several frames according to the corresponding trajectory to create frames of target regions $\{T_1, T_2, \ldots, T_n\}$, which are appended to $K_f$ temporally. For training, we generate regions from motion trajectories using optical-flow segmentation. Further details are available in Fig. 2. The "fox" example in the figure illustrates how we extract the region corresponding to the direction of sparse motion trajectories and append it to the input keyframes.

We also extract a binary mask that associates valid and invalid regions, and goes as an extra condition into content encoder. This helps the model to separate valid pixel information (in keyframes and target regions) from invalid information. Once the model learns to interpret target regions, we can manually set these guiding regions. Users can specify exactly where the model should place content, such as moving a region from one spot to another. This allows explicit control over the generated frames. The target region control reduces the need for users to draw extensive trajectories, helping the model accurately identify and track the complete moving object with minimal input. More details and results can be found

in the experiment section. During training, we dropout this content condition with probability of 50%, making it optional at inference.

### 3.3. Stage-wise Training with Dual-Branch Encoders

To train our model, we utilize a dual-branch encoder structure. First, a set of random keyframes $\{K_f, K_2, \ldots, K_l\}$ is extracted from $X$. First and last keyframe are always provided, and we select 0-5 random keyframes in between first and last keyframe to help the model learn multiple keyframe interpolation. We extract $\{T_1, T_2, \ldots, T_n\}$ from these keyframes, for which we use the dense optical flow of $X$ to create sparse trajectories and optical flow segmentations. The first branch encodes the content information including $\{K_f, K_2, \ldots, K_l\}$, $\{T_1, T_2, \ldots, T_n\}$ and $\{M_1, M_2, \ldots, M_n\}$.

For motion, we extract $\{P_1, P_2, \ldots, P_m\}$ which includes both motion and depth as discussed above and details in Fig 3. The second branch encodes this motion information. Both branches have a similar structure. The input (motion or content) is first passed through a frozen VAE to encode it into a latent representation. For content, the latent representation of noise is channel-concatenated with the latent output of conditional images (keyframes and target regions). These latent outputs are then passed through Embedders, which first transform the inputs into patches and then funnels the output through a linear layer. These outputs are again channel-concatenated and passed through a final linear layer before being fed into the transformer denoiser.

To train our model, we utilize a stage-wise training strategy, where we gradually introduce conditional inputs to the model. First, the model is trained on the image branch alone to learn core video interpolation, ensuring it can interpolate between two images without conditions. Afterwards, to embed the motion and depth as a condition, we first performed a trial experiment. Using the architecture in Fig. 2 we directly train with $\{P_1, P_2, \ldots, P_m\}$. From the results we saw that the model struggles to properly follow the motion and depth, specifically in localizing the movement. This analysis is discussed in Sec. 4.4. To address this issue, we adopted an alternative approach inspired by [YWL*23, WYW*24].

We first trained the model solely with dense optical flow and dense depth maps, and then gradually introduced the sparse motion inputs. This phased approach enables the model to better interpret the limited motion information. In the last step, we train with guided pixels ($\{T_1, T_2, \ldots, T_n\}$) and ($\{M_1, M_2, \ldots, M_n\}$). Intuitively, we opted for a two-branch system to separate the two very different conditional inputs. In Sec. 4.4 we show how this approach leads to better stability in the output.

## 4. Experiments

To evaluate MultiCOIN's performance, we conduct both quantitative and qualitative assessments across a range of video sequences and datasets. Currently, Framer [WWZ*24] is the only baseline that supports controllable interpolation, but it relies solely on trajectory control. Therefore, we compare our method to Framer under the trajectory control setting. For the quantitative evaluation, we assess both the generative quality and motion control of our model.

**Implementation Details**: We apply our method to a pretrained DiT text-to-video diffusion model, similar in architecture to Open-Sora. The model is trained on the latent space of a 3D VAE that encodes 32 video frames into 5 latent frames. Training videos consist of 64 frames at resolution of 352x640, paired with text captions. The training uses 16 Nvidia A100 GPUs. We use an Adam optimizer with $1 \times 10^{-4}$ learning rate. Approximately 5k steps are used to train the image-to-video model, 2k steps for optical flow training, 2k for sparse and, 2k for target region input. The entire model, except the VAE and text encoders, is finetuned end-to-end.

*Automatic Trajectory Generation*: For a fair quantitative comparison with baseline, we employ an automatic trajectory generation method similar to Framer [WWZ*24]. Specifically, SIFT is used to extract features from the first and last frames of the video, and then point pairs are selected with high correspondence. A linear trajectory is generated between these matched points.

