Show Me: Unifying Instructional Image and Video Generation with Diffusion Models

Supplementary Material

A. Data Curation

A.1. Something-Something V2

The Something-Something v2 (SSv2) dataset is a comprehensive collection of labeled video clips illustrating human hand gestures and interactions with objects. It contains a diverse range of actions such as "Putting something into something," "Turning something," and "Pushing something from left to right." The original dataset comprises approximately 220,847 video clips, each around 4 seconds long, spanning 174 action categories. Compared to Epic100, SSv2 videos clearly exhibit an initial state and generally conclude with a frame depicting action completion. However, we identified several action categories without distinct human actions or meaningful object state changes, such as "holding," "showing," "fail," and "nothing happens," along with potentially harmful actions like "hitting" and "throwing." To ensure dataset relevance and safety, we manually reviewed all 174 action templates, identified and filtered out 62 categories containing keywords unlikely to reflect meaningful or useful changes. As a result, we curated the dataset to include 112,321 training samples and 2,048 test samples.

A.2. EPIC-KITCHENS 100

The EPIC-KITCHENS (Epic100) dataset comprises approximately 90,000 egocentric action segments with 20,000 unique narrations, covering 97 verb classes and 300 noun classes, recorded from daily kitchen activities. However, the continuous and unscripted nature of the recordings leads to challenges in generating coherent instructional videos. Specifically, egocentric videos often include highly dynamic actions and abrupt scene transitions, as shown in Fig. 9, and the end frames of narrations may not precisely align with the corresponding video clips. To address these challenges, we introduce a dual-similarity filtering approach to identify and remove videos with significant scene changes or misaligned content.

Our dual-similarity strategy involves two metrics:

- **Semantic Similarity**: We compute the similarity between text narrations and candidate frames using CLIP embeddings to ensure semantic alignment.
- **Visual Consistency**: We compute a framewise DINO similarity score between the initial frame and each of the downsampled last 8 frames.

By multiplying these two similarity scores, we measure the in-context semantic relevance and visual coherence of target frames. The frame with the highest combined similarity score is selected as the candidate target frame if its score exceeds a predefined threshold (We empirically set it to 0.1). Using this filtering strategy, we refined the dataset to 57,602 training samples and 8,236 test samples.

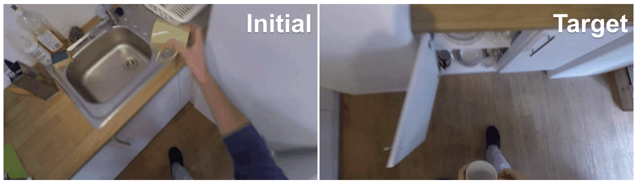


Figure 9. Inconsistent visual context in egocentric instructional videos. From left to right: initial and target frames.

B. Ego4D Evaluation

To validate the effectiveness of our method, we followed LEGO's benchmark [34] setting and conducted additional experiments on Ego4D for instructional image generation. As shown in Table 5, our method achieves competitive results compared to LEGO. Importantly, LEGO fully finetunes Stable Diffusion with MLLM-enriched text instructions, whereas we only apply LoRA tuning with the vanilla action instructions provided in Ego4D. We further extend it for instructional video generation, where ShowMe consistently improves over the baseline and achieves the best FVD among all compared methods, as illustrated in Table 6.

Methods	EgoVLP ↑	EgoVLP$^+$ ↑	CLIP-I ↑	FID ↓	PSNR ↑	LPIPS ↓	BLIP-B ↑	BLIP-L ↑
ProxEdit	44.51	72.68	68.17	33.01	11.88	40.90	17.73	17.35
SDEdit	50.07	72.90	73.35	33.35	11.81	41.60	19.80	19.74
IP2P	62.19	78.84	78.75	24.73	12.16	37.16	20.00	20.56
LEGO	65.65	**80.44**	**80.61**	23.83	**12.29**	36.43	**20.38**	20.70
ShowMe	**66.93**	79.14	79.77	**19.38**	11.92	**35.59**	19.76	**22.28**

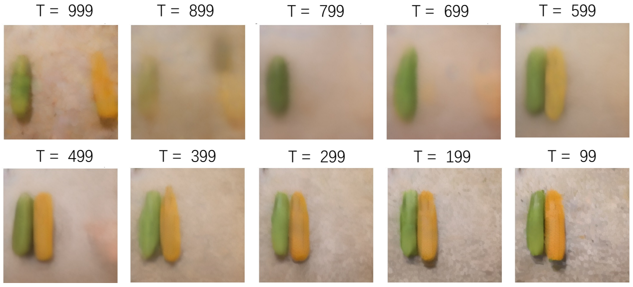
Table 5. Comparison of instructional image generation on Ego4D.

