Motion Modes: What Could Happen Next?

Supplementary Material

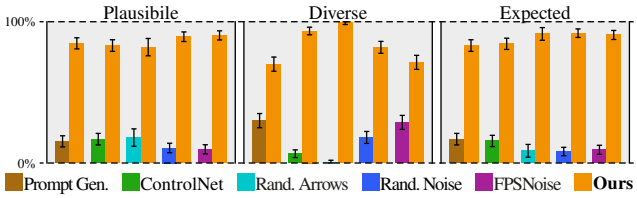

Figure 8. **Extended user study.** We compare the *plausible*, *diverse*, and *expected* nature of our motions to five baselines, including the Random Arrows baseline. Each pair of bars shows the percentage of comparisons in which our method or a baseline was judged favorably with 95% confidence intervals.

A. Overview

In this appendix, we present extended versions of the user study (Section B) and the ablation study (Section C). Additionally, we examine how much the generated motions $\mathcal{X}$ differ for different generator seeds (Section D), how much a given motion constrains the video generator by showing different videos generated for the same motion (Section E), show examples of secondary motions F, and apply our method to augment one additional drag-based image editor (Section G). We also provide additional implementation and timing details (Section H), a more detailed description for baselines (Section I) and more details on the arrow-based motion prompting application (Section J).

Our project website, `https://motionmodes.github.io`, also contains, among other details, a full qualitative comparison on 28 images, results of our method on a total of 54 different input images, and our arrow-based motion prompting application using a different video generator [25].

B. Extended User Study

In Figure 8, we present an extended version of the user study that includes the random arrows baseline. Results for this baseline are collected from 16 instead of 32 participants, the other study details are the same as for all other baselines. Results confirm our findings for all other baselines: users find our motions significantly more plausible and diverse, and they also better agree with the motions users expected for the selected object.

C. Extended Ablation

In Table 3, we provide an extended ablation study that includes an ablation of the smoothness guidance. Apart from its function as regularizer, surprisingly, this energy also improves object focus, i.e. it tends to better avoid static objects. Our interpretation is that object motions are suppressed by the motion generator's prior during the denoising process if they start out unrealistically jerky or jittery. Our smoothness energy guides the denoising trajectory away from these bad object motions early on, resulting in a less suppression from the prior.

Table 3. **Extended ablation** of key components with metrics based on *diverse*, *focused* metrics and their tradeoff $\bar{E} := 0.5(\bar{E}_d + \bar{E}_f)$. Underlined values are closer to the best than to the worst value.

		div.	focused		
	$\bar{E}\downarrow$	$\bar{E}_d\downarrow$	$\bar{E}_f\downarrow$	($\bar{E}_c\downarrow$	$\bar{E}_o\downarrow$)
without E_c	0.83	1.02	0.64	1.29	**0.00**
without E_o	0.97	1.03	0.91	**0.06**	1.75
without E_d	0.72	1.36	0.08	0.13	0.04
without E_s	0.58	1.02	0.13	0.10	0.16
FPS instead of E_d	0.79	1.49	0.10	0.11	0.08
ControlNet instead of E_c,E_o	0.88	**0.96**	0.80	0.15	1.45
Motion Modes	**0.55**	1.04	**0.07**	0.09	0.05

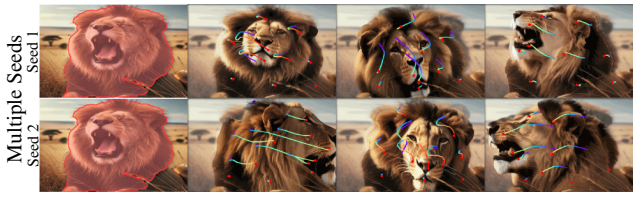

Figure 9. **Multiple seeds for one input image.** We generate multiple videos from the same motion $\mathbf{x}$. They differ in small details, but overall follow the motion accurately.

Figure 10. **Multiple videos from one motion.** We generate multiple videos from the same motion $\mathbf{x}$. They differ in small details, but overall follow the motion accurately.

D. Multiple Seeds for One Input Image

To show the variance a user may expect from our motion generator, in Figure 9 we show to different sets of motions for the same input image, generated with two different seeds. We can see that the variety of motions is similar, although slightly different motions are found in each case.

