VIVA: VLM-Guided Instruction-Based Video Editing with Reward Optimization

Supplementary Material

1. Preliminaries

1.1. DiT-Based Video Generation and Editing

Diffusion Transformers (DiTs) have become a powerful backbone for modern video generation. They model a sequence of latent video tokens through a transformer-based denoising process conditioned on multimodal inputs such as text or reference frames. DiTs are commonly trained with the *Flow Matching* objective [5], which learns a velocity field $\mathbf{v}_\theta(\mathbf{x}, t)$ that transports a simple prior $p_0(\mathbf{x})$ (*e.g.*, Gaussian noise) toward a target data distribution $p_1(\mathbf{x})$ via a probability flow Ordinary Differential Equation (ODE):

$$\frac{d\mathbf{x}_t}{dt} = \mathbf{v}_\theta(\mathbf{x}_t, t), \quad \text{with} \quad \mathbf{x}_0 \sim p_0(\mathbf{x}). \tag{1}$$

In the commonly used *Rectified Flow* formulation [7], the trajectory is defined as a linear interpolation between noise and data samples:

$$\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\mathbf{x}_1, \quad \mathbf{x}_1 \sim p_1(\mathbf{x}), \tag{2}$$

and the model is trained to regress the target velocity field $\mathbf{u}_t(\mathbf{x}_t) = \frac{d\mathbf{x}_t}{dt} = \mathbf{x}_1 - \mathbf{x}_0$ via the loss:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{t,\mathbf{x}_0,\mathbf{x}_1}\left[\|\mathbf{v}_\theta(\mathbf{x}_t, t) - (\mathbf{x}_1 - \mathbf{x}_0)\|_2^2\right]. \tag{3}$$

This objective learns a continuous flow that transports noisy latents toward clean samples, providing smoother temporal dynamics and faster convergence than conventional noise-prediction objectives.

For video editing, the DiT backbone is conditioned on the source video and an editing instruction, enabling it to synthesize instruction-aligned output based on the input video.

1.2. Group Relative Policy Optimization (GRPO)

Group Relative Policy Optimization (GRPO) [10] is a recent reinforcement learning algorithm that extends Proximal Policy Optimization (PPO) [9] for efficient post-training alignment. For each input query c, a group of G sampled trajectories $\{\mathbf{x}_t^i\}_{i=1,\cdots,G}^{t=0,\cdots,T}$ are generated and evaluated by the task-specific reward criterion to get the corresponding reward signals $\{R(\mathbf{x}_0^i)\}_{i=0}^G$. These reward signals are then transformed to relative advantages within the group by:

$$\hat{A}_t^i = \frac{R(\mathbf{x}_0^i) - \text{mean}(\{R(\mathbf{x}_0^i)\}_{i=1}^G)}{\text{std}(\{R(\mathbf{x}_0^i)\}_{i=1}^G)}. \tag{4}$$

This relative formulation reduces variance and stabilizes optimization, making GRPO effective for aligning large generative models with human or semantic preferences.

The policy model is then optimized by maximizing the following objective without requiring a separate critic network:

$$\begin{aligned}\mathcal{L}(\theta) &= \mathbb{E}_{c,\{\mathbf{x}^i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|c)} f(r, \hat{A}, \theta, \epsilon, \beta) \\ &= \frac{1}{G}\sum_{i=1}^{G}\frac{1}{T}\sum_{t=0}^{T-1}(\mathcal{L}_{policy}(\theta) - \beta\mathcal{L}_{KL}(\theta)),\end{aligned} \tag{5}$$

where

$$\begin{aligned}\mathcal{L}_{policy}(\theta) &= \min(r_t^i(\theta)\hat{A}_t^i, \text{clip}(r_t^i(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t^i), \\ \mathcal{L}_{KL}(\theta) &= D_{\text{KL}}(\pi_\theta||\pi_{\text{ref}}), \\ r_t^i(\theta) &= \frac{p_\theta(\mathbf{x}_{t-1}^i \mid \mathbf{x}_t^i, \mathbf{c})}{p_{\theta_{\text{old}}}(\mathbf{x}_{t-1}^i \mid \mathbf{x}_t^i, \mathbf{c})}.\end{aligned} \tag{6}$$

Since GRPO relies on stochastic sampling diverse trajectories $\{\mathbf{x}_t^i\}_{i=1,\cdots,G}^{t=0,\cdots,T}$, Flow-GRPO [6] introduce randomness into Flow Matching by converting the deterministic Flow-ODE into an equivalent Flow-SDE

$$dx_t = \left[v_\theta(\mathbf{x}_t, t) + \frac{\sigma_t^2}{2t}\left(\mathbf{x}_t + (1-t)\hat{v}_\theta(\mathbf{x}_t, t)\right)\right]dt + \sigma_t\sqrt{dt}\boldsymbol{\epsilon} \tag{7}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$ is a newly sampled gaussian noise, $\sigma_t = \eta\sqrt{\frac{t}{1-t}}$ for Flow-GRPO.

Coefficients-Preserving Sampling [11] investigates the problem of an excess of noise injected during Flow-SDE sampling in Eq. 7 and reformulates the sampling process to eliminate the noise artifacts by modifying $\sigma_t = \sin(\frac{\eta\pi}{2})dt$.

