MotionBridge: Dynamic Video Inbetweening with Flexible Controls

Maham Tanveer[1,2*] Yang Zhou[2] Simon Niklaus[2]
Ali Mahdavi Amiri[1] Hao Zhang[1] Krishna Kumar Singh[2] Nanxuan Zhao[2]
[1]Simon Fraser University [2]Adobe Research

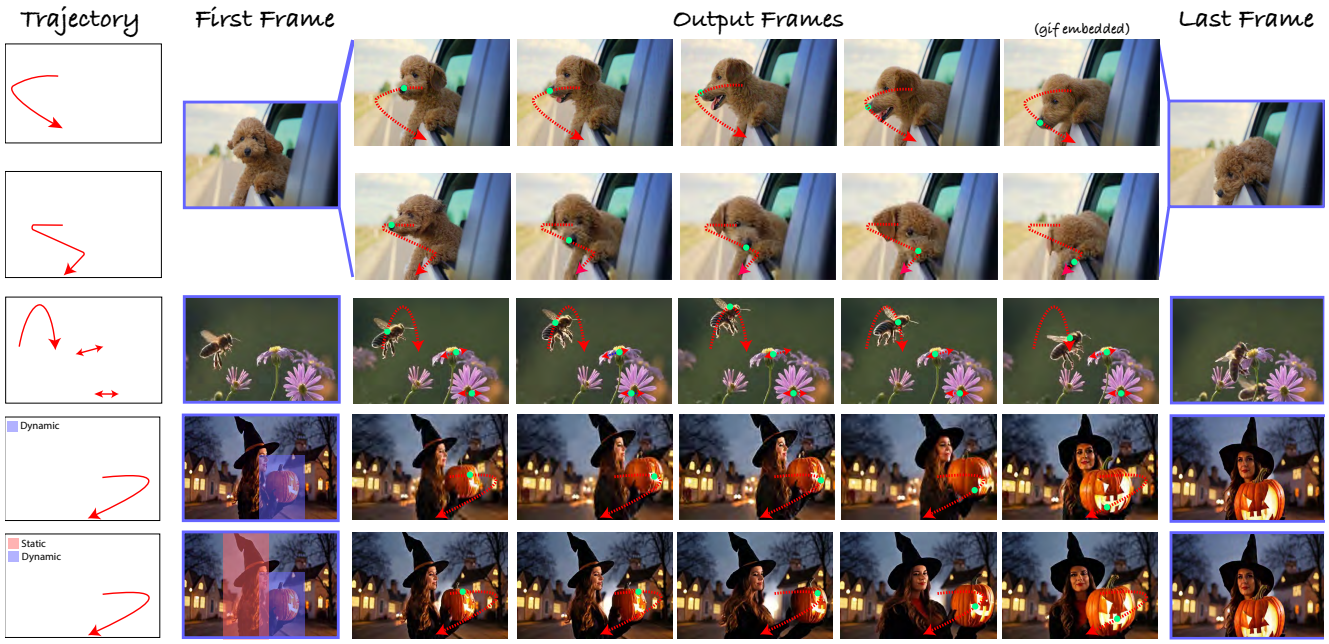

Figure 1. MotionBridge generates smooth and plausible transitions between two RGB images following user-defined trajectories, producing large and intricate motions (see top two rows for diverse results for the same dog). It offers multi-object control with motions varying between objects (bee + flowers), as well as mask control specifying static (red) vs. dynamic regions (blue); see last two rows. In last row, the static mask helps maintain the lady in the same position while turning her body naturally. Animated GIFs best viewed with Acrobat.

Abstract

By generating plausible and smooth transitions between two image frames, video inbetweening is an essential tool for video editing and long video synthesis. Traditional works lack the capability to generate complex large motions. While recent video generation techniques are powerful in creating high-quality results, they often lack fine control over the details of intermediate frames, which can lead to results that do not align with the creative mind. We introduce MotionBridge, a unified video inbetweening framework that allows flexible controls, including trajectory strokes, keyframes, masks, guide pixels, and text. However, learning such multi-modal controls in a unified framework is a challenging task. We thus design two generators to extract the control signal faithfully and encode features through dual-branch embedders to resolve ambiguities. We further introduce a curriculum training strategy to smoothly learn various controls. Extensive qualitative and quantitative experiments have demonstrated that such multi-modal controls enable a more dynamic, customizable, and contextually accurate visual narrative.

1. Introduction

Video Inbetweening refers to the process of generating intermediate frames between two keyframes, creating a smooth transition from one scene to another. It is becoming an increasingly important building block for video con-

*Work done during an internship at Adobe.

tent creators and animators to conduct video editing, storytelling, and short-to-long video synthesis [17, 32]. Frame interpolation is typically done in two steps, motion estimation and motion compensation [5, 8, 19, 26]. However, as the temporal or spatial gap between input frames widens, motion estimation and compensation become increasingly difficult, as generating realistic intermediate frames requires synthesizing novel content to bridge the missing information between inputs. With the emerging success of video generative models, the exploration space for the generated frames becomes larger, thus opening up new possibilities for inbetweening of distant inputs. At the same time, just blindly applying a foundational video model will typically not suffice since users are often not interested in just a possible interpolation result but one that follows their artistic expression through various means of control.

To this end, we propose MotionBridge, the initial effort for conducting controllable video inbetweening, which can generate diverse large motions through multi-modal controls (*e.g.*, trajectories, keyframes, masks, guide pixels, and text), in a unified framework. This allows users to generate dynamic, accurate, and customizable results. We take advantage of the Diffusion Transformer (DiT) architecture [23], which shows promising capability for generating long and high-quality videos. We design our model in a backbone-agnostic manner, which can work with different DiT designs/backbones.