Methods	FVD ↓	FID ↓	CLIP-I ↑	CLIP-T ↑	EgoVLP ↑	Motion ↑
AnimateAnything	271.14	13.24	**0.8805**	0.2541	0.3123	14.80
ConsistI2V	81.60	**8.72**	0.8768	0.2585	**0.3628**	74.61
DynamiCrafter	105.76	16.20	0.8401	0.2601	0.3064	96.69
ShowMe	**72.51**	12.00	0.8609	**0.2613**	0.3196	**96.75**

Table 6. Comparison of instructional video generation on Ego4D.

C. More Implementation Details

For action-object state manipulation and prediction, we set the LoRA rank to 128 and 64, respectively, with a dropout rate of 0.1. For structure reward tuning, we empirically set the diffusion threshold Γ for the one-step approximation to 200, and to 500 for motion reward tuning, as evidenced



Action Instruction: *"moving corn and cucumber closer to each other"*

Figure 10. Visualization of one-step denoising results at different diffusion steps.

in Fig. 10. The weights for $\mathcal{L}_{\text{dp}}$ and $\mathcal{L}_{\text{ed}}$ are both set to 1, while the weight for $\mathcal{L}_{\text{mo}}$ is set to 0.001. During reward tuning, these loss functions work together with the noise prediction loss to optimize the model.

D. Reward Computation Cost

Table 7 shows that the dominant cost in Stage-1 is the grad-enabled VAE decode, while the rewards themselves add only modest increments. In Stage-2, the motion reward operates entirely in latent space, incurring only a moderate computational cost. In practice, the observed training throughput reduction is acceptable given the gains in alignment and motion quality.

Stage	Component	ΔVRAM Peak	Time (ms)	ΔGFLOPs
Stage 1	VAE decode (grad)	+1.92 GB	187	–
	+ Edge reward	+0.19 GB	368	810
	+ Depth reward	+0.14 GB	1022	641
Stage 2	+ Motion reward	+0.79 GB	453	1559

Table 7. Complexity of reward tuning (per sample at 256^2, fp16).

E. Rationale for One-Step Approximation

We visualize one-step denoising samples at different diffusion steps during training, as shown in Fig. 10. It can be observed that at lower noise levels, the approximation closely resembles the target image and preserves spatial details well. In contrast, at higher noise levels (above 600), although structural information is lost, the approximated samples still capture the overall motion trends. This behavior aligns with the sampling process during inference: the model typically plans the global layout and motion patterns in the early high-noise steps, while refining spatial and structural details in the later denoising stages.

F. Limitations and Future Work

Finally, we present several failure cases in Figure 11 to illustrate the limitations of our method. In the first and third examples, when a scene contains multiple similar or complex objects, the model struggles to ground the instruction precisely. For instance, distinguishing between "coffee beans" and the "grinder", which leads to task failure. In the second example, although the action is correct, the model's limited understanding of 3D space causes the plastic cup to appear split in half. In the final example, the model hallucinates a blank sheet of paper, indicating a failure to recognize the paper already being held. One possible solution is to introduce spatial grounding models for accurate target localization. In addition, explicitly conditioning on segmentation and depth information, along with using MLLMs to generate more detailed instructions, could further improve performance.

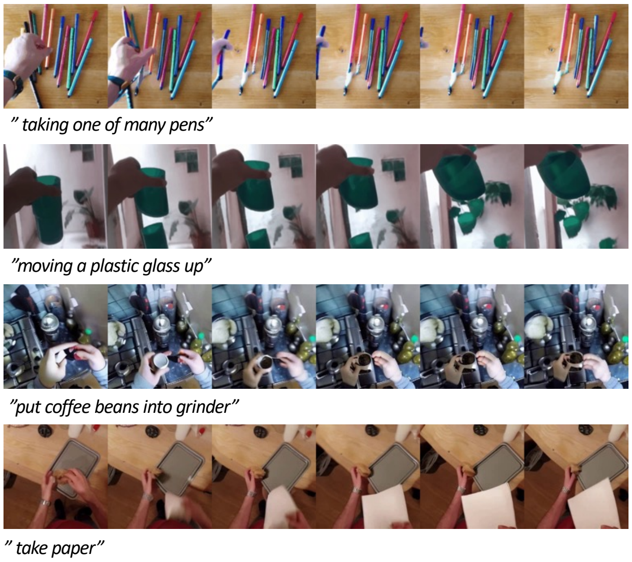

" taking one of many pens"
"moving a plastic glass up"
"put coffee beans into grinder"
" take paper"

