AID: Adapting Image2Video Diffusion Models for Instruction-guided Video Prediction
(— Supplementary Materials —)

A. Additional Experimental Results

A.1. Additional Results on UCF-101

Our method AID, is primarily designed for the text-guided video prediction(TVP) task and is validated on task-level datasets. Most previous text-guided video generation methods [12, 13, 24, 33] use UCF-101 [25] as a benchmark for testing. Although class-conditioned video prediction on UCF-101 is not particularly suitable for TVP tasks, we still conduct experiments under two settings to evaluate the video generation performance.

Settings We fine-tune our model on the UCF-101 dataset, resizing the video resolution to 256x256 with 16 frames per video. We conduct experiments under two settings: predicting videos conditioned on 1 and 5 reference frames following [11, 14]. We report FVD and FID metrics following the methods of Seer [11] and VDM [13]. During the testing phase, we sample 2,048 samples from the test set following [11]. We perform class-conditioned video prediction on this dataset by writing one sentence for each class as the caption for video generation, following the PYoCo [8] method. For example, we rewrite "biking" as "A person is riding a bicycle."

Results We present the class-conditioned video prediction results on UCF-101 in Table 1. When given 1 reference frame, our method significantly outperforms other video generation models in Fréchet inception distance (FID) [21] and achieves comparable Fréchet video distance (FVD) [28] results to the large-scale pre-trained Make-A-Video [24] method. For the TVP task with 5 reference frames, our method performs much better than other methods. Additionally, we provide qualitative results in Figure 1 and Figure 2.

A.2. Additional Results of Ablation Study

Comparisons of Different Fine-tune Strategies In Table 2, we compare and analyze the impact of different fine-tuning strategies on performance. We selected SVD [2] and Open-Sora [32] as two baselines and conducted experimental validation on SSv2 [10], the largest dataset. Both methods exhibited poor zero-shot performance, indicating that even general image-to-video models show limited quality in generating domain-specific first-person perspective videos. Fine-tuning all parameters proves effective for adapting to this domain, but it is constrained by coarse text-based control, resulting in a significant gap compared to our method. Finally, we compared the reconstruction performance of VAE, which serves as an upper baseline.

Visualization of Main Ablations In the main text, we present the quantitative ablation results for the MCondition and Adapters design. Here, we will show the qualitative ablation comparison results. As illustrated in Fig. 4, we demonstrate the importance of the various components of MCondition. It is evident that the multi-modal branch is crucial, as it effectively aligns and integrates the instruction with the initial frames, guiding the subsequent generation steps. The decomposed branch significantly enhances the stability and consistency of the predicted videos. We also show in Figure 5 that removing different adapters results in varying degrees of performance degradation, demonstrating the effectiveness of our proposed spatial and temporal adapters.

The Comparison of Computation Efficiency To validate the computational efficiency of our method, we compare the training time and GPU usage of our approach with Seer and VideoFusion on a single 80G NVIDIA A100 GPU in Table 3. Our

Table 1. Class-conditioned video prediction performance on UCF-101. We evaluate the AID on the UCF-101 with 16-frames-long videos. Ex.data indicates that the model has been pre-trained or fine-tuned on extra datasets.

Method	**Ex.data**	**Cond.**	**Resolution**	**FVD (↓)**	**FID (↓)**
MoCoGAN [27]	No	No	64×64	-	26998
MoCoGAN-HD [26]	No	Class.	256×256	700	-
TGAN-ODE [9]	No	No	64×64	-	26512
TGAN-F [15]	No	No	128×128	-	7817
DIGAN [31]	No	No	-	577	-
TGANv2 [23]	No	Class.	128×128	1431	3497
VDM [13]	No	No	64×64	-	295
TATS-base [7]	No	Class.	128×128	278	-
MCVD [29]	No	No	64×64	1143	-
LVDM [12]	No	No	256×256	372	-
MAGVIT-B [30]	No	Class.	128×128	159	-
PYoCo [8]	No	No	256×256	310	-
Dysen-VDM [8]	No	No	256×256	255	-
VDT [18]	No	No	64×64	226	-
VideoFusion [19]	txt-video	Class.	128×128	173	-
CogVideo [14]	txt-img & txt-video	Class.	160×160	626	-
Make-A-Video [24]	txt-img & txt-video	Class.	256×256	81.25	-
MagicVideo [33]	txt-img & txt-video	Class.	-	699	-
AID (1 Ref. frames)	txt-img & txt-video	Class.	256×256	102	16.5
CogVideo [14] (5 Ref. frames)	txt-img & txt-video	Class.	160×160	109.23	-
Seer [11] (5 Ref. frames)	txt-img	Class.	256×256	260.7	-
AID (5 Ref. frames)	txt-img & txt-video	Class.	256×256	**61.22**	**12.1**

Table 2. The comparisons of different fine-tune strategies of SVD [2] and Open-Sora [32]. We report the FVD in SSv2 datasets.

Methods	Strategy	Finetuned Parameters (↓)	FVD (↓)
Open-Sora [32]	Zero-Shot	-	903.37
Open-Sora [32]	Fully-Finetune	1147M	213.42
SVD [2]	Zero-Shot	-	592.14
SVD [2]	Fully-Finetune	1528M	163.78
SVD (Ours)	Adapter-Tuning	216M	50.23
Upper Baseline	VAE Reconstruction	-	7.14

method, which fixes the 3D UNet and only trains the newly added parameters, shows significantly lower GPU usage and faster training compared to full-finetuning methods. Additionally, in Table 4, we compare the training time, GPU consumption, and batch size of our method under the condition where GPU memory usage exceeds 90%. The default training approach allows for a larger batch size and shorter training time on a single card compared to full fine-tuning.