Technically, our model is characterized by several core design choices to address the unique challenges of our task. 1). Rather than fusing all the control signals together all at once, to reduce ambiguity, we group controls into two categories: content control (*e.g.*, masks and guide pixels) and motion controls (*e.g.*, trajectories). We then utilize dual-branch embedders to compute the required features respectively before guiding the denoising process. 2). Representing video motion control with simple yet accurate representations is challenging. We propose a generator that synthesizes trajectories from optical flow and converts them into sparse RGB points as the motion representation used in model training. 3). We go beyond conventional trajectory control by complementing it with spatial content control such as masks and guide pixels. Through these, users can specify the regions they want to move or keep static. It helps further reduce ambiguity in the generation, offering a soft condition, as shown in the last example of Fig. 1 (last two rows). 4). With multi-modal controls, straightforward training does not work well, and we thus propose a curriculum learning strategy to ensure the model learns various controls smoothly. We feed the model with more dense and easy control, and gradually move to more sparse and high-level control.

We conduct extensive experiments to evaluate the effectiveness of MotionBridge both quantitatively and qualitatively. We also demonstrate several practical applications. We found our model is rather powerful and can go beyond the inbetweening scenarios, to work on controllable image-to-video (I2V) generation. Furthermore, our model can not only customize results but also improve the text-to-video (T2V) generation quality by reducing ambiguity. Our contributions are summarized as below:

- We take the initial effort to solve controllable video inbetweening task that supports multi-modal controls for customizing diverse large motions, in a unified framework.
- We group controls into two sets (*i.e.*, content and motion) and encode them through dual-branch embedders.
- We introduce two separate generators to extract compact control signals, and design a curriculum training strategy to learn multi-control sequentially.
- We demonstrate the flexibility and superior performance of our model through extensive experiments.

2. Related Work

2.1. Video Generation

Creating realistic and novel videos has long been an interesting research problem [25, 45]. Earlier studies have employed various generative models including GANs [29, 30, 34, 45] and temporally aware networks such as LSTM or autoregressive models [12, 33, 43]. 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 [9, 10, 13, 31]. Stable Video Diffusion [2] leverages latent diffusion models [3, 28] for generating temporally coherent content. Few-shot video generation is facilitated by methods like Tune-a-video [40], which fine-tunes pre-trained image diffusion models, while training-free methods [11] leverage large language models for generative guidance. Another approach to generating videos in a controllable manner is to use keyframes along with text conditions [7, 14, 38, 46], where initial frames are generated to guide subsequent frames, with latent-consistency networks ensuring temporal and visual coherence. However, our approach is different from such keyframe conditioning techniques as we aim to interpolate between two given frames following flexible multi-modal controls in a unified framework.

2.2. Video Inbetweening

Video inbetweening has many names such as frame interpolation, frame rate up-conversion, or temporal super-resolution. It has a long history, with early approaches operating at a block- instead of a pixel-level due to compute constraints at the time [5, 8]. While we have more compute nowadays, the underlying framework of motion estimation and compensation has largely remained the same through-