E. Multiple Videos Generated for One Motion

All videos in our experiments are obtained by first generating a motion $\mathbf{x}$ and then generating a video conditioned on $\mathbf{x}$. To examine how closely the generated video follows $\mathbf{x}$, in Figure 10, we show multiple videos generated conditioned

Figure 11. **Secondary motion.** The video prior generates secondary motions outside of the masked region if necessary to maintain the plausibility of a video, such as motions of the steam of the train engine, the reflection of the boat, and the shadow of the tank (secondary motions are visualized with purple flow trajectories).

on the same motion $\mathbf{x}$ from different random noises. We can see that small details are different, but overall, the motions of the different videos are similar to each other and follow the generated motion $\mathbf{x}$ accurately.

F. Secondary Motion

While our method encourages motions to be focused only on the masked region, the prior of the video generator ensures that any secondary motions that are caused by the motion in the masked region are also generated if necessary to maintain the overall plausibility of the video. A few examples are shown in Figure 11, like the steam of the train engine, the reflection of the boat, and the shadow of the tank.

G. Application to Puppet-Master

We demonstrate our application that augments coarse drag motion inputs on one additional drag-based image editor Puppet-Master [21]. Similar to our result in Figure 6 of the main paper, we can see that the more detailed input motion provides enough information to the drag-based image editor to avoid ambiguities and implausible results.

H. Implementation Details

Guided Denoising As described in the paper, we use the flow generation module from Motion-I2V [29] as our backbone. We further disconnect the ControlNet module described in their paper, as we don't need the conditioning and

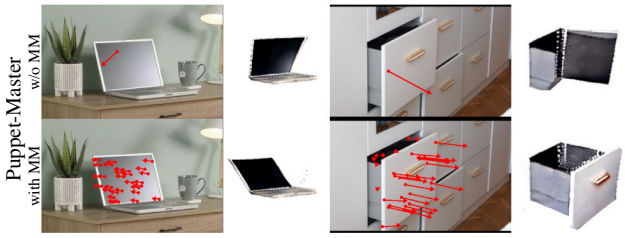

Figure 12. **Arrow-based prompting with Puppet Master.** We use Motion Modes to augment coarse drag motion inputs for the drag-based image editor Puppet-Master [21], avoiding results that are implausible (right) or do not follow the input drag (left).

we found that the constraints from ControlNet conditioning limits the diversity of our motions. The flow generator uses 25 total timesteps for denoising out of which the first 20 timesteps are guided in our approach.

Timing and Memory In our experiments, we further used gradient checkpointing on the U-Net to minimize the memory cost of backpropagating the guidance gradients in each denoising timestep. Given the time cost of gradient checkpointing and additional memory costs of backpropagation, our guided denoising approach has a peak memory usage of 21.7GB and requires on average 2 minutes 35 seconds to fully denoise a sample across 25 timesteps. Unguided vanilla denoising, on the other hand, has 12.3GB peak memory usage and requires 1 minute 18 seconds on average to fully denoise a sample.

I. Additional Baseline Details

Prompt Generation. Our backbone Motion-I2V [29] supports text-conditioning for image-to-video generation. In the Prompt Generation baseline, we aim to sample diverse and focused object motions using a set of distinct text prompts. To automate this process, we use GPT-4 to generate text prompts that correspond to distinct object motions for a given input image and object. The prompts are then used as text conditioning for Motion-I2V for video generation.

Specifically, we query GPT-4 for the prompts as follows. GPT-4 is first provided the following context: *"I am using a text-based video generator to discover all the different ways a specific object in an image can move, and I wish to generate a set of text prompts in order to achieve this. In particular, I will provide an image and specify an object. For each such specification, I would like to generate 6 text prompts that can be input to the video generator in order to explore the distinct motions the specified object can have in the scene. Remember that we want the motions to be focused only on the specified object and to each be distinct from the other."* We then provide the model with an image along with a text specification of the object in the context of the same conversation to retrieve the text prompts. Some examples of retrieved prompts follow. For a scene with a basketball

near a net: *"video of a basketball swishing through the hoop after a jump shot"*, *"video of a basketball bouncing off the rim and falling away from the hoop"*, *"video of a basketball spinning around the rim before dropping in"*. For a scene with a cat on a ledge: *"video of a cat walking gracefully along a ledge with a scenic background"*, *"video of a cat jumping off the ledge gracefully"*, *"video of a cat stopping and looking around curiously"*.