The Comparison of Model Generalization Capability To validate the generalization capability of our method and its performance in an open set, we present a qualitative comparison in Fig. 3. We use cartoon images generated by DALL-E [22] and real images captured by an iPhone as reference images to generate videos based on given instructions. We compared our method against Seer [11], as well as several state-of-the-art open-source [3, 32] and even commercial Image2Video methods [6, 16, 17]. It can be observed that our model demonstrates more precise instruction-following capabilities compared to other methods, highlighting the superior generalization ability of our approach. We also conducted experiments as shown in Table 5, comparing Seer, SVD, and our method. After training on Sthv2 [10], we evaluated their zero-shot performance

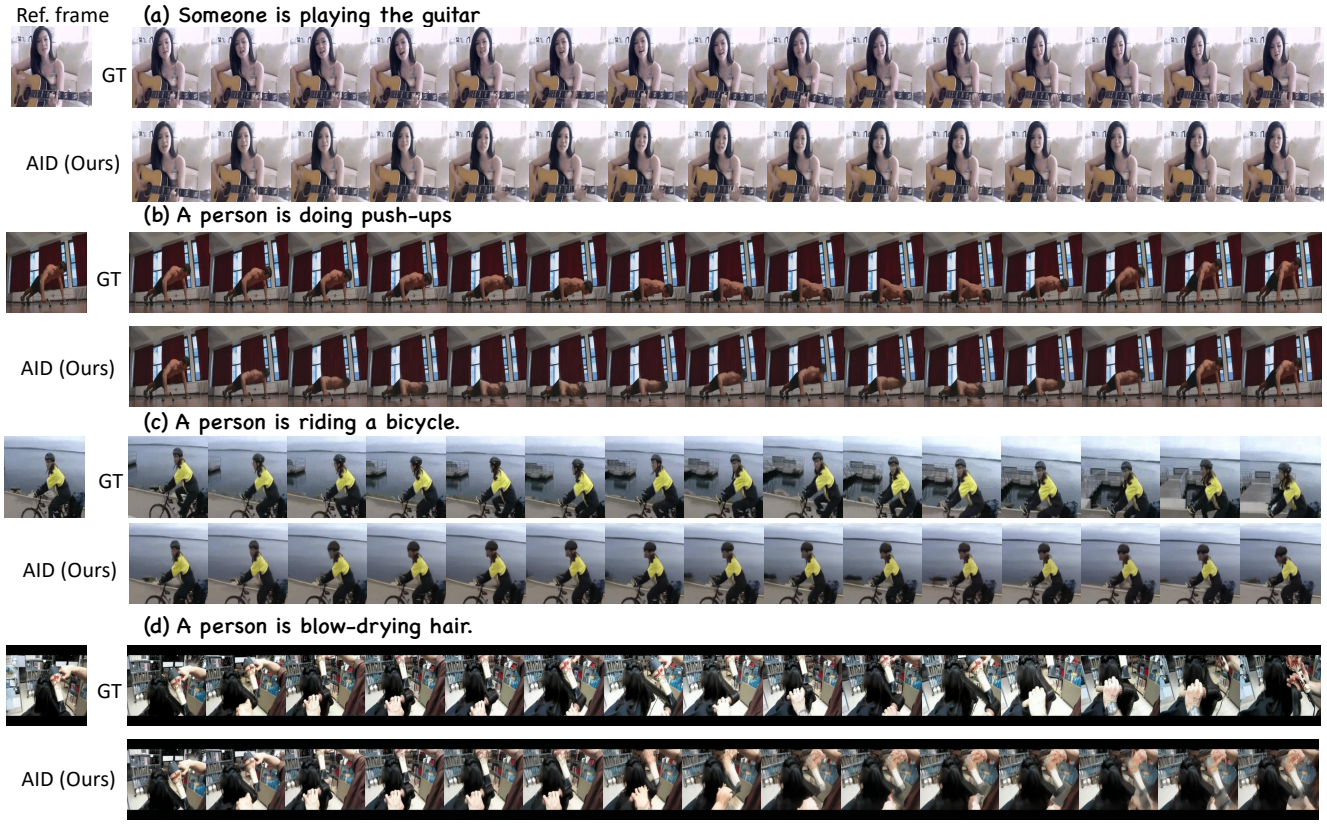


Figure 1. Visualization of text-conditioned video prediction on UCF-101 with 1 reference frame.

on Epic-Kitchen [5] using the FVD [28] metric. The results show that our method achieves the best zero-shot performance, which further demonstrates the generalizability of our approach.

The Comparison of the instruction-following capability To further quantitatively assess the instruction-following ability of our method, we compared its performance on Sthv2 against Seer, SVD, and GroundTruth. The metric CLIPSIM measures the CLIP similarity between the text instruction features and the visual features of video frames; higher values indicate better compliance with the given instructions. As shown in Table 6, our method outperforms both Seer and SVD and is close to GroundTruth.

Table 3. Training time (time.) and GPU memory (Mem.) consumption of the models (16-frame).

Model	time (s/iter.)	Mem. (GB)
Ours	1.03	24.68
Ours(full finetune)	1.45	45.49
Seer	0.75	24.97
VideoFusion	1.07	45.00

Table 4. Training time (time.) and GPU memory (Mem.) consumption of the models (16-frame, $\geq 90\%$ GPU memory usage).

Model	time (s/iter.)	Mem. (GB)	Batch
Ours	1.99	76.9	6
Ours(full finetune)	2.56	71.7	3
Seer	3.10	72.9	6
VideoFusion	7.68	78.7	3

A.3. Human Evaluation

In the main text, we provide a quantitative analysis of the experimental results. Additionally, we conducted human evaluation experiments. We randomly selected 50 samples from the Something-Something v2 dataset, and 25 samples each from

