Table 5. **More baselines in our evaluation set and DragBench.** IF and MD are computed with source images / target points, while other metrics use target images.

Method	Our Evaluation Dataset					DragBench	
	CLIP-FID (↓)	LPIPS (↓)	SSIM (↑)	IF (↑)	MD (↓)	IF (↑)	MD (↓)
Readout Guidance	11.592	0.271	0.734	0.736	54.12	0.790	52.22
SDE-Drag	9.923	0.209	0.789	0.828	47.96	0.921	45.78
DiffEditor	9.364	0.196	0.793	0.802	32.71	0.856	28.58
DragDiffusion	9.192	0.187	0.811	0.807	35.83	0.881	33.16
LightningDrag	9.894	0.214	0.794	0.798	22.31	0.885	18.62
FramePainter	**8.513**	**0.166**	**0.825**	**0.834**	**19.52**	**0.925**	**16.37**

Table 6. **FramePainter (Ours) vs. Frame2Frame.**

Method	CLIP-FID (↓)	LPIPS (↓)	SSIM (↑)
Frame2Frame	9.738	0.202	0.756
Ours	**7.783**	**0.140**	**0.859**

Table 7. **Ablation on SVD and Stable Diffusion v2.1.**

Base Model	CLIP-FID (↓)	LPIPS (↓)	SSIM (↑)
Stable Diffusion v2.1	9.497	0.193	0.795
SVD(Ours)	**7.783**	**0.140**	**0.859**

A. Implementation Details of Different Visual Editing Instructions.

By default, the visual editing instructions (*e.g.*, sketch images and coarsely edited images) are directly encoded using sparse control encoder and injected into the denoising U-Net. However, it is challenging to encode images that only contain source and target points, which cannot accurately represent the correspondence between each pair of points. Since this paper aims to explore a general paradigm for interactive learning, rather than focusing on the specific editing method of dragging points, we adopt a simple and intuitive way to encode dragging points. Specifically, at the output of each attention block, we directly copy the source image tokens corresponding to the positions of source points, and add them to the edited image tokens at the positions of target points. As a result, this simple approach allows for an accurate understanding of dragging points and enables plausible editing of input images, *e.g.*, in Fig. 4 and Fig. 14.

B. More Visualizations and Comparisons.

Fig. 11 show more visualizations on sketch images. Fig. 12 and Fig. 13 provide more comparisons with alternative approaches on sketch images and coarsely edited images, respectively. Fig. 14 compares a wide range of drag-based methods, including encoder-based (*i.e.*, LightningDrag [52]) and optimization-based (*i.e.*, DragDiffusion [51], SDE-Drag [40], Readout Guidance [27], and DiffEditor [36]). Compared to the baselines, FramePainter presents superior performance in understanding the dragging points and maintaining the structural integrity of objects. In contrast, due to the absence of real-world dynamic priors, optimization-based methods struggle with moving object parts, *e.g.*, duplicated tail in row 2 of Fig. 14 (top) and duplicate hair in row 2 of Fig. 14 (bottom). Encoder-based method cannot preserve the overall structure of objects, *e.g.*, separated mushrooms in row 1 of Fig. 14 (top).

Comparison with Frame2Frame. Our method differs in two key aspects: *(i)* F2F performs text-guided editing, while we focus on interactive visual instructions (*e.g.*, sketch), enabling more intuitive and controllable edits; *(ii)* F2F uses VDM to directly produce entire video clips, whereas we finetune VDM to generate only two frames for efficient image editing. We follow F2F to annotate temporal captions of our test set with GPT-4o. Table 6 and Fig. 9 shows that our method achieves more accurate edited results and better generalization to out-of-domain cases, *e.g.*, transform a clownfish into shark-like shape.