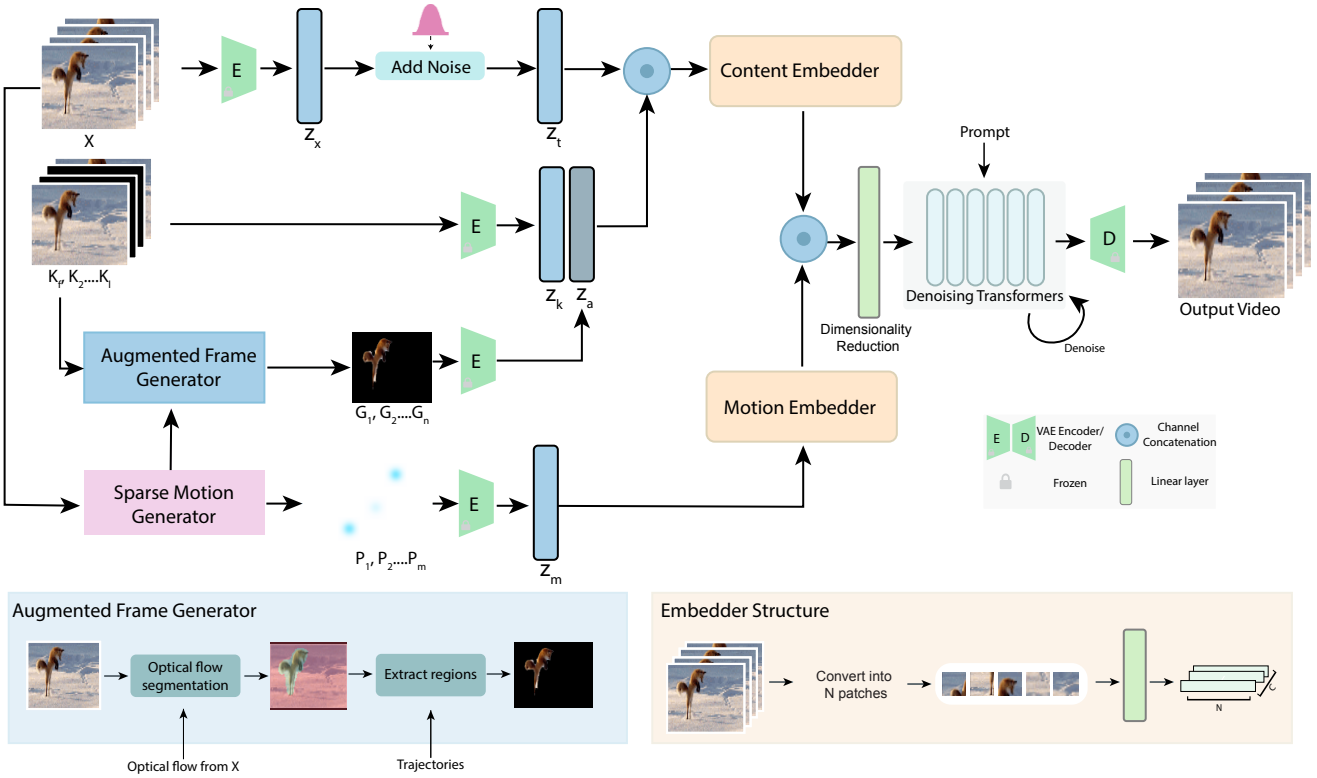

Figure 2. Overview of our MotionBridge pipeline. Given a video X, we propose a Sparse Motion Generator to provide conditioning for the motion trajectory with sparse RGB point controls, and an Augmented Frame Generator to compute guiding pixels for providing fine-grained control. The control signals are encoded through dual-branch embedders respectively to capture accurate content and motion features. Our model is flexible to take multi-modal controls for interpolation during the inference.

out the years [18, 19, 22, 26]. And even with approaches like phase- or kernel-based interpolation [16, 20, 21, 50], 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 in-betweening 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 [6], 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.

2.3. Motion Control

A variety of methods have recently been proposed for controllable video generation. Approaches such as MotionDirector [49], Customize-A-Video [27] and Tune-a-Video [40] learn motion patterns from a reference video, typically requiring fine-tuning for each template which can not only be cumbersome but is restricted to pre-existing motions. VideoComposer [37] and ToonCrafter [41] incorporate additional inputs, such as depth maps and sketches, to facilitate video generation. These types of input conditions are often difficult to generate, especially for an average user. DragNUWA [44] introduced trajectory-based control for video generation, allowing control over both object and camera motion. Subsequent works [15, 36, 39] build on this approach to improve precision and control, though they remain limited to single-image inputs. We propose a method that merges precise and intuitive motion control with the video inbetweening task.

3. Method

Traditional video inbetweening methods handle simple motions [20, 26], while recent diffusion-based methods [4, 6, 41, 42] boost the generation capability, but provides limited control by strongly relying on model priors and optional text guidance. To facilitate intuitive control, we propose a unified method called MotionBridge, using motion (*e.g.*, trajectories) and content (*e.g.*, masks, guide pixels) guidance to provide precise and user-friendly video inbetweening customization, as shown in Fig. 2.

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 trajectories consist of sparse RGB

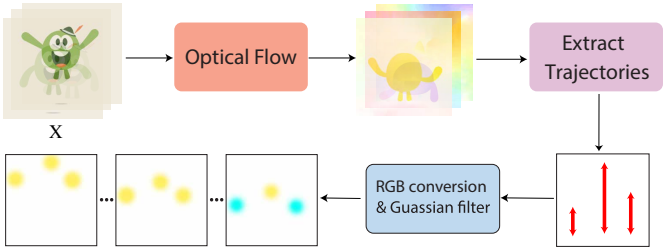

Figure 3. Structure of Sparse Motion Generator. The input video X is processed through an optical flow generator to extract trajectories. These are then filtered with a Gaussian filter and converted to images to create sparse RGB point controls.

points $\{P_1, P_2, \ldots, P_m\}$ and extract through the proposed *Sparse Motion Generator*; represent the content control as guide pixels $\{G_1, G_2, \ldots, G_n\}$ and extract through the proposed *Augmented Frame Generator*. Sparse points refer to sparse motion trajectory that the model needs to follow and guide pixels refer to specific regions in which the model is instructed to follow certain pixel values. They are integrated together through dual-branch embedders. We choose the DiT-based model as our training backbone due to the long video generation capability. In the remaining section, we provide an overview of the DiT-based model and a detailed discussion of our model design.

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 ϵ_θ represents the learned noise, and c represents the conditions.

Diffusion Transformer (DiT) [23] models combine diffusion-based denoising processes with transformer architectures. Compared to traditional UNet-based models like SVD, 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 ϵ: $\mathcal{L}_{diff} = ||\hat{\epsilon} - \epsilon||_2^2$. Our design is agnostic to the DiT backbone and can be used on different open-source codebases, such as OpenSora.

3.2. Control Generation

Large-motion interpolation is challenging due to ambiguity, artifacts, and distortions, requiring precise and high-quality control. Our project employs two mechanisms: the Sparse Motion Generator, which focuses on key motion paths, and the 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 Generator

The Sparse Motion Generator (Fig. 3) creates motion outputs aligned with the model and the input video X. Lacking motion trajectory data, we generate trajectories by extracting a dense optical flow from X and using feature tracking to get motion paths. Since this tracking corresponds to a single pixel, the output is too sparse to be meaningful. To improve interpretability, we expand feature locations using a Gaussian filter similar to [39], yielding a set of sparse trajectories.