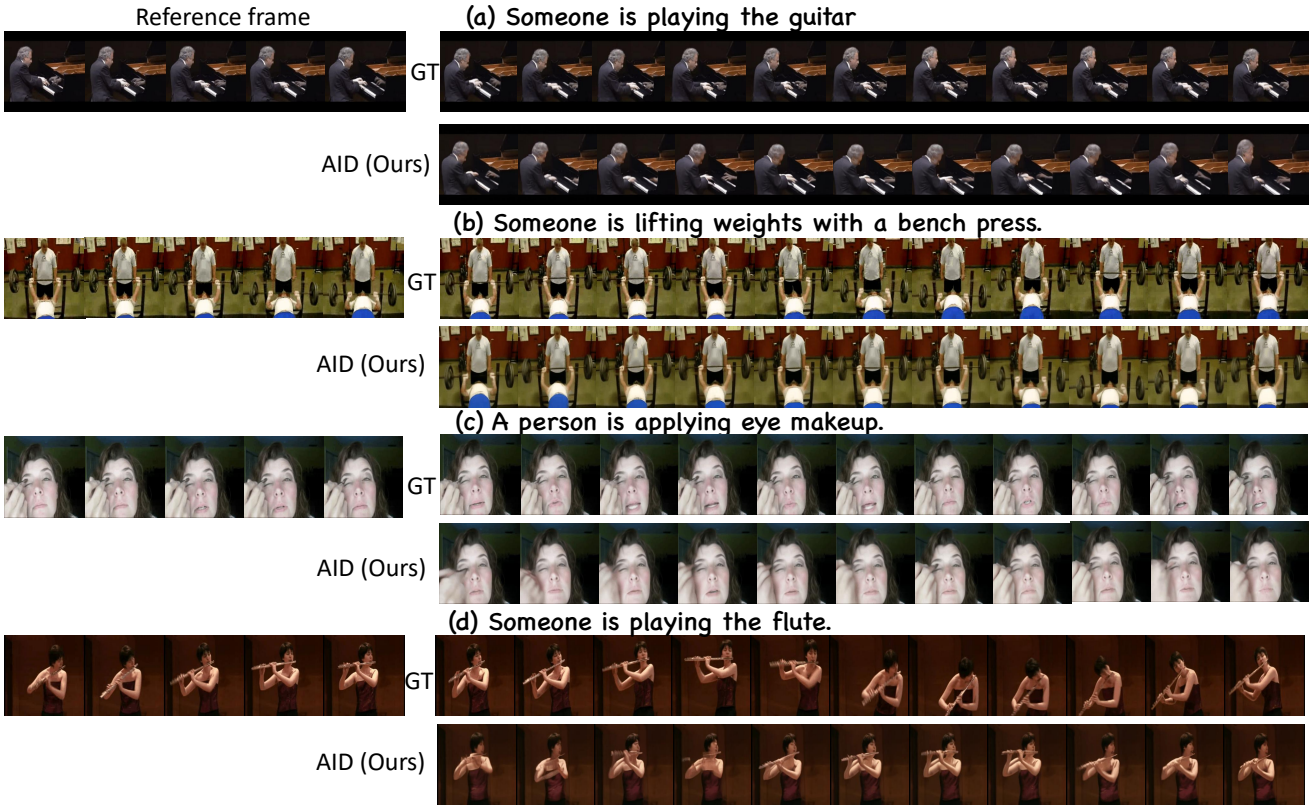


Figure 2. Visualization of text-conditioned video prediction on UCF-101 with 5 reference frames.

Table 5. The FVD results of Zero-shot setting.

	Seer	SVD	AID(ours)
Sthv2 (Training)	112.9	163.8	**50.2**
Epic100 (Zero-shot)	631.88	831.34	**487.2**

Table 6. The results of instruction following capability.

	Seer	SVD	AID(ours)	G.T.(upper)
FVD(↓)	112.9	163.8	**50.2**	-
CLIPScore(↑)	26.36	25.79	**26.98**	27.80

the BridgeData and Epic-Kitchen datasets, creating a total of 100 data pairs generated from Seer and AID. We invited 30 volunteers for anonymous selection, resulting in 3000 samples. As shown in Figure 12, we designed a questionnaire asking them to choose the video with higher quality, better alignment with the instruction and better frame consistency. The results indicate that our method significantly outperforms the previous state-of-the-art method, Seer [11], in terms of video quality and text alignment.

B. Broader Impact and Limitations

Broader Impact The generative models for text-guided video prediction have the potential to revolutionize media creation and utilization. When exploring their applications in tutorial video production and robotics, it is crucial to mitigate the risk of these models being used to generate misinformation or cause harm before they can be deployed in real-world scenarios. Additionally, a thorough examination of the models themselves, their intended uses, safety concerns, associated risks, and potential biases is essential before practical implementation.

Limitations Although our method shows significant improvements in Text-guided video prediction (TVP) tasks compared to previous approaches, it has fundamental limitations in synthesizing long videos. While we can progressively generate long videos using an iterative autoregressive method, this inference process remains quite costly. Additionally, our method is based on the SVD [2] image2video model, which lacks any textual descriptions and may not perform well on TVP tasks

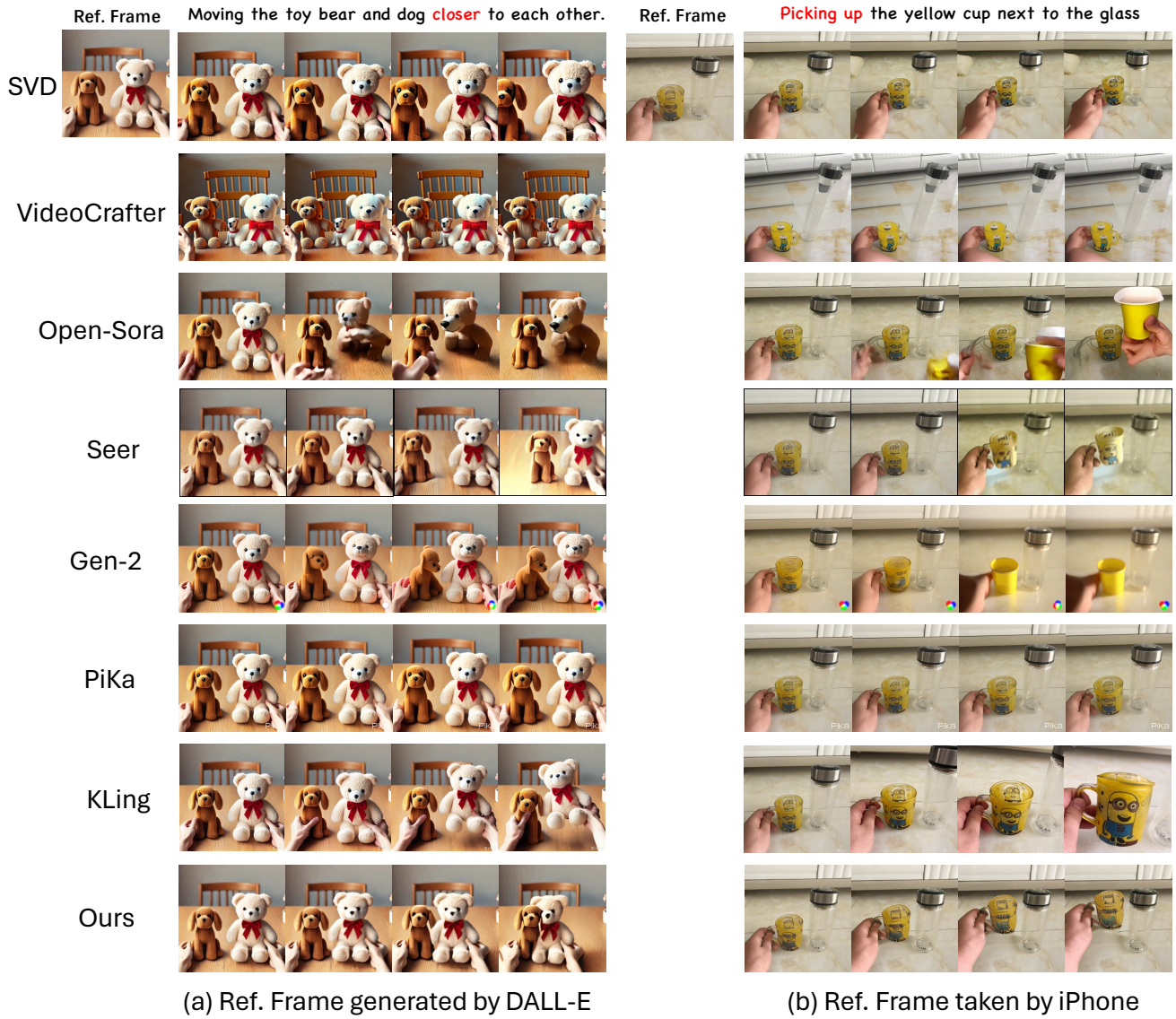


Figure 3. Qualitative comparison of video prediction with baselines on generated and real photoed reference image.

involving rare or novel protagonists. The optimal solution would be to pre-train our MLLM-aided method on a large-scale text-video dataset (*e.g.*, WebVid-10M [1], Panda-70M [4]) before transferring it to domain-specific TVP datasets.