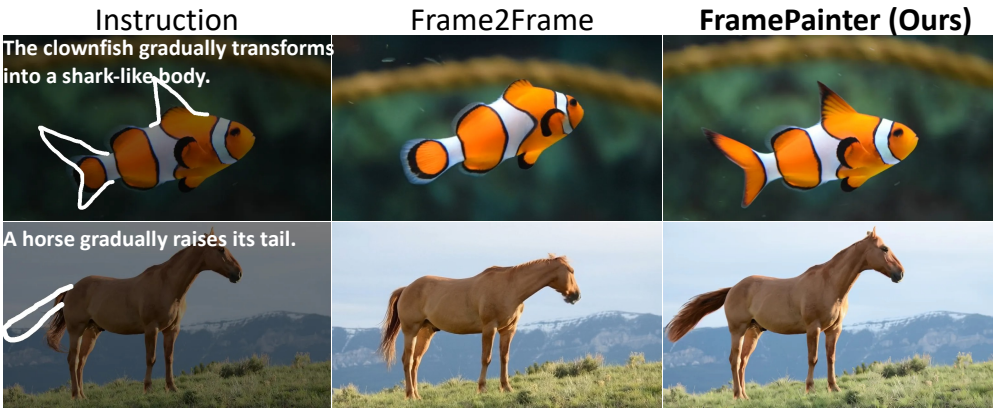

Figure 9. **FramePainter (sketch condition) vs. Frame2Frame (text condition).**

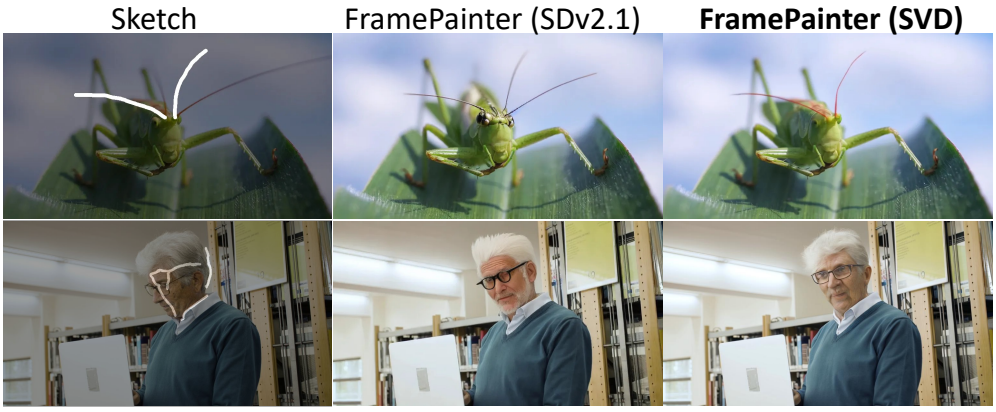

Figure 10. **Ablation on the use of SVD and Stable Diffusion v2.1 (image counterpart of SVD).**