Due to the patchify module in DiT, which divides input images into patches, aligning the extracted motion trajectories with the X patches is non-trivial. To address this, we converted trajectories into RGB format, which means mapping the direction/speed of motion to color space to create a visual representation of the movement, so that it can follow the same process as the keyframe inputs. The sparse trajectories are thus converted similarly into sparse RGB point controls $\{P_1, P_2, \ldots, P_m\} \in R^{H \times W \times 3}$. While a similar approach is explored in a concurrent work [48], we chose a simpler method. Instead of developing a custom VAE for motion, we utilized the DiT's existing VAE to effectively embed motion, which yielded successful results.

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, reduces the number of motion paths needed, and supports both static and dynamic regional control. 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 motion trajectories to guide the output. To implement this, we extract a region of interest (defined by a mask at inference time) from K_f and translate it across several frames according to the corresponding trajectory to create frames of guide pixels $\{G_1, G_2, \ldots, G_n\}$, which are appended to K_f temporally. For training, we generate masks 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.

One interesting application we identified for this training is the use of "guide pixels" for providing explicit conditions. Once the model learns to interpret guide pixels, 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 mask and guide pixel controls reduce 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 will be demonstrated in the Experiments section. During training, we randomly dropout this content condition by 20% to support mask-free control.

3.3. Curriculum Learning for Multi-modal Control 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. We always keep the first and last keyframe, and select 0-5 random keyframes in between. From these keyframes, we extract $\{G_1, G_2, \ldots, G_n\}$, 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, which includes both $\{K_f, K_2, \ldots, K_l\}$ and $\{G_1, G_2, \ldots, G_n\}$.

For motion, we extract $\{P_1, P_2, \ldots, P_m\}$ as discussed above. 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 guide pixels). These latent outputs are then passed through Embedders, which first transform the inputs into patches and then funnels the output through a linear layer. The output is again channel-concatenated and passed through a final linear layer before being fed into the transformer denoiser.

To train our model, we utilize a curriculum training strategy, where we gradually introduce conditional inputs to the model. First, the model is trained only on the image branch to develop a core image interpolation model.

Afterwards, to embed the motion as a condition, we first performed a dummy 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 quickly learns to ignore the motion input. This analysis is discussed in Sec. 4.5. To address this issue, we adopted an alternative approach inspired by [39, 44]. We first trained the model solely with optical flow 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 ($\{G_1, G_2, \ldots, G_n\}$).

Intuitively, we opted for a two-branch system to separate the two very different conditional inputs. To verify this design choice, we experiment with a single-branch system and share the findings in Sec. 4.5.

4. Experiments

To evaluate MotionBridge's performance, we conduct both quantitative and qualitative assessments across a variety of video sequences and datasets. For the quantitative assessment, we compare our model's generative quality and motion control.

Implementation Details: Our method is applied to a DiT text-to-video diffusion generative model. We use an Adam optimizer with 1×10^{-4} learning rate. Approximately 50k steps are used to train the image-to-video model, 2k steps for optical flow training, 5k for sparse and, 5k for mask input. The entire model, except the VAE and text encoders, is finetuned end-to-end.

Automatic Trajectory Generation: For fair comparison with baselines quantitatively, we utilize an automatic trajectory generation method. First, we generate optical flow between the first and last frames. Then, we use feature tracking to create three trajectories, which represent the shortest paths between the two frames.

Baselines: Currently, no dedicated methods exist for controllable inbetweening, so we utilize general interpolation techniques for comparison. For this, we select two types of baselines. First, we compare our method with recent diffusion-based video interpolation methods including Explorative Inbetweening of Time and Space (TimeReversal) [6], Dynamicrafter [42], and SEINE [4]. We also compare with FILM [26], a non-diffusion method for large motion interpolation.

Metrics and Datasets: We use FVD [35] and LPIPS [47] for quality comparison. Additionally, we introduce a "Motion" metric to evaluate our model's trajectory control. This metric uses the optical flow of the generated output to create trajectory paths corresponding to the input trajectory, and we compute the Fréchet Distance to assess their similarity. We show more details on this in supplementary. We use DAVIS [24] and Objectron [1] datasets for general analysis. We also curate a small dataset of 10 videos to analyze the effect of customized motion input.

4.1. Qualitative Results

As shown in Fig. 4, we demonstrate how our model effectively incorporates both content and motion controls. In the top row, the "fox" jumps along the specified trajectory, using a single motion vector combined with a mask to define the movement region. The 2^{nd} row showcases multi-trajectory results applied to multiple objects, while 3^{rd} row illustrates the model's capability to smoothly animate examples like moving a phone and turning a head. The last two