**Metrics and Datasets**: Following [FDX*24, CWZ*23, WWZ*24] we use SSIM, FVD [UvSK*19], and LPIPS [ZIE*18] for quality comparison. Additionally, we introduce a "Motion" metric to evaluate trajectory control. This extracts trajectory paths from the generated output using optical flow and compares them to the input trajectory via Fréchet Distance, which captures entire path geometry, unlike MSE that measures pointwise error and is sensitive to small misalignments. We use DAVIS [PTPC*17] and UCF (Sports Action) [RAS08] for analysis, as both feature large frame-to-frame motion across diverse cases.

### 4.1. Qualitative Results

As shown in Fig. 5, our model integrates both content and motion controls, including trajectory, depth, target regions, and text. Trajectory produces smooth, realistic motion along the given path, while depth handles both relative cases (e.g. a cat moving around a pumpkin) and single-point inputs (e.g. a balloon). Combining trajectory and depth enables simultaneous 2D translation and depth variation. Fig. 9 shows different outputs based on different control signals. Text further refines outputs, and target regions provide intuitive content editing. The model also supports more than two input frames, as illustrated in Fig. 7, with/without motion control. Morphing is another application, for which results are shown in Fig. 6.

### 4.2. Qualitative Evaluation

We provide a qualitative comparison with the baseline Framer [WWZ*24] in Fig. 8. We also attach videos for comparison in supplementary. Our model achieves smoother transitions with fewer distortions and artifacts, producing more natural interpolations. In Framer, motion is introduced as an external condition via ControlNet that interacts with video features indirectly. In contrast, our method embeds motion into the same latent space as the video, enabling stronger spatio-temporal alignment. This integration preserves frame quality while seamlessly incorporating user-defined motion and demonstrating the versatility of multi-modal control. In the second example, depth is leveraged to create compelling effects, while in the third, text serves as an additional condition to guide the model when trajectory alone is insufficient.

### 4.3. Quantitative Evaluation

Quantitative results are reported in Tab. 1 on the DAVIS [PTPC*17] and UCF (Sports) [AZA*21] datasets. The motion metric demon-