Failure Case We also present the failure cases of our method in Figure 9. In case (a), the instruction is "Moving pen and marker closer to each other." Although our method successfully moves the pen and marker closer together, the left hand does not touch the pen, yet the pen moves, which violates physical laws. However, even the state-of-the-art Sora [20] generation model occasionally produces videos that violate real-world physics. Expanding the training dataset and increasing the number of training steps may help alleviate this issue. In case (b), the instruction is "Putting jar on a surface." Since there is no jar in the reference frame, our model successfully predicts the future motion but generates a jar that differs from the real scene. This failure case might be addressed by expanding the training dataset. As for cases (c) and (d), these are failure examples from the Epic-Kitchen [5] dataset. We find that the videos in this dataset have intense motions and involve fine-grained objects, such as "pick up knife" in case (d). For these challenging cases, the generation performance of our model deteriorates.

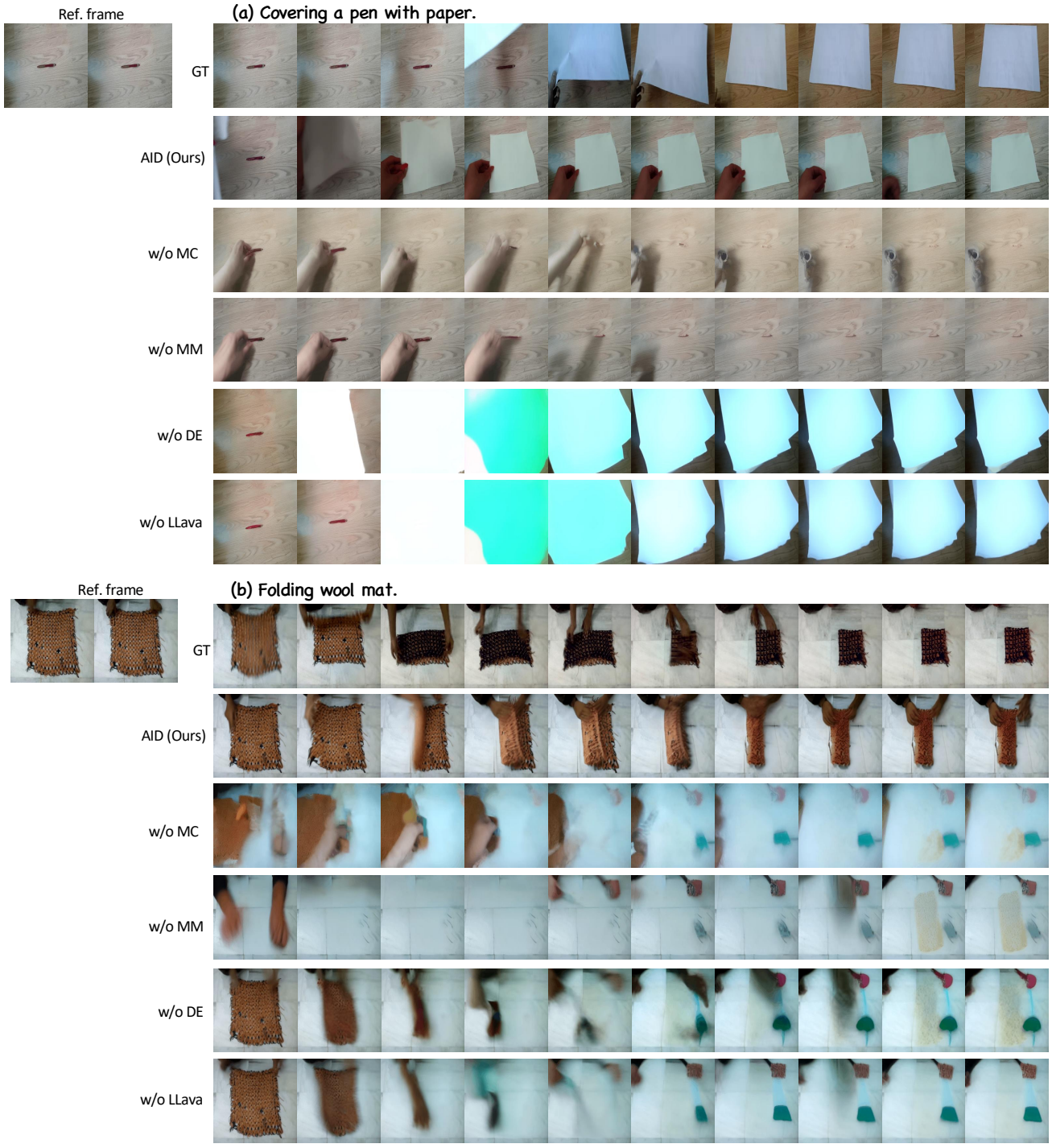


Figure 4. Ablation Study of different conditions on Something Something V2.

C. Implementation Details

C.1. Details of hyperparameters and fine-tuning

In this section, Table 7 provides an overview of the hyperparameters, fine-tuning details, sampling procedures, and hardware specifications of our AID model.