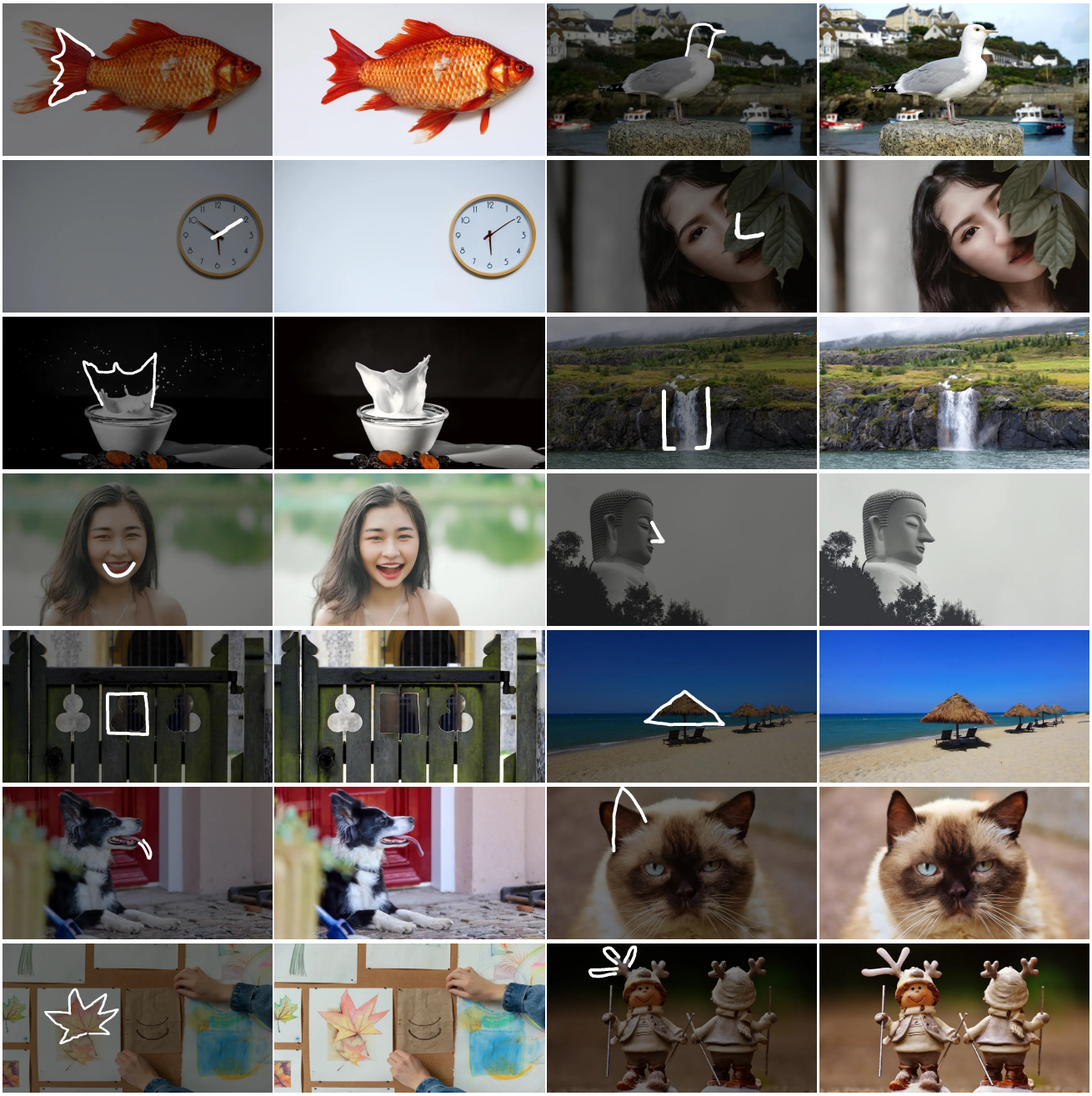

Figure 11. **More visualization examples of FramePainter.** This figure presents both a wide range of scenarios, including in-domain (*e.g.*, change the position of cat ear) and out-of-domain cases (*e.g.*, enlarge the dear horn in hat).