Figure 11. Failure examples. The first two rows are from SSv2 and the last two rows are from Epic100.

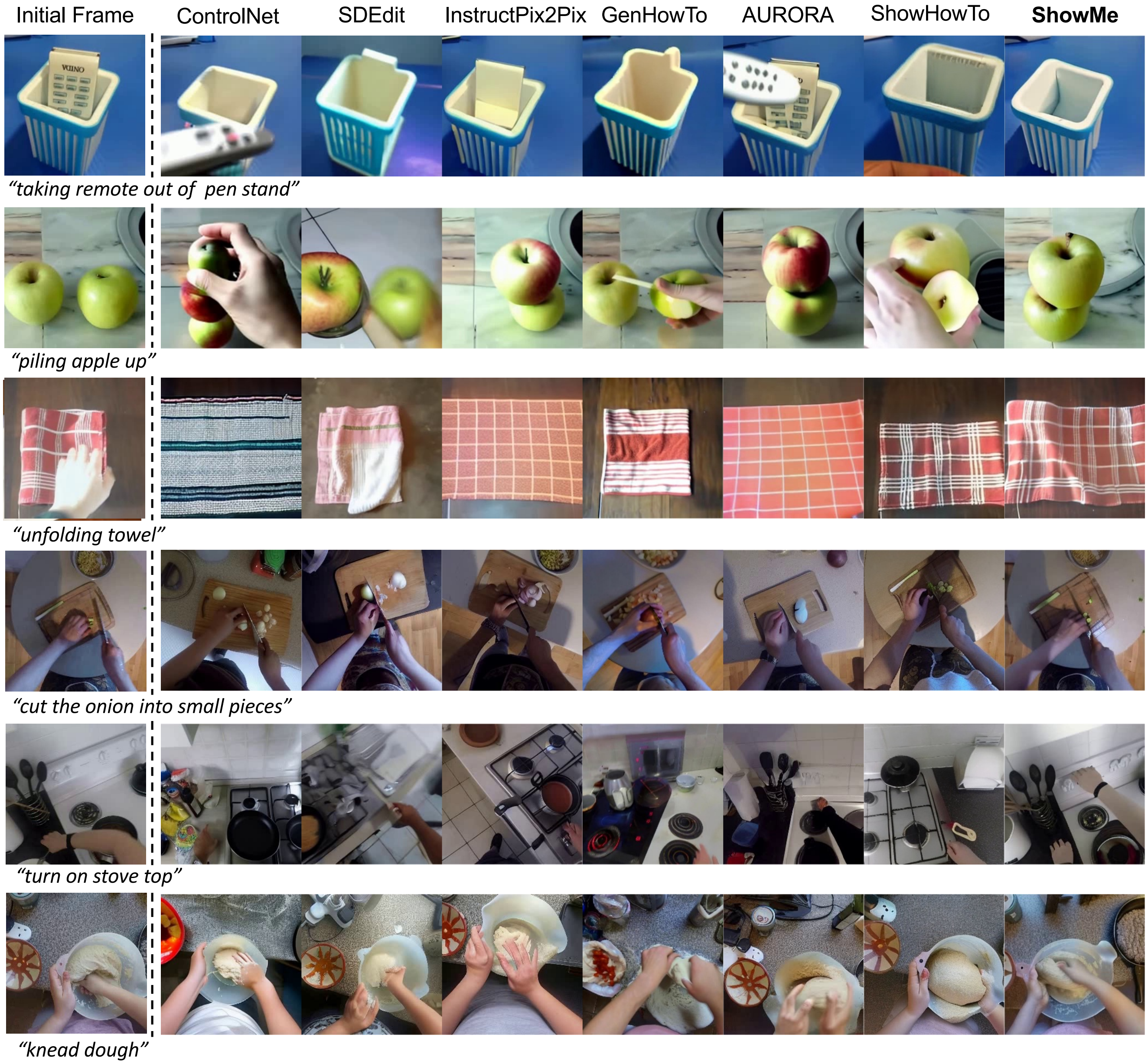


Figure 12. Comparison of different methods for instructional image generation. The first three rows are test samples from SSv2, and the last three rows are from Epic100. Our method is better at completing action instructions and maintaining contextual consistency.