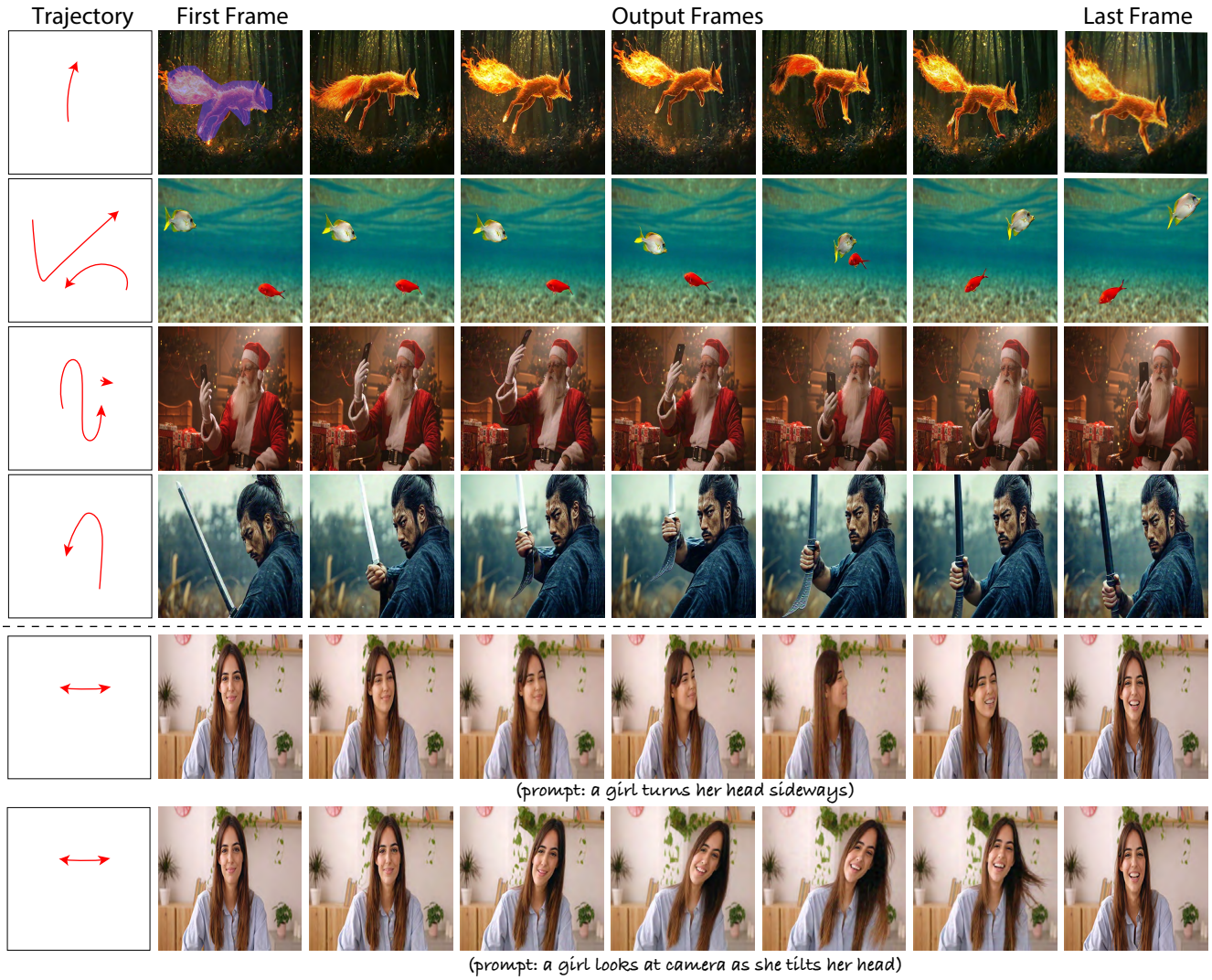

Figure 4. Our results. MotionBridge seamlessly integrates motion trajectories and two input frames, enabling smooth transitions. Additionally, we present an example using a mask to control the complete object movement in the first row. By further specifying different input prompts, the results are adapted accordingly, as shown in the last two rows.

rows show how effectively the prompt can reduce ambiguity. With the same motion trajectories, and same keyframes, we can adjust the outcome based on provided text.

Augmented frames offer an intriguing control where users can paste the interested pixels in the target location as guidance. Fig. 6 demonstrates example results of manually created augmented frames. By generating these frames and appending them to the end of the video, we are able to produce interesting interpolations.

We also provide a qualitative comparison with baseline methods in Fig. 5. Our model generates smooth transitions with minimal distortions and artifacts, resulting in natural-looking interpolations. In contrast, FILM [26] often morphs keyframes directly, leading to noticeable distortions. Dynamicrafter [42] tends to introduce features inconsistent with the keyframes (e.g., changing the appearance of a bike). Both TimeReversal [6] and SEINE [4] can produce distortions, artifacts, and unsteady motion, which our method effectively minimizes.

4.2. Quantitative Evaluation

Quantitative results are provided in Tab. 1. We randomly sample 100 samples from DAVIS [24] and Objectron [1] datasets. The results show that our method generates comparable or better results in terms of visual quality compared to the latest interpolation methods, while outperforming related works on motion control by a large margin.

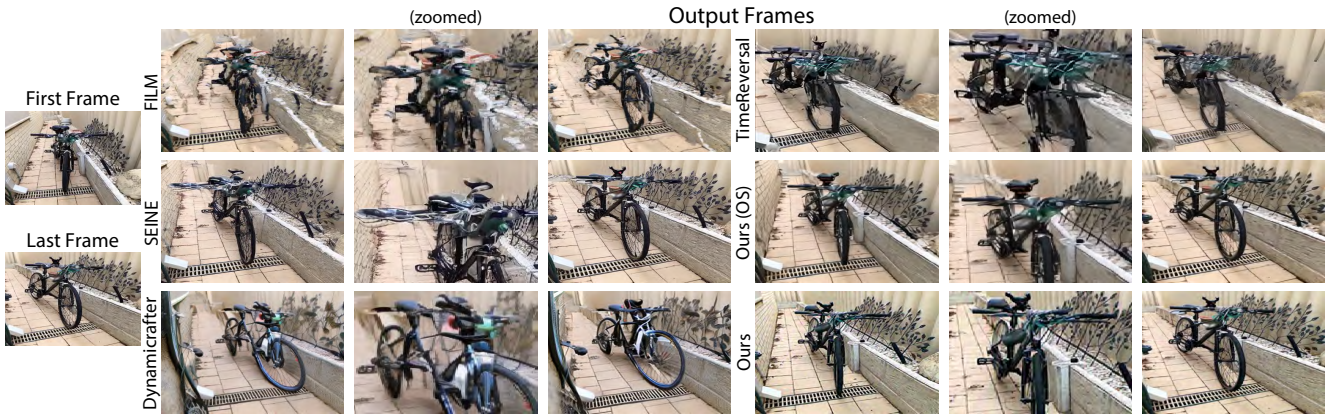

Figure 5. Qualitative comparisons. Our model can generate better interpolation results with less artifacts.