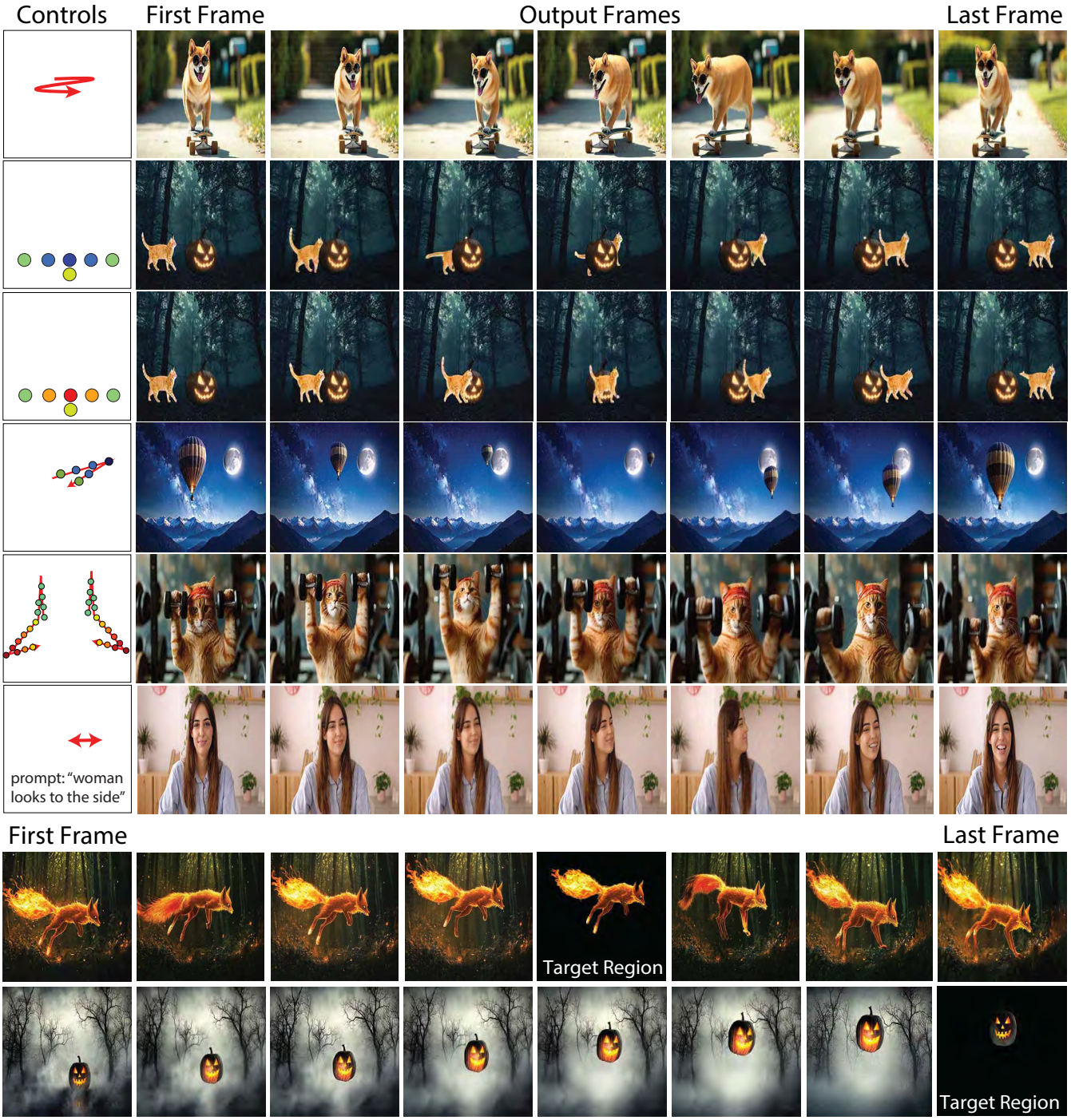


**Figure 5:** *Our results illustrate several ways multi-modal controls can be applied to frame interpolation. In the top section, we show trajectory control on its own, followed by two depth variations that place the cat either in front of or behind the pumpkin. Combining trajectory with depth produces richer motion: the balloon recedes along the z-axis while the weights with the cat are pushed outward. Prompts can also be paired with trajectories, where the trajectory sets the overall movement and the prompt refines details. In the bottom section, we highlight target region control. The temporal placement of target regions determines content editing at that point: in the first case, they are inserted in the middle with both first and last frames given, while in the second they appear at the end serving as a soft replacement for the last frame.*

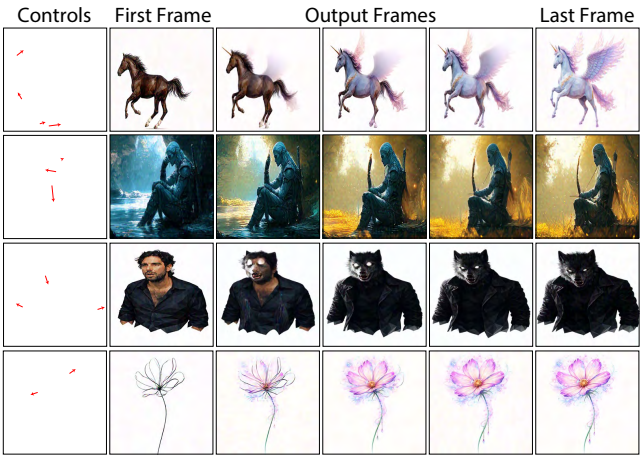


**Figure 6:** *Results showcasing morphing.*

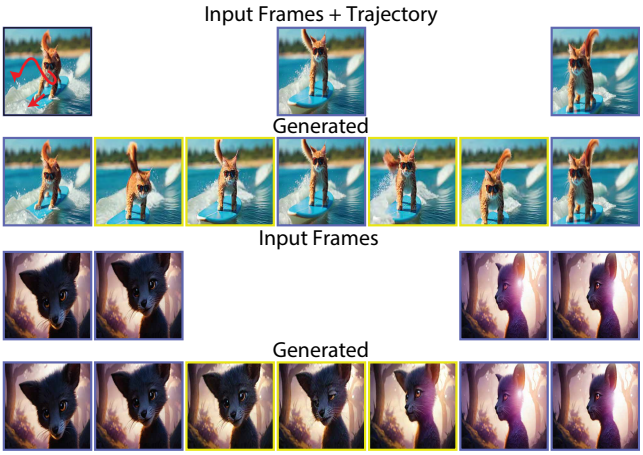


**Figure 7:** *Results with multiple input frames, with and w/o motion.*

strates a clear improvement over the baseline in capturing motion trajectories. In terms of visual quality, our model generally matches or surpasses the baseline, with DAVIS-SSIM being the only exception, where we observe a minor decrease. Since SSIM is highly sensitive to pixel-level alignment, the slight drop in this metric is not critical. More importantly, the improved FVD indicates that our method produces perceptually more realistic and temporally consistent videos. Overall, our approach delivers comparable or superior visual quality with substantially stronger motion control.