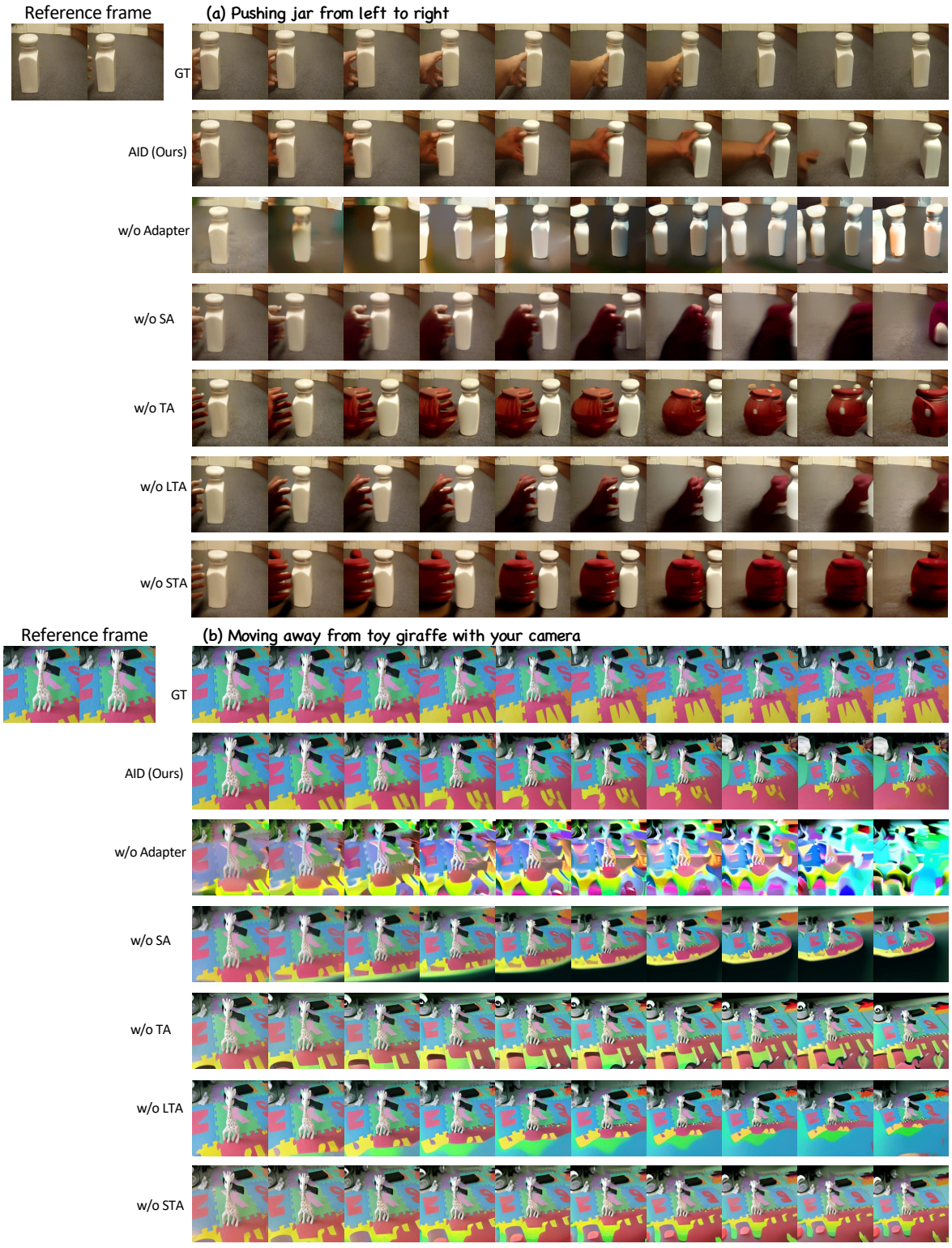


Figure 5. Ablation Study of different adapters on Something Something V2.

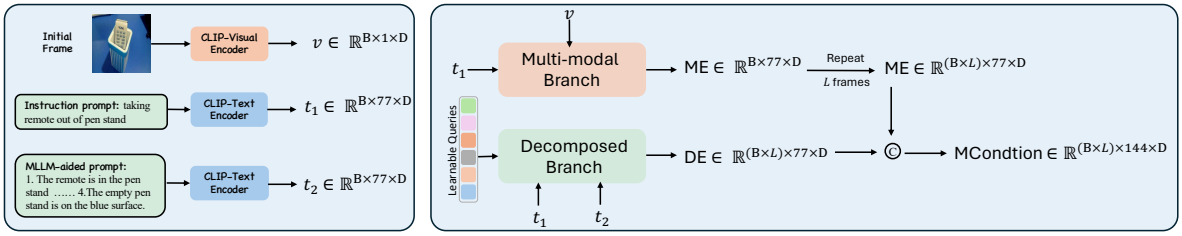


Figure 6. Illustration of the dimensionalities of each feature.

C.2. Details of DQFormer

In this section, we show the hyperparameters of DQFormer in Table 8. In addition, we present the specific dimensions of each feature in the DQFormer architecture in Fig. 6.

Table 7. Hyperparameters and details of Fine Tuning/Inference.

Param.	Value
optim.	AdamW
Adam-beta1	0.9
Adam-beta2	0.999
Adam-epsilon	$1e^{-8}$
weight decay	$1e^{-2}$
lr	$5e^{-5}$
lr_scheduler	constant
train batch size	8/GPU
resolution	256×256
train. steps	100k
train. hardware	4 NVIDIA-A100
sampling steps	30
text guidance scale	12
visual guidance scale	1.5

Table 8. Hyperparameters of setting of DQFormer.

HyperParam.	Value
learnable tokens channels	1024
output channels	1024
base channels	1024
Number of layers	2
Number of atten. heads	8
Dimension of cross-atten.	1024
Number of query length	77

D. Additional Visualization

In this section, we provide additional visualizations of AID. The results on the Epic-Kitchen dataset are shown in Figure 11, and the results on the Bridge dataset are presented in Figure 10. We also demonstrate the long video prediction and text-guided video manipulation examples in Figure 13 and Figure 8. In cases (2)-(4), we use the last two frames of the previous clip as the initial frames for the next clip, allowing iterative extension into longer videos. Other cases demonstrate that given the same reference frame, different instructions can predict different future video frames. Finally, in Figure 14, we provide examples of the prompts and feedback of MLLM model. Although we provide many sample figures in this paper, we recommend readers visit the website "`website/index.html`" in the supplementary materials to see the demos in video style.