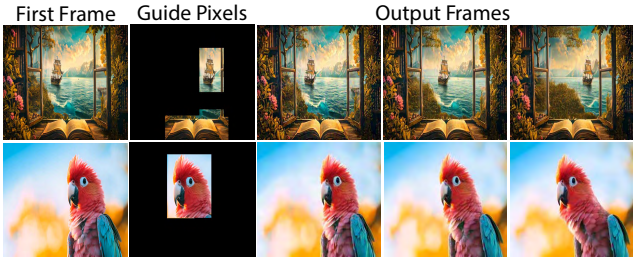

Figure 6. Example results of taking guide pixels as control. We paste the interested pixels in the target regions for the last frames.

Method	DAVIS			Objectron			Custom
	FVD↓	LPIPS↓	Motion↓	FVD↓	LPIPS↓	Motion↓	Motion↓
FILM	2.69	**0.29**	123	3.52	**0.34**	113	116
SEINE	1.72	0.33	166	2.48	0.38	167	108
Dynami	1.79	0.39	226	2.51	0.41	287	104
TimeReversal	**1.66**	0.36	128	3.61	0.53	132	121
Ours	**1.66**	0.34	114	**2.37**	0.37	116	**75**
Ours (OS)	1.80	0.35	**106**	2.57	0.37	**85**	78

Table 1. Quantitative comparisons with state-of-the-art video in-betweening models. We show our results on different base model architectures and "OS" represents OpenSora.

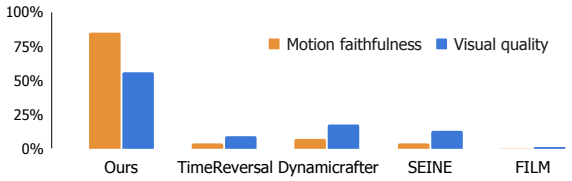

Figure 7. User study results. We show the human preference (%) over two factors: motion faithfulness and visual quality.

4.3. User Study

We have conducted user studies to evaluate two factors: **motion faithfulness** to measure whether the generated results have followed the input motion trajectory, and **visual quality** to measure whether the generated results are natural and smooth. We randomly select ten results from evaluation sets, and manually design several different motion trajectories for different samples. For each sample, we show results from different methods side by side in a random order and ask participants to choose the best one. More details can be found in the supplementary. We show the results in Fig. 7, and demonstrate that our model effectively follows motion while maintaining high-quality outputs.

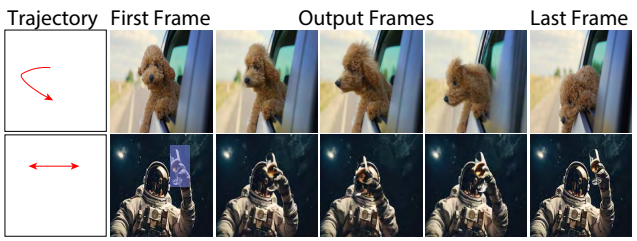

Figure 8. We verify generalizability of our method by showing results using OpenSora as the base model.

4.4. Generalizability

Our method is designed to work in a backbone-agonistic manner, so it is easy to apply to DiT models with different structures. To verify this generalizability, we show some results with OpenSora (**OS**) in Fig. 8. We also use OpenSora to generate quantitative results shared in Tab. 7.

4.5. Ablation Study

The Effectiveness of Curriculum Training. We initially experimented with training directly from sparse motions. However, this approach failed to integrate the motion information, leading the model to focus only on interpolating between the two frames. As shown in Fig. 9, the generated frames effectively bridge the input images but disregard the provided motion cues.

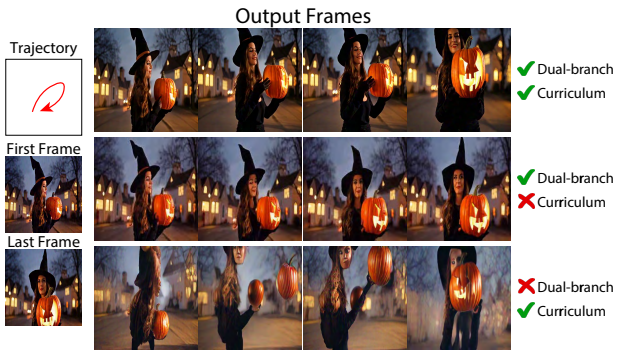

Figure 9. Results of ablation studies. The second row illustrates how the absence of curriculum training prevents the model from accurately recognizing sparse motion trajectories. The last row highlights the importance of dual-branch embedders, showing that using a single branch results in significant distortions.

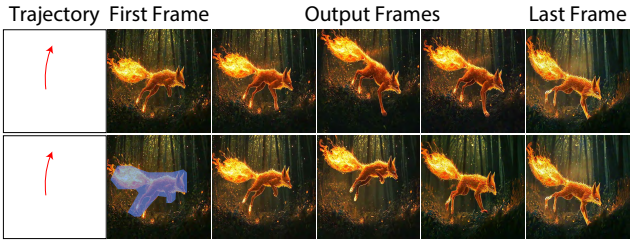

Figure 10. Results of using mask control. It shows how masks enable targeted region movement with fewer sparse input paths. Without a mask (top), movement can be ignored or misdirected (e.g., the fox's back shifts upward), while with a mask (bottom), we achieve the intended jump animation.