**User Study:** In addition to quantitative evaluation, we conducted a user study to assess perceptual realism and faithfulness to the specified control signals. The study includes a total of 40 examples with custom motion controls, covering trajectory control and morphing. To ensure fair comparison, half of the examples are drawn from Framer's representative cases. A total of 50 people participated in the study. The results, summarized in Fig. 11, corroborate our findings, showing a clear preference for our method.

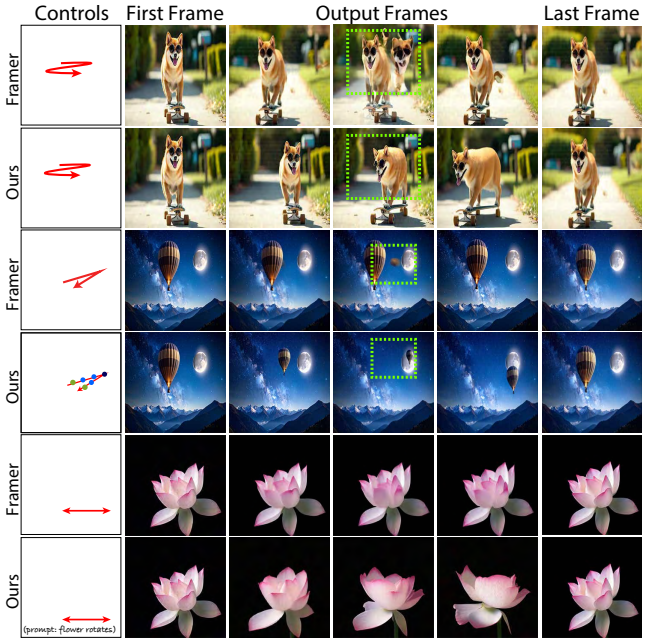


**Figure 8:** *Comparison with Framer [WWZ*24]. Top row shows reduced trajectory distortion. Bottom rows demonstrate the benefits of additional controls, including depth and text guidance.*

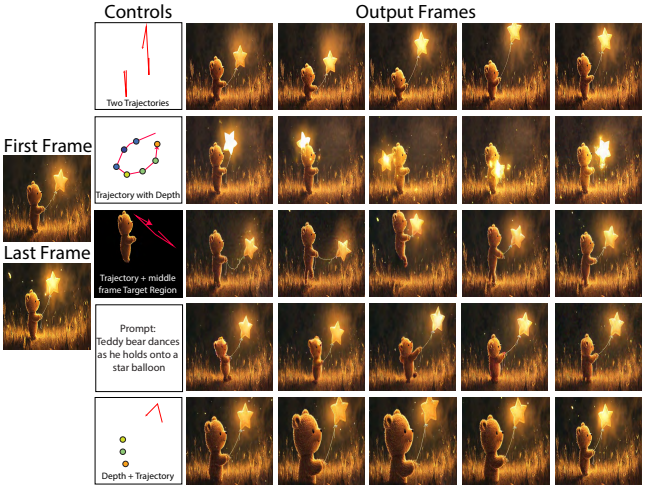


**Figure 9:** *Identical start and end frames with different controls.*

### 4.4. Ablation Study

**The Effectiveness of Stage-wise Training.** We initially experimented with training directly on sparse motion and depth inputs. While this approach produced outputs with comparable perceptual quality, evidenced by FVD and LPIPS scores that remain on par with stage-wise training, the model failed to integrate the motion and depth cues effectively. As illustrated in Fig. 10, motion fails to

| Model | DAVIS | | | | UCF (Sports) | | | |
|---|---|---|---|---|---|---|---|---|
| | FVD↓ | LPIPS↓ | SSIM↑ | Motion↓ | FVD↓ | LPIPS↓ | SSIM↑ | Motion↓ |
| Framer | 4.42 | **0.50** | **0.18** | 5.25 | **2.15** | 0.48 | 0.21 | 3.31 |
| Ours | **4.33** | **0.50** | 0.16 | **2.44** | 2.14 | **0.31** | **0.34** | **2.34** |