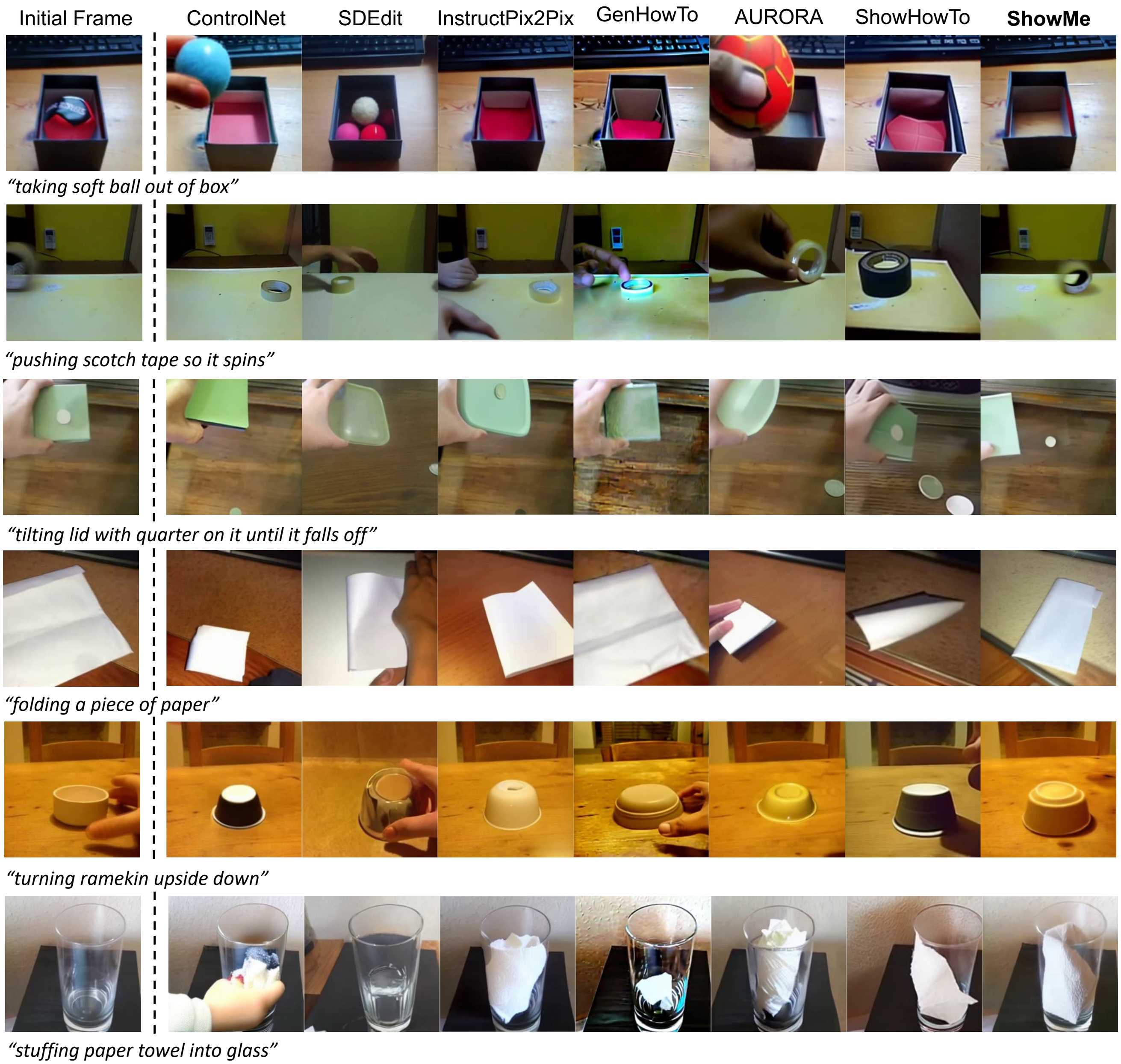