The Effectiveness of Dual-branch Embedders. In our system, we use two branches to embed content and motion information. In this ablation, we show results using a single branch. Fig. 9 demonstrates that significantly more artifacts are present if the two conditions are not separated.

The Effectiveness of Content Control. We show the effect of adding masks as opposed to purely using motion vectors as input. Using masks not only improves the reliability and predictability of the results by providing an intuitive control to the user but it also allows for more complicated motions as shown in Fig. 10 and also Fig. 12 (top).

5. Applications

Our model's versatility supports a wide range of applications and can integrate with existing text-to-video and image-to-video models to refine results with added controls. Here, we showcase several use cases with additional examples available in the supplementary material.

Looping Video Generation. One application of our model is looping video generation. When the two input frames are identical, our model generates seamless looping videos that precisely follow the input's motion trajectories, enabling smooth, continuous playback. This capability, illustrated in Fig. 11, is particularly valuable for applications requiring immersive and repetitive animations, such as digital art, virtual environments, and background animations.

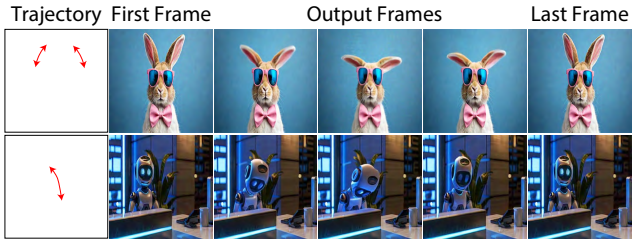

Figure 11. Looping video generation. Motion becomes periodic by shortening the trajectory to less than the clip's duration.

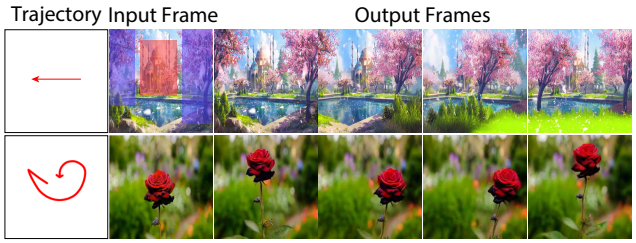

Figure 12. Animation results from a single frame. We show the combination of motion trajectory and mask control in the first row and only trajectory control in the second row. Even without training for the single image animation, our model can still generate plausible good results.

Image Animation. Our model also supports single-image animation, expanding its flexibility beyond inbetweening tasks. Although trained for inbetweening, it can animate a single frame by generating plausible motions, as shown in Fig. 12. This feature enables creative applications such as bringing static images to life, producing engaging animations from still photos, and enhancing digital storytelling.

6. Conclusion

We introduced MotionBridge a DiT based framework to address the task of controllable inbetweening. Our method is capable of generating high-quality interpolated frames guided by inputs such as keyframes, trajectories and masks. We show the versatility of our method through extensive experiments and applications. While our model performs well, we have identified several areas for potential improvement in future work. For instance, the trajectories and image content must maintain a certain level of alignment; otherwise, the motion may be overshadowed by the stronger image condition. Additionally, our mask control is limited to 2D translation, meaning that 3D movements, such as

rotation, cannot currently be captured through masks. Expanding this capability to include 3D transformations would be a valuable direction in future versions.

References

[1] Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. 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*, pages 7822–7831, 2021. 5, 6

[2] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 2

[3] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023. 2

[4] Xinyuan Chen, Yaohui Wang, Lingjun Zhang, Shaobin Zhuang, Xin Ma, Jiashuo Yu, Yali Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Seine: Short-to-long video diffusion model for generative transition and prediction. In *The Twelfth International Conference on Learning Representations*, 2023. 3, 5, 6

[5] Byung-Tae Choi, Sung-Hee Lee, and Sung-Jea Ko. New frame rate up-conversion using bi-directional motion estimation. *IEEE Trans. Consumer Electron.*, 46(3):603–609, 2000. 2

[6] Haiwen Feng, Zheng Ding, Zhihao Xia, Simon Niklaus, Victoria Abrevaya, Michael J Black, and Xuaner Zhang. Explorative inbetweening of time and space. *arXiv preprint arXiv:2403.14611*, 2024. 3, 5, 6

[7] Rohit Girdhar, Mannat Singh, Andrew Brown, Quentin Duval, Samaneh Azadi, Sai Saketh Rambhatla, Akbar Shah, Xi Yin, Devi Parikh, and Ishan Misra. Emu video: Factorizing text-to-video generation by explicit image conditioning. *arXiv preprint arXiv:2311.10709*, 2023. 2

[8] Taehyeun Ha, Seongjoo Lee, and Jaeseok Kim. Motion compensated frame interpolation by new block-based motion estimation algorithm. *IEEE Trans. Consumer Electron.*, 50(2): 752–759, 2004. 2

[9] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 2

[10] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022. 2

[11] Susung Hong, Junyoung Seo, Heeseong Shin, Sunghwan Hong, and Seungryong Kim. Large language models are frame-level directors for zero-shot text-to-video generation. In *First Workshop on Controllable Video Generation@ ICML24*, 2023. 2

[12] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers, 2022. 2

[13] Mingi Kwon, Seoung Wug Oh, Yang Zhou, Difan Liu, Joon-Young Lee, Haoran Cai, Baqiao Liu, Feng Liu, and Youngjung Uh. Harivo: Harnessing text-to-image models for video generation. *arXiv preprint arXiv:2410.07763*, 2024. 2