**Table 1:** *Quantitative comparison with Framer [WWZ*24]*

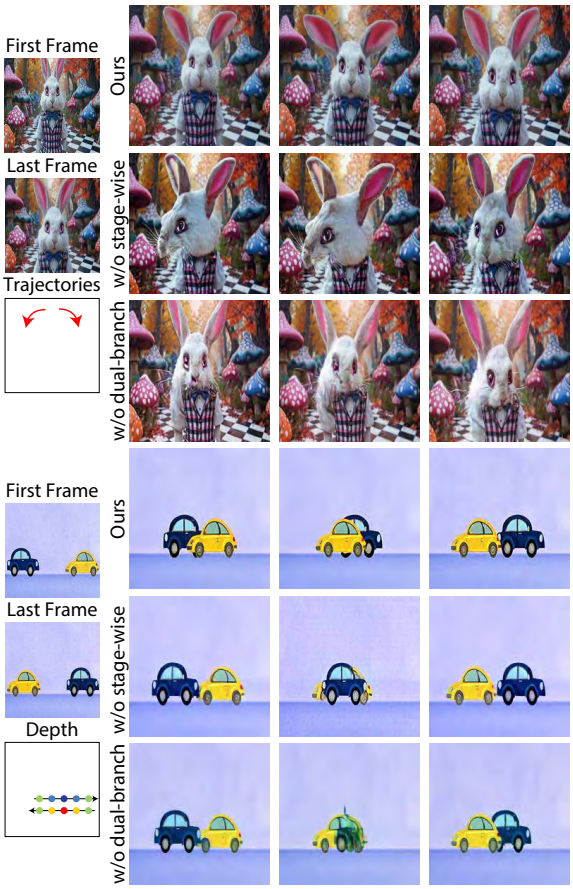


**Figure 10:** *Ablation: w/o stage-wise: Skipping stage-wise training degrades performance; without dense flow t he m odel mislocalizes motion, and without dense depth it misinterprets depth control. w/o dual-branch: Removing the dual-branch design increases artifacts and causes depth confusion.*

localize accurately and depth information is misinterpreted. Quantitative results in Tab. 2 further confirm that motion control deteriorates severely without stage-wise training. Thus, even though the overall fidelity of generated frames is preserved, since the model was still trained on two-image interpolation and can hallucinate visually plausible outputs, the absence of stage-wise training leads to poor motion adherence, with the model failing to accurately follow the provided cues.

**The Effectiveness of Dual-branch Encoders.** In our system, content and motion are encoded through two dedicated branches. In this ablation, we replace the dual-branch design with a single branch, where all conditions are concatenated with the input noise.

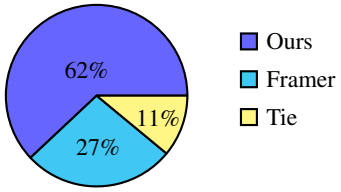


**Figure 11:** *User Study results.*

| | DAVIS | | | | UCF (Sports) | | | |
|---|---|---|---|---|---|---|---|---|
| | FVD ↓ | LPIPS ↓ | SSIM ↑ | Mot. ↓ | FVD ↓ | LPIPS ↓ | SSIM ↑ | Mot. ↓ |
| w/o stage-wise | **4.25** | **0.49** | 0.14 | 4.41 | 2.32 | **0.26** | **0.38** | 4.81 |
| w/o dual-branch | 5.90 | 0.53 | 0.13 | 3.24 | 3.28 | 0.32 | 0.33 | 4.93 |
| **Ours** | 4.33 | 0.50 | **0.18** | **2.44** | **2.14** | 0.31 | 0.34 | **2.34** |

**Table 2:** *Ablation study using w/o stage-wise and w/o dual-branch.*

As shown in Fig. 10, this design leads to noticeably more artifacts when content and motion are not disentangled. The quantitative results in Tab. 2 further highlight a substantial decline in motion control, along with reduced realism as indicated by the higher FVD.

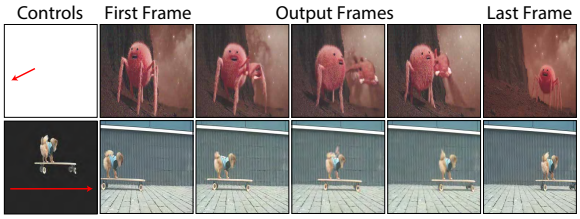


**Figure 12:** *Failure cases.* ***Top:*** *Trajectory conflicts with end frame.* ***Bottom:*** *Target region is overridden by trajectory.*

## 5. Limitations

Our method focuses on object-level motion control for video inbetweening and does not explicitly model camera motion. While text prompts can influence camera behavior, this effect is inconsistent and outside our intended scope. Because multiple conditioning signals are integrated into a unified latent space, conflicting cues (e.g., motion, depth, region, and text) may degrade results as shown in Fig. 12. The proposed target-region controls act as weak guidance rather than strict constraints, which can limit precision. Stronger integration and better adaptation of user-provided controls into a more tightly coupled space are left for future work.