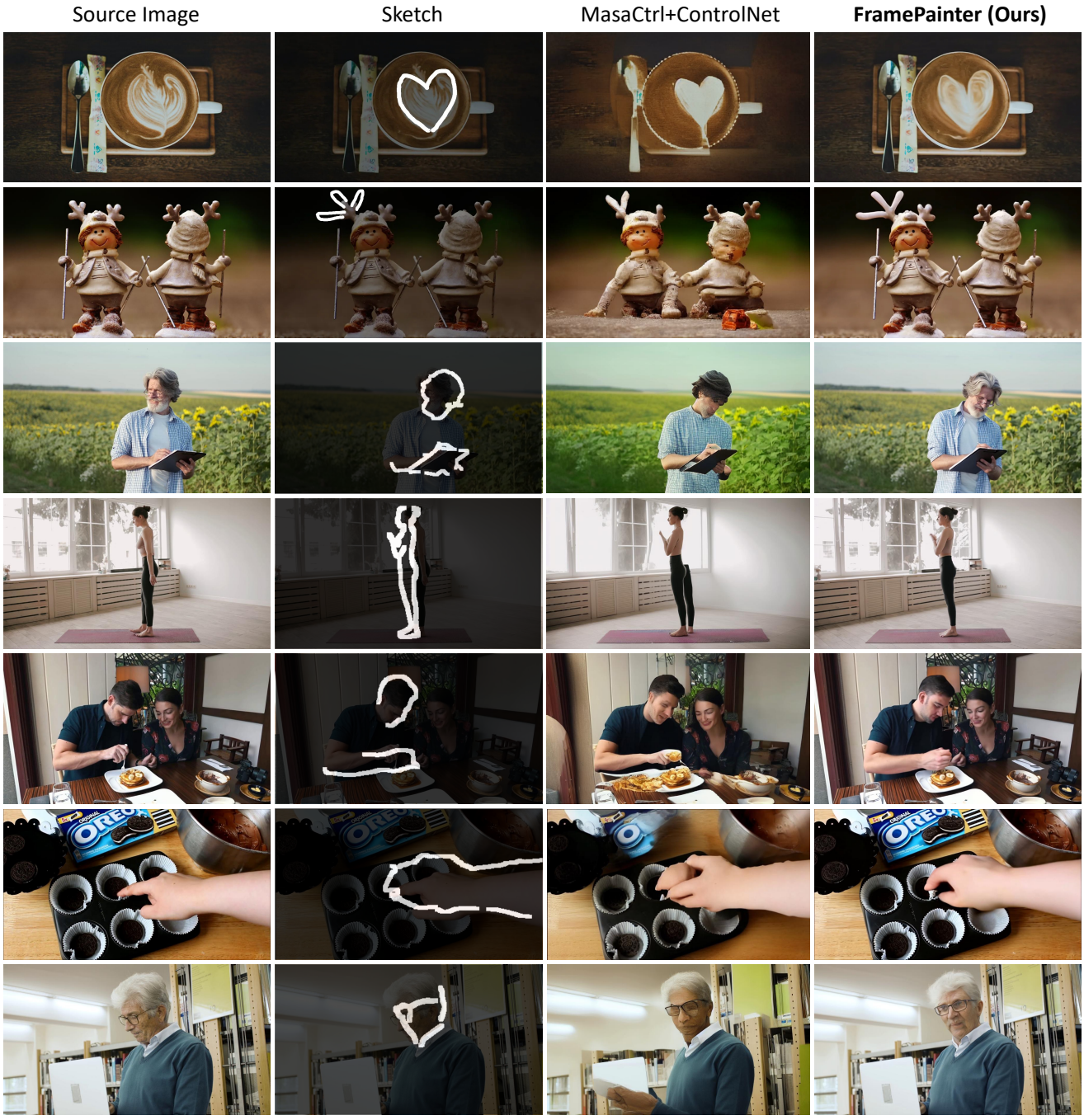

Figure 12. **More qualitative comparisons in sketch images.**

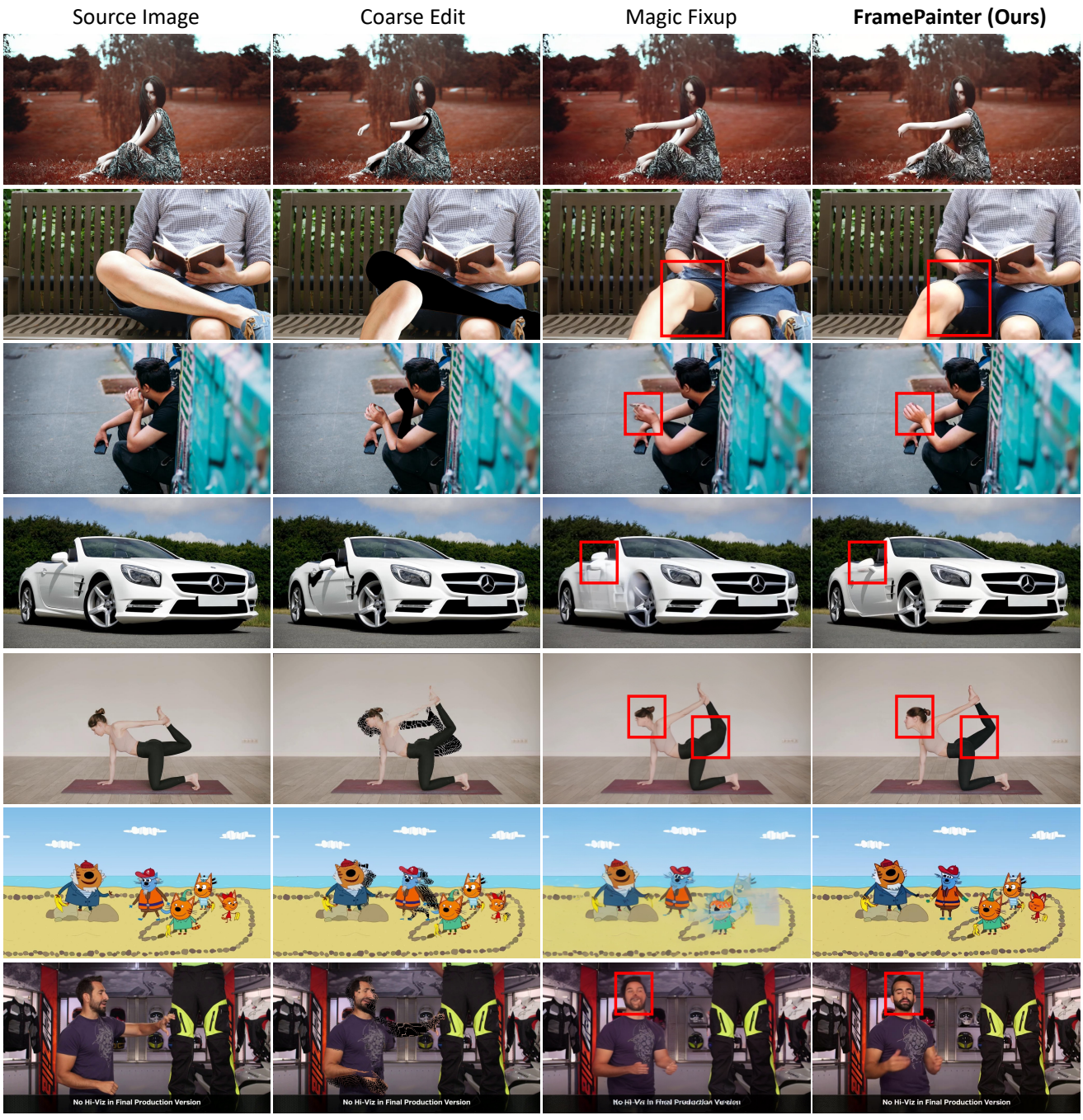

Figure 13. **More qualitative comparisons in coarsely edited images.**

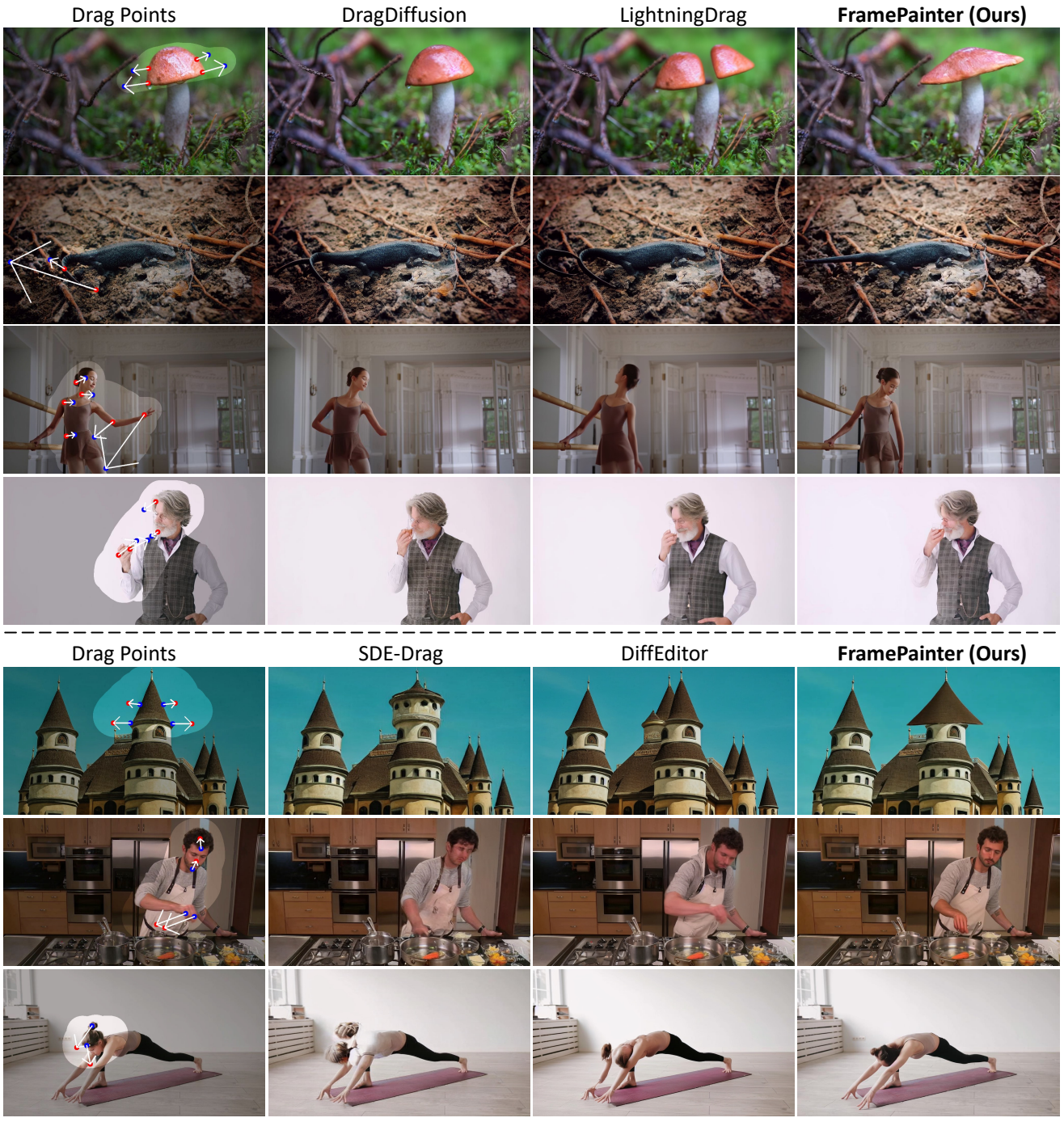

Figure 14. **More qualitative comparisons in dragging points.** We compare FramePainter with both encoder-based (*i.e.*, LightningDrag) and optimization-based methods (*i.e.*, DragDiffusion, SDE-Drag, and DiffEditor). During inference, DragDiffusion and SDE-Drag require to finetune additional LoRAs to preserve the visual appearance of source images.