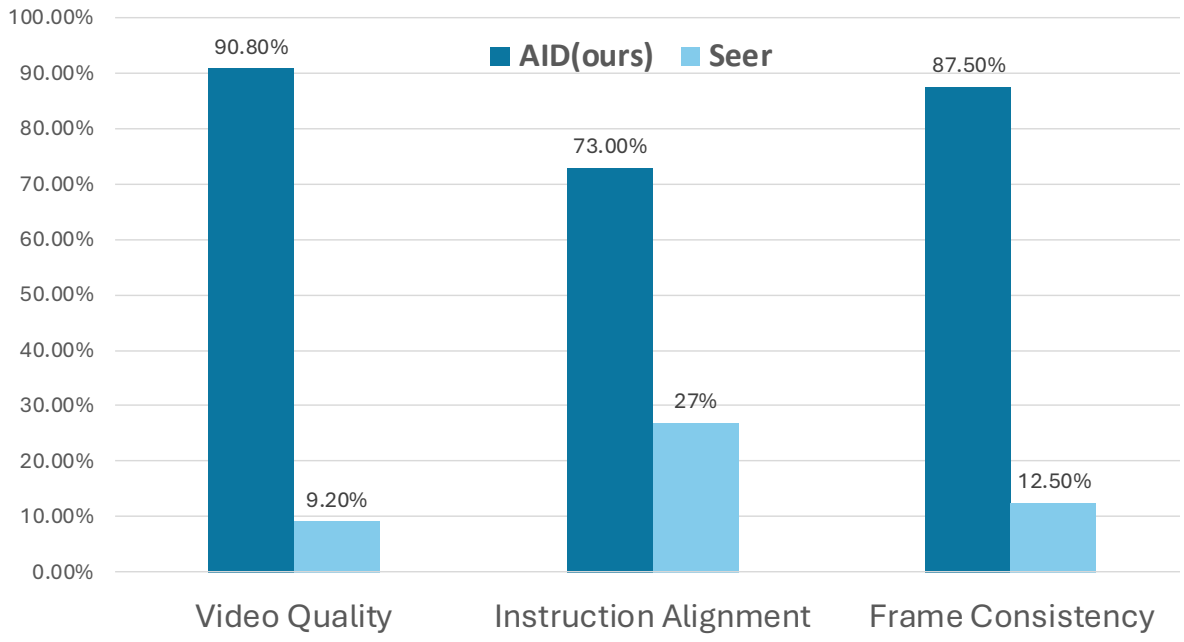


Figure 7. Preference percentage of human evaluation results.

Figure 8. Examples of long video prediction with different instruction.

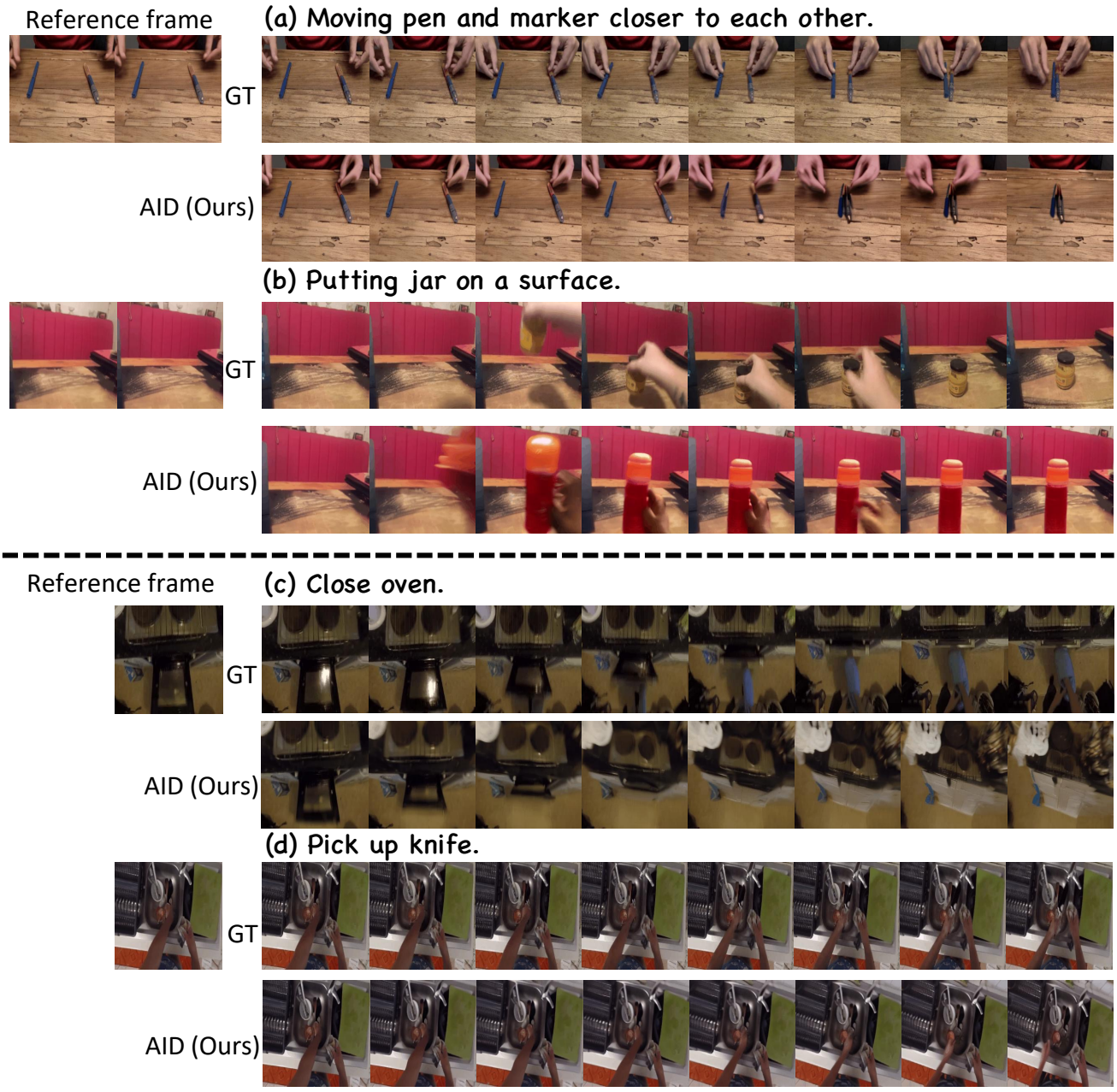


Figure 9. Failure cases of AID on Something Something-V2 and Epic-kitchen.