## 6. Conclusion

We introduced MultiCOIN, a DiT-based framework for controllable inbetweening that generates high-quality interpolated frames conditioned on trajectories, depth, target regions, and text prompt. These conditions may be used individually or in combination with each other. Extensive experiments both qualitative and quantitative, demonstrate its versatility and effectiveness across a wide variety of use-cases. Nonetheless, challenges remain, particularly in aligning trajectories with image content, as strong content conditioning can dominate and suppress motion cues. Future work may incorporate lightweight pre-processing modules to better balance such controls, thereby preserving user intent while maintaining quality.

## References


[AZA*21] AHMADYAN A., ZHANG L., ABLAVATSKI A., WEI J., GRUNDMANN M.: Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2021), pp. 7822–7831.

[BDK*23] BLATTMANN A., DOCKHORN T., KULAL S., MENDELEVITCH D., KILIAN M., LORENZ D., LEVI Y., ENGLISH Z., VOLETI V., LETTS A., ET AL.: Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127* (2023).

[BDO*25] BRIEDIS K. M., DJELOUAH A., ORTIZ R., GROSS M., SCHROERS C.: Controllable tracking-based video frame interpolation. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers* (2025), pp. 1–11.

[BRL*23] BLATTMANN A., ROMBACH R., LING H., DOCKHORN T., KIM S. W., FIDLER S., KREIS K.: Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 22563–22575.

[CLK00] CHOI B., LEE S., KO S.: New frame rate up-conversion using bi-directional motion estimation. *IEEE Trans. Consumer Electron. 46*, 3 (2000), 603–609.

[CSZ*25] CHEFER H., SINGER U., ZOHAR A., KIRSTAIN Y., POLYAK A., TAIGMAN Y., WOLF L., SHEYNIN S.: Videojam: Joint appearance-motion representations for enhanced motion generation in video models. *arXiv preprint arXiv:2502.02492* (2025).

[CWZ*23] CHEN X., WANG Y., ZHANG L., ZHUANG S., MA X., YU J., WANG Y., LIN D., QIAO Y., LIU Z.: Seine: Short-to-long video diffusion model for generative transition and prediction. In *The Twelfth International Conference on Learning Representations* (2023).

[DZB24] DANIER D., ZHANG F., BULL D.: Ldmvfi: Video frame interpolation with latent diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2024), vol. 38, pp. 1472–1480.

[FDX*24] FENG H., DING Z., XIA Z., NIKLAUS S., ABREVAYA V., BLACK M. J., ZHANG X.: Explorative inbetweening of time and space. *arXiv preprint arXiv:2403.14611* (2024).

[GHH*25] GENG D., HERRMANN C., HUR J., COLE F., ZHANG S., PFAFF T., LOPEZ-GUEVARA T., AYTAR Y., RUBINSTEIN M., SUN C., ET AL.: Motion prompting: Controlling video generation with motion trajectories. In *Proceedings of the Computer Vision and Pattern Recognition Conference* (2025), pp. 1–12.

[GSB*23] GIRDHAR R., SINGH M., BROWN A., DUVAL Q., AZADI S., RAMBHATLA S. S., SHAH A., YIN X., PARIKH D., MISRA I.: Emu video: Factorizing text-to-video generation by explicit image conditioning. *arXiv preprint arXiv:2311.10709* (2023).

[HCS*22] HO J., CHAN W., SAHARIA C., WHANG J., GAO R., GRITSENKO A., KINGMA D. P., POOLE B., NOROUZI M., FLEET D. J., ET AL.: Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303* (2022).

[HDZ*22] HONG W., DING M., ZHENG W., LIU X., TANG J.: Cogvideo: Large-scale pretraining for text-to-video generation via transformers, 2022. URL: https://arxiv.org/abs/2205.15868, arXiv:2205.15868.

[HLK04a] HA T., LEE S., KIM J.: Motion compensated frame interpolation by new block-based motion estimation algorithm. *IEEE Transactions on Consumer Electronics 50*, 2 (2004), 752–759.

[HLK04b] HA T., LEE S., KIM J.: Motion compensated frame interpolation by new block-based motion estimation algorithm. *IEEE Trans. Consumer Electron. 50*, 2 (2004), 752–759.

[HSG*22] HO J., SALIMANS T., GRITSENKO A., CHAN W., NOROUZI M., FLEET D. J.: Video diffusion models. *Advances in Neural Information Processing Systems 35* (2022), 8633–8646.

[HSS*23] HONG S., SEO J., SHIN H., HONG S., KIM S.: Large language models are frame-level directors for zero-shot text-to-video generation. In *First Workshop on Controllable Video Generation@ ICML24* (2023).

[HZH*20] HUANG Z., ZHANG T., HENG W., SHI B., ZHOU S.: Rife: real-time intermediate flow estimation for video frame interpolation. arxiv preprint arxiv. 2011: 06294. In *Rife: Real-time intermediate flow estimation for video frame interpolation. arXiv preprint arXiv: 2011.06294*. 2020.