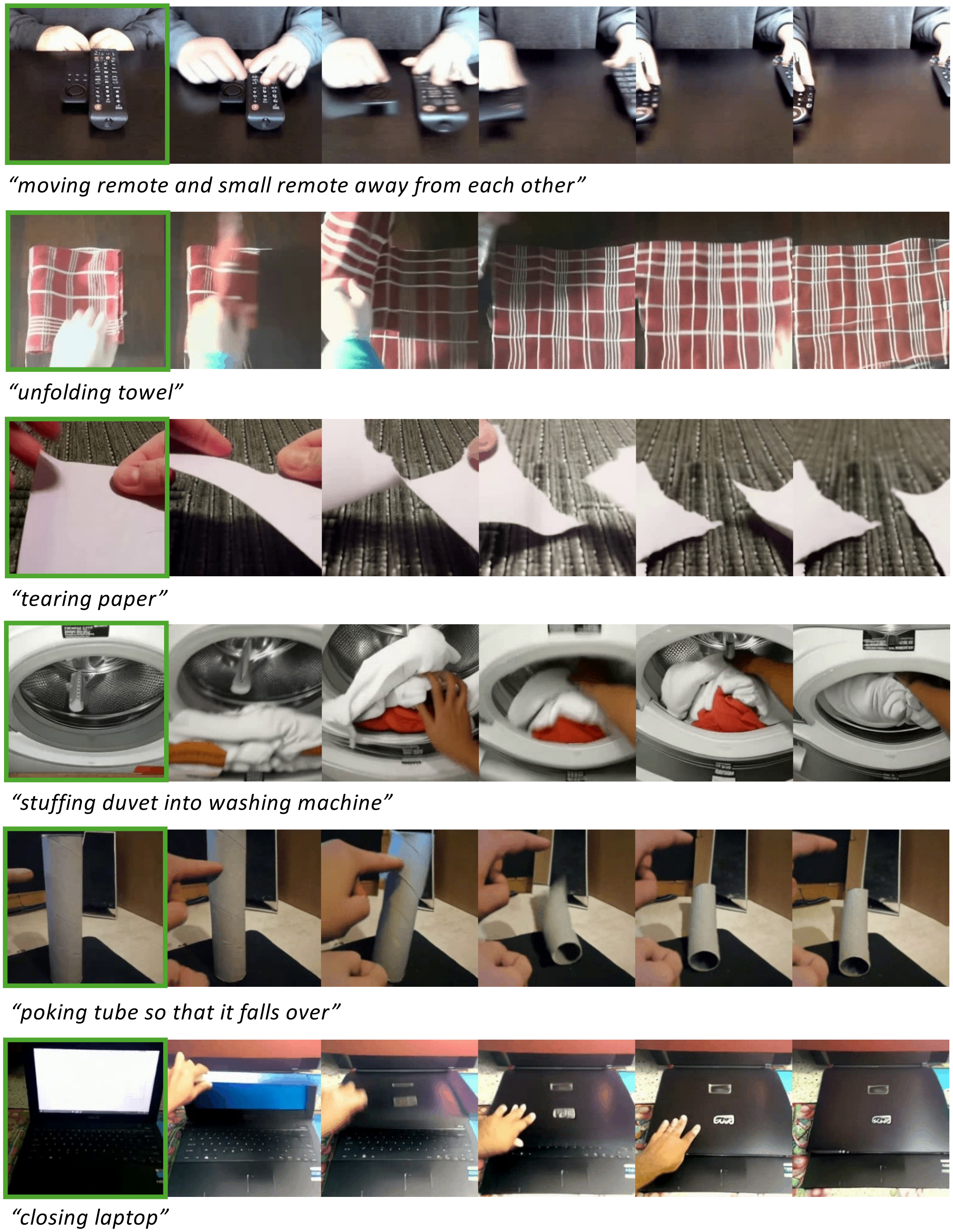