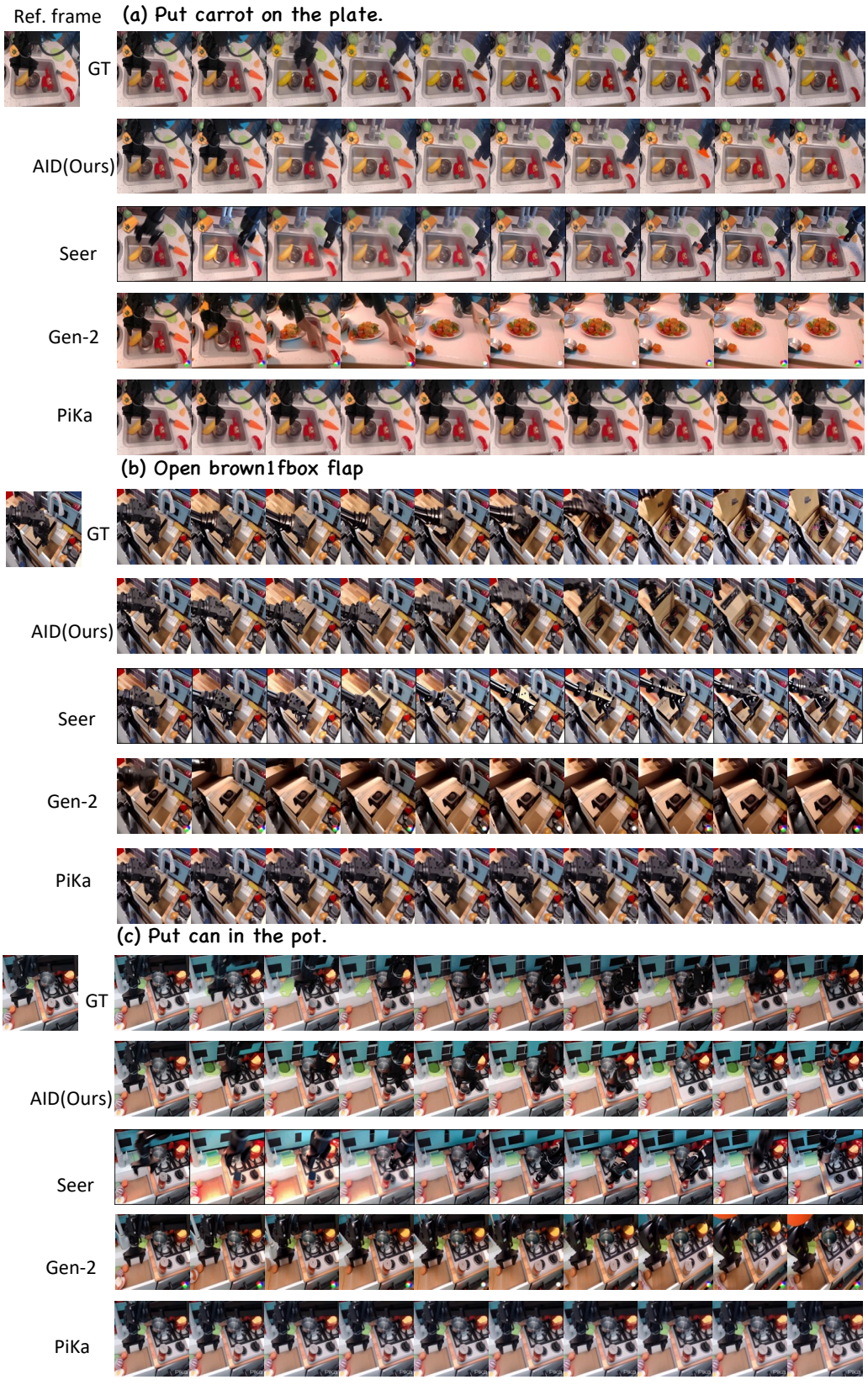


Figure 10. Visualization of text-conditioned video prediction on Bridge with 1 reference frame compared to Seer [11], Gen-2 [6], PiKa [17].

Figure 11. Visualization of text-conditioned video prediction on Epic-kitchen with 1 reference frame compared to Seer [11].

Text-conditioned Video Prediction

Please review the text prompt and reference frame. Based on the description, **choose the video that has higher quality and more coherence**, or **select the one that more closely matches the text description from the two candidates**.

Reference Frame:

*1. According to the following text prompt, the video that needs to be generated is: "push jar from left to right". Please note that the given reference video may not match the language description exactly, but your task is to choose a video that accurately reflects the given language description.

B

A

*2. which video has higher quality and more coherence?

B

A

Figure 12. Screenshot of a questionnaire example shown to human evaluators.

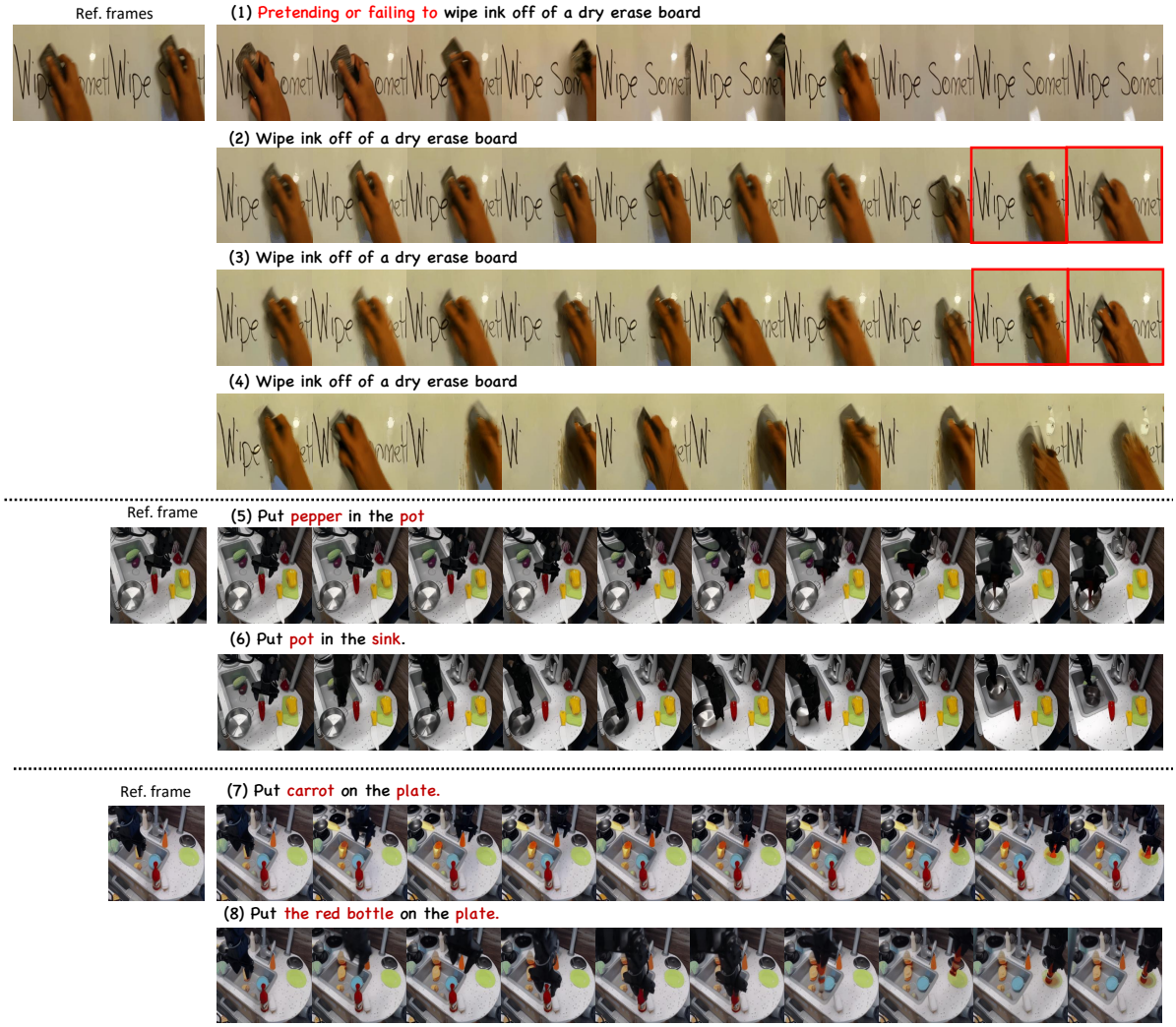
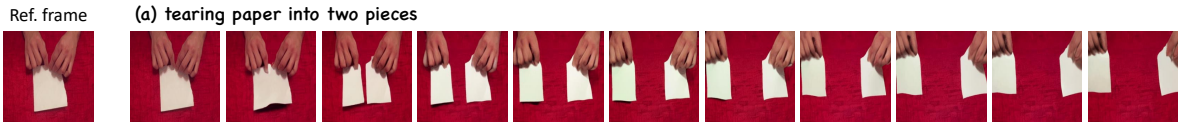