Random Arrows. Our backbone Motion-I2V [29] can be conditioned on a drag arrow that describes the rough motion direction and motion magnitude of a point in the image, in an application the authors call *MotionDrag*. In the Random Arrows baseline, we use random drag arrows to explore a diverse set of motions for a selected object. Specifically, given an object mask $\mathbf{m}$, we set the starting point for the drag arrow to a random point inside the object mask, randomly sample a direction, and sample the length of the drag arrow uniformly from an interval of reasonable lengths (20 to 80 pixels in an image with 320p resolution). We found that arrow lengths outside this interval tended to either result in zero object motion or implausible motions.

J. Additional Arrow-based Prompting Details

Our arrow-based prompting application shows that Motion Modes can be used to facilitate user interaction with drag-controlled image editors and video generators. As image editors, we work with Drag-A-Part [20] and DragonDiffusion [24], and as video editors, we use MOFA [25] and the *MotionDrag* application of Motion-I2V [29]. We take as input a given drag arrow, defined by a start point $\mathbf{a} \in [1, H] \times [1, W]$ and end point $\mathbf{b} \in [1, H] \times [1, W]$, both given as pixel indices for resolution $W \times H$. We then use this drag arrow to retrieve the closest motion $\mathbf{x}$ from our motion set $\mathcal{X}$. Recall that in each frame, our motions describe the same offset of each image point from its starting position as a drag arrow. Thus we can simply compare the drag arrow to each frame of the motion $\mathbf{x}$ at the starting position $\mathbf{a}$ of the drag arrow:

$$\min_{\mathrm{k}} \left\| \mathbf{x}_{k,\mathbf{a}} - \overrightarrow{\mathbf{ab}} \right\|_2, \tag{5}$$

where $\mathbf{x}_{k,\mathbf{a}}$ is the offset vector of the motion $\mathbf{x}$ in frame k at the starting point $\mathbf{a}$ of the drag arrow. The motion $\mathbf{x}$ with closest distance to the drag arrow describes a motion similar to the drag arrow, but typically has good plausibility and much more detail than the drag arrow. We then convert the retrieved motion back into a representation that the image or video editors can use as input. Specifically, Drag-A-Part can take up to 10 drag arrows as input, for DragonDiffusion, we can fit up to 100 arrows into memory, for MOFA, we use up to 50 arrows (we found that more arrows result in non-static backgrounds), and for Motion-I2V, we can directly provide the retrieved motion $\mathbf{x}$ as conditional input. To convert a motion to n drag arrows, we cluster the offsets in the retrieved frame of the motion into n clusters using K-Means, and use the cluster means as drag arrows.

References

[1] Omri Avrahami, Rinon Gal, Gal Chechik, Ohad Fried, Dani Lischinski, Arash Vahdat, and Weili Nie. Diffuhaul: A training-free method for object dragging in images. *arXiv preprint arXiv:2406.01594*, 2024. 3

[2] Hugo Bertiche, Niloy J. Mitra, Kuldeep Kulkarni, Chun-Hao Paul Huang, Tuanfeng Y. Wang, Meysam Madadi, Sergio Escalera, and Duygu Ceylan. Blowing in the wind: Cyclenet for human cinemagraphs from still images. In *CVPR*, 2023. 1, 3

[3] Andreas Blattmann, Timo Milbich, Michael Dorkenwald, and Björn Ommer. ipoke: Poking a still image for controlled stochastic video synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14707–14717, 2021. 3

[4] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets, 2023. 1

[5] 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

[6] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. 2

[7] Tsai-Shien Chen, Chieh Hubert Lin, Hung-Yu Tseng, Tsung-Yi Lin, and Ming-Hsuan Yang. Motion-conditioned diffusion model for controllable video synthesis. *arXiv preprint arXiv:2304.14404*, 2023. 3

[8] Gabriele Corso, Yilun Xu, Valentin De Bortoli, Regina Barzilay, and Tommi S. Jaakkola. Particle guidance: non-i.i.d. diverse sampling with diffusion models. In *ICLR*, 2024. 1, 3

[9] Zuozhuo Dai, Zhenghao Zhang, Yao Yao, Bingxue Qiu, Siyu Zhu, Long Qin, and Weizhi Wang. Animateanything: Fine-grained open domain image animation with motion guidance, 2023. 2

[10] Abe Davis, Michael Rubinstein, Neal Wadhwa, Gautham J Mysore, Fredo Durand, and William T Freeman. Interactive dynamic video. *ACM TOG (SIGGRAPH)*, 34(4):1–9, 2015. 2

[11] Aram Davtyan and Paolo Favaro. Learn the force we can: Enabling sparse motion control in multi-object video generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11722–11730, 2024. 3

[12] Dave Epstein, Allan Jabri, Ben Poole, Alexei A. Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation, 2023. 4

[13] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls.