Initial Frame
ControlNet
SDEdit
InstructPix2Pix
GenHowTo
AURORA
ShowHowTo
ShowMe
"taking soft ball out of box"
"pushing scotch tape so it spins"
"tilting lid with quarter on it until it falls off"
"folding a piece of paper"
"turning ramekin upside down"
"stuffing paper towel into glass"

Figure 13. More visualization results of instructional image generation on SSv2.



“moving remote and small remote away from each other”

“unfolding towel”

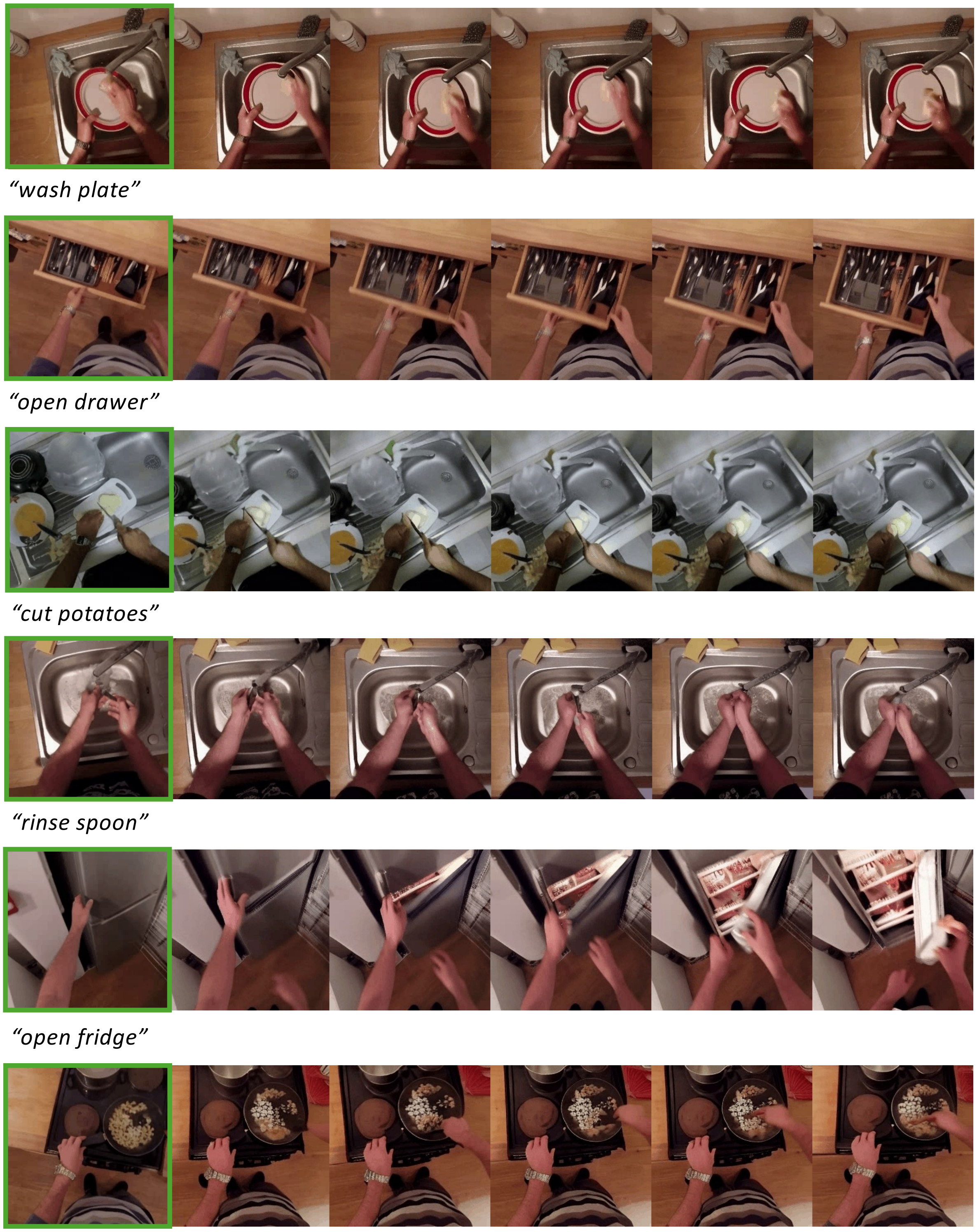
“tearing paper”

“stuffing duvet into washing machine”

“poking tube so that it falls over”

“closing laptop”

Figure 14. Generated videos on SSv2 dataset.



“wash plate”

“open drawer”

“cut potatoes”

“rinse spoon”

“open fridge”

“stir food”

Figure 15. Generated videos on Epic100 dataset.