[14] Xin Li, Wenqing Chu, Ye Wu, Weihang Yuan, Fanglong Liu, Qi Zhang, Fu Li, Haocheng Feng, Errui Ding, and Jingdong Wang. Videogen: A reference-guided latent diffusion approach for high definition text-to-video generation. *arXiv preprint arXiv:2309.00398*, 2023. 2

[15] Wan-Duo Kurt Ma, John P Lewis, and W Bastiaan Kleijn. Trailblazer: Trajectory control for diffusion-based video generation. *arXiv preprint arXiv:2401.00896*, 2023. 3

[16] Simone Meyer, Oliver Wang, Henning Zimmer, Max Grosse, and Alexander Sorkine-Hornung. Phase-based frame interpolation for video. In *CVPR*, pages 1410–1418. IEEE Computer Society, 2015. 3

[17] Simone Meyer, Victor Cornillère, Abdelaziz Djelouah, Christopher Schroers, and Markus H. Gross. Deep video color propagation. In *BMVC*, page 128. BMVA Press, 2018. 2

[18] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *CVPR*, pages 1701–1710. Computer Vision Foundation / IEEE Computer Society, 2018. 3

[19] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *CVPR*, pages 5436–5445. Computer Vision Foundation / IEEE, 2020. 2, 3

[20] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 670–679, 2017. 3

[21] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *ICCV*, pages 261–270. IEEE Computer Society, 2017. 3

[22] Simon Niklaus, Ping Hu, and Jiawen Chen. Splatting-based synthesis for video frame interpolation. In *WACV*, pages 713–723. IEEE, 2023. 3

[23] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 2, 4

[24] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 5, 6

[25] MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: a baseline for generative models of natural videos, 2016. 2

[26] Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. Film: Frame inter-

polation for large motion. In *European Conference on Computer Vision*, pages 250–266. Springer, 2022. 2, 3, 5, 6

[27] Yixuan Ren, Yang Zhou, Jimei Yang, Jing Shi, Difan Liu, Feng Liu, Mingi Kwon, and Abhinav Shrivastava. Customize-a-video: One-shot motion customization of text-to-video diffusion models. *arXiv preprint arXiv:2402.14780*, 2024. 3

[28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2

[29] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping, 2017. 2

[30] Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Mostgan-v: Video generation with temporal motion styles, 2023. 2

[31] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 2

[32] Li Siyao, Shiyu Zhao, Weijiang Yu, Wenxiu Sun, Dimitris N. Metaxas, Chen Change Loy, and Ziwei Liu. Deep animation video interpolation in the wild. In *CVPR*, pages 6587–6595. Computer Vision Foundation / IEEE, 2021. 2

[33] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms, 2016. 2

[34] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation, 2017. 2

[35] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Fvd: A new metric for video generation. In *ICLR 2019 Workshop DeepGenStruct*, 2019. 5

[36] Jiawei Wang, Yuchen Zhang, Jiaxin Zou, Yan Zeng, Guoqiang Wei, Liping Yuan, and Hang Li. Boximator: Generating rich and controllable motions for video synthesis. *arXiv preprint arXiv:2402.01566*, 2024. 3

[37] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion controllability. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[38] Yanhui Wang, Jianmin Bao, Wenming Weng, Ruoyu Feng, Dacheng Yin, Tao Yang, Jingxu Zhang, Qi Dai, Zhiyuan Zhao, Chunyu Wang, 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*, pages 8414–8424, 2024. 2

[39] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 3, 4, 5

[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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7623–7633, 2023. 2, 3

[41] Jinbo Xing, Hanyuan Liu, Menghan Xia, Yong Zhang, Xintao Wang, Ying Shan, and Tien-Tsin Wong. Tooncrafter: Generative cartoon interpolation. *arXiv preprint arXiv:2405.17933*, 2024. 3

[42] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Gongye Liu, Xintao Wang, Ying Shan, and Tien-Tsin Wong. Dynamicrafter: Animating open-domain images with video diffusion priors. In *European Conference on Computer Vision*, pages 399–417. Springer, 2025. 3, 5, 6

[43] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers, 2021. 2

[44] 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. *arXiv preprint arXiv:2308.08089*, 2023. 3, 5

[45] Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G. Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, and Lu Jiang. Magvit: Masked generative video transformer, 2023. 2

[46] Yan Zeng, Guoqiang Wei, Jiani Zheng, Jiaxin Zou, Yang Wei, Yuchen Zhang, and Hang Li. Make pixels dance: High-dynamic video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8850–8860, 2024. 2

[47] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 5

[48] Zhenghao Zhang, Junchao Liao, Menghao Li, Long Qin, and Weizhi Wang. Tora: Trajectory-oriented diffusion transformer for video generation. *arXiv preprint arXiv:2407.21705*, 2024. 4

[49] Rui Zhao, Yuchao Gu, Jay Zhangjie Wu, David Junhao Zhang, Jia-Wei Liu, Weijia Wu, Jussi Keppo, and Mike Zheng Shou. Motiondirector: Motion customization of text-to-video diffusion models. In *European Conference on Computer Vision*, pages 273–290. Springer, 2025. 3

[50] Yang Zhou, Jimei Yang, Dingzeyu Li, Jun Saito, Deepali Aneja, and Evangelos Kalogerakis. Audio-driven neural gesture reenactment with video motion graphs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3418–3428, 2022. 3