In *NeurIPS*, pages 9841–9850. Curran Associates, Inc., 2020. 2

[14] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. 4

[15] 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

[16] Ruizhen Hu, Wenchao Li, Oliver Van Kaick, Ariel Shamir, Hao Zhang, and Hui Huang. Learning to predict part mobility from a single static snapshot. *ACM TOG (SIGGRAPH)*, 36 (6), 2017. 2

[17] Yash Jain, Anshul Nasery, Vibhav Vineet, and Harkirat Behl. Peekaboo: Interactive video generation via masked-diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8079–8088, 2024. 3

[18] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 4

[19] Guojun Lei, Chi Wang, Hong Li, Rong Zhang, Yikai Wang, and Weiwei Xu. Animateanything: Consistent and controllable animation for video generation. *arXiv preprint arXiv:2411.10836*, 2024. 3

[20] Ruining Li, Chuanxia Zheng, Christian Rupprecht, and Andrea Vedaldi. Dragapart: Learning a part-level motion prior for articulated objects. In *ECCV*, 2024. 1, 3, 6

[21] Ruining Li, Chuanxia Zheng, Christian Rupprecht, and Andrea Vedaldi. Puppet-master: Scaling interactive video generation as a motion prior for part-level dynamics. *arXiv preprint arXiv:2408.04631*, 2024. 3, 2

[22] Zhengqi Li, Richard Tucker, Noah Snavely, and Aleksander Holynski. Generative image dynamics. In *CVPR*, 2024. 1, 2

[23] Niloy J. Mitra, Yong-Liang Yang, Dong-Ming Yan, Wilmot Li, and Maneesh Agrawala. Illustrating how mechanical assemblies work. *ACM TOG (SIGGRAPH)*, 29(3):58:1–58:12, 2010. 2

[24] Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. Dragondiffusion: Enabling drag-style manipulation on diffusion models. In *ICLR*, 2024. 3, 6

[25] Muyao Niu, Xiaodong Cun, Xintao Wang, Yong Zhang, Ying Shan, and Yinqiang Zheng. Mofa-video: Controllable image animation via generative motion field adaptions in frozen image-to-video diffusion model. *ECCV*, 2024. 1, 3

[26] Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie Liu, Abhimitra Meka, and Christian Theobalt. Drag your gan: Interactive point-based manipulation on the generative image manifold. In *ACM TOG (SIGGRAPH)*, page 1–11, 2023. 3

[27] Karran Pandey, Paul Guerrero, Matheus Gadelha, Yannick Hold-Geoffroy, Karan Singh, and Niloy J. Mitra. Diffusion handles: Enabling 3d edits for diffusion models by lifting activations to 3d. 2024. 3, 4

[28] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024. 2

[29] Xiaoyu Shi, Zhaoyang Huang, Fu-Yun Wang, Weikang Bian, Dasong Li, Yi Zhang, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, et al. Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 1, 2, 3, 4, 5, 6

[30] Yujun Shi, Jun Hao Liew, Hanshu Yan, Vincent Y. F. Tan, and Jiashi Feng. Lightningdrag: Lightning fast and accurate drag-based image editing emerging from videos, 2024. 3

[31] Yujun Shi, Chuhui Xue, Jun Hao Liew, Jiachun Pan, Hanshu Yan, Wenqing Zhang, Vincent Y. F. Tan, and Song Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. In *CVPR*, pages 8839–8849, 2024. 3

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

[33] 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*, 36:7594–7611, 2023. 3

[34] Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, David Junhao Zhang, Mike Zheng Shou, Yan Li, Tingting Gao, and Di Zhang. Draganything: Motion control for anything using entity representation, 2024. 3

[35] Zeqi Xiao, Yifan Zhou, Shuai Yang, and Xingang Pan. Video diffusion models are training-free motion interpreter and controller. *arXiv preprint arXiv:2405.14864*, 2024. 2, 3

[36] Haitao Zhou, Chuang Wang, Rui Nie, Jinlin Liu, Dongdong Yu, Qian Yu, and Changhu Wang. Trackgo: A flexible and efficient method for controllable video generation. *arXiv preprint arXiv:2408.11475*, 2024. 3