[JKJ*25] JANG S., KI T., JO J., YOON J., KIM S. Y., LIN Z., HWANG S. J.: Frame guidance: Training-free guidance for frame-level control in video diffusion models. *arXiv preprint arXiv:2506.07177* (2025).

[JSJ*18] JIANG H., SUN D., JAMPANI V., YANG M.-H., LEARNED-MILLER E., KAUTZ J.: Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 9000–9008.

[KOZ*24] KWON M., OH S. W., ZHOU Y., LIU D., LEE J.-Y., CAI H., LIU B., LIU F., UH Y.: Harivo: Harnessing text-to-image models for video generation. *arXiv preprint arXiv:2410.07763* (2024).

[LCW*23] LI X., CHU W., WU Y., YUAN W., LIU F., ZHANG Q., LI F., FENG H., DING E., WANG J.: Videogen: A reference-guided latent diffusion approach for high definition text-to-video generation. *arXiv preprint arXiv:2309.00398* (2023).

[LWZ*25] LI Y., WANG X., ZHANG Z., WANG Z., YUAN Z., XIE L., SHAN Y., ZOU Y.: Image conductor: Precision control for interactive video synthesis. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2025), vol. 39, pp. 5031–5038.

[MCD*18] MEYER S., CORNILLÈRE V., DJELOUAH A., SCHROERS C., GROSS M. H.: Deep video color propagation. In *BMVC* (2018), BMVA Press, p. 128.

[MWZ*15] MEYER S., WANG O., ZIMMER H., GROSSE M., SORKINE-HORNUNG A.: Phase-based frame interpolation for video. In *CVPR* (2015), IEEE Computer Society, pp. 1410–1418.

[NHC23] NIKLAUS S., HU P., CHEN J.: Splatting-based synthesis for video frame interpolation. In *WACV* (2023), IEEE, pp. 713–723.

[NL18] NIKLAUS S., LIU F.: Context-aware synthesis for video frame interpolation. In *CVPR* (2018), Computer Vision Foundation / IEEE Computer Society, pp. 1701–1710.

[NL20] NIKLAUS S., LIU F.: Softmax splatting for video frame interpolation. In *CVPR* (2020), Computer Vision Foundation / IEEE, pp. 5436–5445.

[NML17a] NIKLAUS S., MAI L., LIU F.: Video frame interpolation via adaptive convolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 670–679.

[NML17b] NIKLAUS S., MAI L., LIU F.: Video frame interpolation via adaptive separable convolution. In *ICCV* (2017), IEEE Computer Society, pp. 261–270.

[PKLK20] PARK J., KO K., LEE C., KIM C.-S.: Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16* (2020), Springer, pp. 109–125.

[PTPC*17] PONT-TUSET J., PERAZZI F., CAELLES S., ARBELÁEZ P., SORKINE-HORNUNG A., VAN GOOL L.: The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675* (2017).

[PX23] PEEBLES W., XIE S.: Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 4195–4205.

[RAS08] RODRIGUEZ M. D., AHMED J., SHAH M.: Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *2008 IEEE conference on computer vision and pattern recognition* (2008), IEEE, pp. 1–8.

[RBL*22] ROMBACH R., BLATTMANN A., LORENZ D., ESSER P., OMMER B.: High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022), pp. 10684–10695.

[RKT*22] REDA F., KONTKANEN J., TABELLION E., SUN D., PANTOFARU C., CURLESS B.: Film: Frame interpolation for large motion. In *European Conference on Computer Vision* (2022), Springer, pp. 250–266.

[RSB*16] RANZATO M., SZLAM A., BRUNA J., MATHIEU M., COLLOBERT R., CHOPRA S.: Video (language) modeling: a baseline for generative models of natural videos, 2016. URL: https://arxiv.org/abs/1412.6604, arXiv:1412.6604.

[SLE23] SHEN X., LI X., ELHOSEINY M.: Mostgan-v: Video generation with temporal motion styles, 2023. URL: https://arxiv.org/abs/2304.02777, arXiv:2304.02777.

[SMS16] SRIVASTAVA N., MANSIMOV E., SALAKHUTDINOV R.: Unsupervised learning of video representations using lstms, 2016. URL: https://arxiv.org/abs/1502.04681, arXiv:1502.04681.

[SMS17] SAITO M., MATSUMOTO E., SAITO S.: Temporal generative adversarial nets with singular value clipping, 2017. URL: https://arxiv.org/abs/1611.06624, arXiv:1611.06624.