Figure 13. Examples of long video prediction and instruction-based video manipulation are provided.

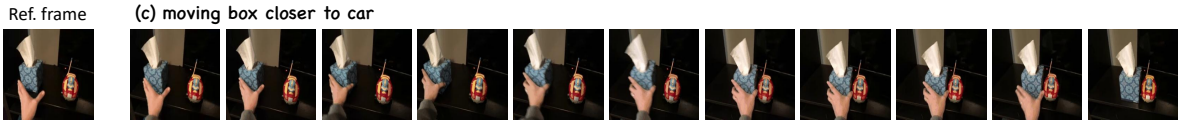
User Prompt:

Input: I have a prompt "[video prompt]" for video generation. This prompt guides the process of generating a video from the given <image>. Can you describe this process in separate states? Each state should be described in only one sentence, and please ensure the descriptions between states are coherent. Be straightforward and avoid a narrative style. Additionally, can you align the description with the provided image? There should be four states in total. Output:



1. The person is holding a piece of paper.
2. The person is tearing the paper into two pieces.
3. The person has torn the paper into two pieces.
4. The two pieces of paper are on the red carpet.

1. The tennis ball is held in a hand.
2. The hand moves the tennis ball closer to the opening of the mug.
3. The tennis ball is aligned above the mug as if to be dropped in.
4. The hand pulls the tennis ball away from the mug.



1. The box is initially positioned away from the car.
2. The box is being moved closer to the car.
3. The box is now in close proximity to the car.
4. The box is now positioned next to the car.

Figure 14. Examples of the input and feedback of the Multi-modal Large Language Model.

References

[1] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, pages 1728–1738, 2021. 5

[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. 1, 2, 4

[3] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter1: Open diffusion models for high-quality video generation, 2023. 2

[4] Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, et al. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. *arXiv preprint arXiv:2402.19479*, 2024. 5

[5] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, 2018. 3, 5

[6] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. *arXiv preprint arXiv:2302.03011*, 2023. 2, 11

[7] Songwei Ge, Thomas Hayes, Harry Yang, Xi Yin, Guan Pang, David Jacobs, Jia-Bin Huang, and Devi Parikh. Long video generation with time-agnostic vqgan and time-sensitive transformer. In *European Conference on Computer Vision*, pages 102–118. Springer, 2022. 2

[8] Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve your own correlation: A noise prior for video diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22930–22941, 2023. 1, 2

[9] Cade Gordon and Natalie Parde. Latent neural differential equations for video generation. In *NeurIPS 2020 Workshop on Pre-registration in Machine Learning*, pages 73–86. PMLR, 2021. 2

[10] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *ICCV*, 2017. 1, 2

[11] Xianfan Gu, Chuan Wen, Weirui Ye, Jiaming Song, and Yang Gao. Seer: Language instructed video prediction with latent diffusion models. In *ICLR*, 2024. 1, 2, 4, 11, 12

[12] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity video generation with arbitrary lengths. *arXiv preprint arXiv:2211.13221*, 2022. 1, 2

[13] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022. 1, 2

[14] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022. 1, 2

[15] Emmanuel Kahembwe and Subramanian Ramamoorthy. Lower dimensional kernels for video discriminators. *Neural Networks*, 132: 506–520, 2020. 2

[16] Kuaishou. Kling, 2024. 2

[17] Pika Lab. Pika, 2023. 2, 11

[18] Haoyu Lu, Guoxing Yang, Nanyi Fei, Yuqi Huo, Zhiwu Lu, Ping Luo, and Mingyu Ding. Vdt: General-purpose video diffusion transformers via mask modeling. In *The Twelfth International Conference on Learning Representations*, 2023. 2

[19] Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan. Videofusion: Decomposed diffusion models for high-quality video generation. In *CVPR*, 2023. 2

[20] OpenAI. Sora, 2024. 5

[21] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *CVPR*, pages 11410–11420, 2022. 1

[22] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2

[23] Masaki Saito, Shunta Saito, Masanori Koyama, and Sosuke Kobayashi. Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal gan. *International Journal of Computer Vision*, 128(10):2586–2606, 2020. 2

[24] 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. *ICLR*, 2023. 1, 2

[25] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 1

[26] Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N Metaxas, and Sergey Tulyakov. A good image generator is what you need for high-resolution video synthesis. In *ICLR*, 2021. 2

[27] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018. 2

[28] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 1, 3

[29] Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. *NeurIPS*, 2022. 2

[30] Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, et al. Magvit: Masked generative video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10459–10469, 2023. 2

[31] Sihyun Yu, Jihoon Tack, Sangwoo Mo, Hyunsu Kim, Junho Kim, Jung-Woo Ha, and Jinwoo Shin. Generating videos with dynamics-aware implicit generative adversarial networks. In *ICLR*, 2022. 2

[32] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all, 2024. 1, 2

[33] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022. 1, 2