[SPH*22] SINGER U., POLYAK A., HAYES T., YIN X., AN J., ZHANG S., HU Q., YANG H., ASHUAL O., GAFNI O., ET AL.: Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792* (2022).

[SZY*21] SIYAO L., ZHAO S., YU W., SUN W., METAXAS D. N., LOY C. C., LIU Z.: Deep animation video interpolation in the wild. In *CVPR* (2021), Computer Vision Foundation / IEEE, pp. 6587–6595.

[TLYK17] TULYAKOV S., LIU M.-Y., YANG X., KAUTZ J.: Mocogan: Decomposing motion and content for video generation, 2017. URL: https://arxiv.org/abs/1707.04993, arXiv:1707.04993.

[UvSK*19] UNTERTHINER T., VAN STEENKISTE S., KURACH K., MARINIER R., MICHALSKI M., GELLY S.: Fvd: A new metric for video generation. In *ICLR 2019 Workshop DeepGenStruct* (2019).

[WBW*24] WANG Y., BAO J., WENG W., FENG R., YIN D., YANG T., ZHANG J., DAI Q., ZHAO Z., WANG C., ET AL.: Microcinema: A divide-and-conquer approach for text-to-video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 8414–8424.

[WGW*23] WU J. Z., GE Y., WANG X., LEI S. W., GU Y., SHI Y., HSU W., SHAN Y., QIE X., SHOU M. Z.: Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 7623–7633.

[WWZ*24] WANG W., WANG Q., ZHENG K., OUYANG H., CHEN Z., GONG B., CHEN H., SHEN Y., SHEN C.: Framer: Interactive frame interpolation. *arXiv preprint arXiv:2410.18978* (2024).

[WYW*24] WANG Z., YUAN Z., WANG X., LI Y., CHEN T., XIA M., LUO P., SHAN Y.: Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers* (2024), pp. 1–11.

[WZG*25] WANG X., ZHANG S., GAO C., WANG J., ZHOU X., ZHANG Y., YAN L., SANG N.: Unianimate: Taming unified video diffusion models for consistent human image animation. *Science China Information Sciences 68*, 10 (2025), 1–14.

[XLX*24] XING J., LIU H., XIA M., ZHANG Y., WANG X., SHAN Y., WONG T.-T.: Tooncrafter: Generative cartoon interpolation. *arXiv preprint arXiv:2405.17933* (2024).

[XXZ*25] XING J., XIA M., ZHANG Y., CHEN H., YU W., LIU H., LIU G., WANG X., SHAN Y., WONG T.-T.: Dynamicrafter: Animating open-domain images with video diffusion priors. In *European Conference on Computer Vision* (2025), Springer, pp. 399–417.

[YCS*23] YU L., CHENG Y., SOHN K., LEZAMA J., ZHANG H., CHANG H., HAUPTMANN A. G., YANG M.-H., HAO Y., ESSA I., JIANG L.: Magvit: Masked generative video transformer, 2023. URL: https://arxiv.org/abs/2212.05199, arXiv:2212.05199.

[YWL*23] YIN S., WU C., LIANG J., SHI J., LI H., MING G., DUAN N.: Dragnuwa: Fine-grained control in video generation by integrating text, image, and trajectory. *arXiv preprint arXiv:2308.08089* (2023).

[YZAS21] YAN W., ZHANG Y., ABBEEL P., SRINIVAS A.: Videogpt: Video generation using vq-vae and transformers, 2021. URL: https://arxiv.org/abs/2104.10157, arXiv:2104.10157.

[ZIE*18] ZHANG R., ISOLA P., EFROS A. A., SHECHTMAN E., WANG O.: The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 586–595.

[ZLL*24] ZHANG Z., LIAO J., LI M., QIN L., WANG W.: Tora: Trajectory-oriented diffusion transformer for video generation. *arXiv preprint arXiv:2407.21705* (2024).

[ZWN*25] ZHOU H., WANG C., NIE R., LIU J., YU D., YU Q., WANG C.: Trackgo: A flexible and efficient method for controllable video generation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2025), vol. 39, pp. 10743–10751.

[ZWZ*24] ZENG Y., WEI G., ZHENG J., ZOU J., WEI Y., ZHANG Y., LI H.: Make pixels dance: High-dynamic video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 8850–8860.

[ZYL*22] ZHOU Y., YANG J., LI D., SAITO J., ANEJA D., KALOGERAKIS E.: Audio-driven neural gesture reenactment with video motion graphs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022), pp. 3418–3428.

[ZYS*25] ZHANG Y., YUAN Y., SONG Y., WANG H., LIU J.: Easycontrol: Adding efficient and flexible control for diffusion transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2025), pp. 19513–19524.