2. Implementation Details

During Edit-GRPO, We insert Low-Rank Adaptation (LoRA) [3] modules into the self-attention and cross-attention layers of the DiT [4]. We freeze the parameters of the model after supervised fine-tuning. Only the LoRA parameters are updated, facilitating efficient optimization. For the LoRA configuration, we set the low-rank dimension to $r = 64$ and the scaling factor to $\alpha = 128$. The adapter weights are initialized using a Gaussian distribution.

During inference, we use a classifier-free guidance scale of 2.0, applying it only to instruction conditions. The inference timestep is set to 50 for the balance of performance and inference speed.

3. Data Strategy

3.1. Detailed Architecture for Data Preparation

Having briefly outlined the pipeline for constructing editing pairs in the main text, we now provide the detailed network

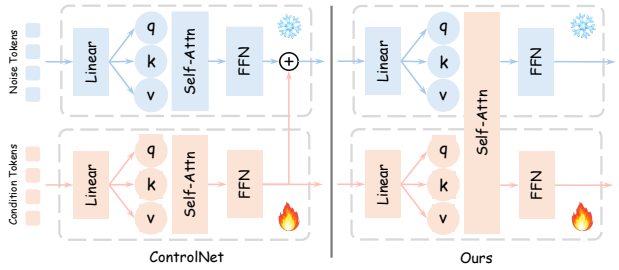

Figure 1. Network architecture for paired data synthesis. We modify the pretrained MMDiT backbone by inserting an additional control branch every four blocks. By concatenating the condition tokens with the noisy latent, the model performs full mutual attention across the sequence, ensuring robust structural alignment between the control signal and the generated video.

architecture used for data synthesis. As illustrated in Figure 1, we integrate an additional branch into the MMDiT architecture [1] to model the interaction between the VAE embeddings of the conditions (e.g., masked video or scribbles) and the noisy latent. The condition tokens and the noisy latent tokens are concatenated, enabling full mutual attention across the entire combined sequence. Furthermore, we align the Rotary Positional Embeddings (RoPE) of the condition tokens with those of the noisy latent tokens. This design is based on the assumption that spatially and temporally corresponding positions in the condition inputs and the target videos are highly correlated and should thus exhibit the highest attention values.

To reduce computational complexity, we insert this additional branch block once every four blocks within the pretrained DiT. During training, we only update the parameters of this additional branch while keeping the pretrained DiT weights frozen to preserve the generative capabilities of the foundation model.

Empirically, we observe that the standard ControlNet architecture is highly sensitive to the Classifier-Free Guidance (CFG) scale. It often generates oversaturated samples at high CFG values, while suffering from poor instruction following at low CFG values. In contrast, our model remains free from saturation artifacts across a wide range of CFG values. We attribute this improvement to the bi-directional attention mechanism between the conditional input and the noisy latent enabled by our branch. Figure 2 visualizes representative training samples spanning addition, removal, replacement, and stylization tasks.

3.2. Edit-GRPO Data

For the Edit-GRPO training stage, only the source videos $\mathbf{V}_{src}$ are required. We curate a subset of high-quality videos and employ Gemini 2.5 Pro [2] to generate suitable editing instructions $\mathbf{t}_{ins}$ for each source video. These prompts are specifically designed to encompass complex combinations of various editing types. We leverage Gemini 2.5 Pro to caption the source video to get the source video prompt $\mathbf{t}_{src}$. Subsequently, conditioned on $\mathbf{t}_{src}$ and $\mathbf{t}_{ins}$, Gemini 2.5 Pro generates the target video caption $\mathbf{t}_{edit}$. This pipeline yields the triplet $(\mathbf{V}_{src}, \mathbf{t}_{src}, \mathbf{t}_{edit})$ required for training Edit-GRPO.

4. Experiments

4.1. Qualitative Comparisons

Figure 3 presents qualitative comparisons of the reference-based video editing on the VIE-Bench [8]. Since existing open-source instruction-based video editing models do not support reference-image control, we compare our model only with Runway Gen-4 Aleph. As shown in Figure 3, although Runway responds to the instruction, it fails to accurately align with the reference image. For example, the toy car and red backpack show noticeable deviations from the reference in the edited results. In contrast, our model correctly understands the intricate relationships among the instruction, the reference image, and the source video, and produces precise editing outcomes.

4.2. Ablation Study on Edit-GRPO

Figure 5 presents qualitative comparisons on before and after applying Edit-GRPO. Edit-GRPO yields a substantial qualitative improvement, consistent with quantitative ablation results in the main paper. In challenging prompts such as “Add a woman with long black hair Lying in the bucket of a green agricultural loader.” and “Replace the man into a gorilla.”, Edit-GRPO preserves the structural integrity and plausibility of the subjects, whereas removing it tends to produce distorted artifacts and collapsing results. For instances like “Add a dinosaur standing on the ground of gas station.” and “Add a big white kitten in the car trunk.”, Edit-GRPO makes the edited areas more natural and realistic, showing exceptional consistency with the source video. Overall, the model equipped with Edit-GRPO demonstrates stronger instruction adherence, higher aesthetic fidelity, more consistent with the source video, and better alignment with human preference.

4.3. Complex Tasks

Figures 6, 7, and 8 present more results of our method on complex instructions that are non-trivial and challenging to be synthesized by the data construction pipeline. Our model is capable of performing complex hybrid instructions like “Make the mountain in the background an active volcano erupting with smoke and lava, and add a little cat running by the couple.”, and non-rigid visual effects such as flames, smokes, fireworks, stylization, and watermark removal. Notably, our model effectively addresses a challenging case in global background editing: ensuring physical coherence

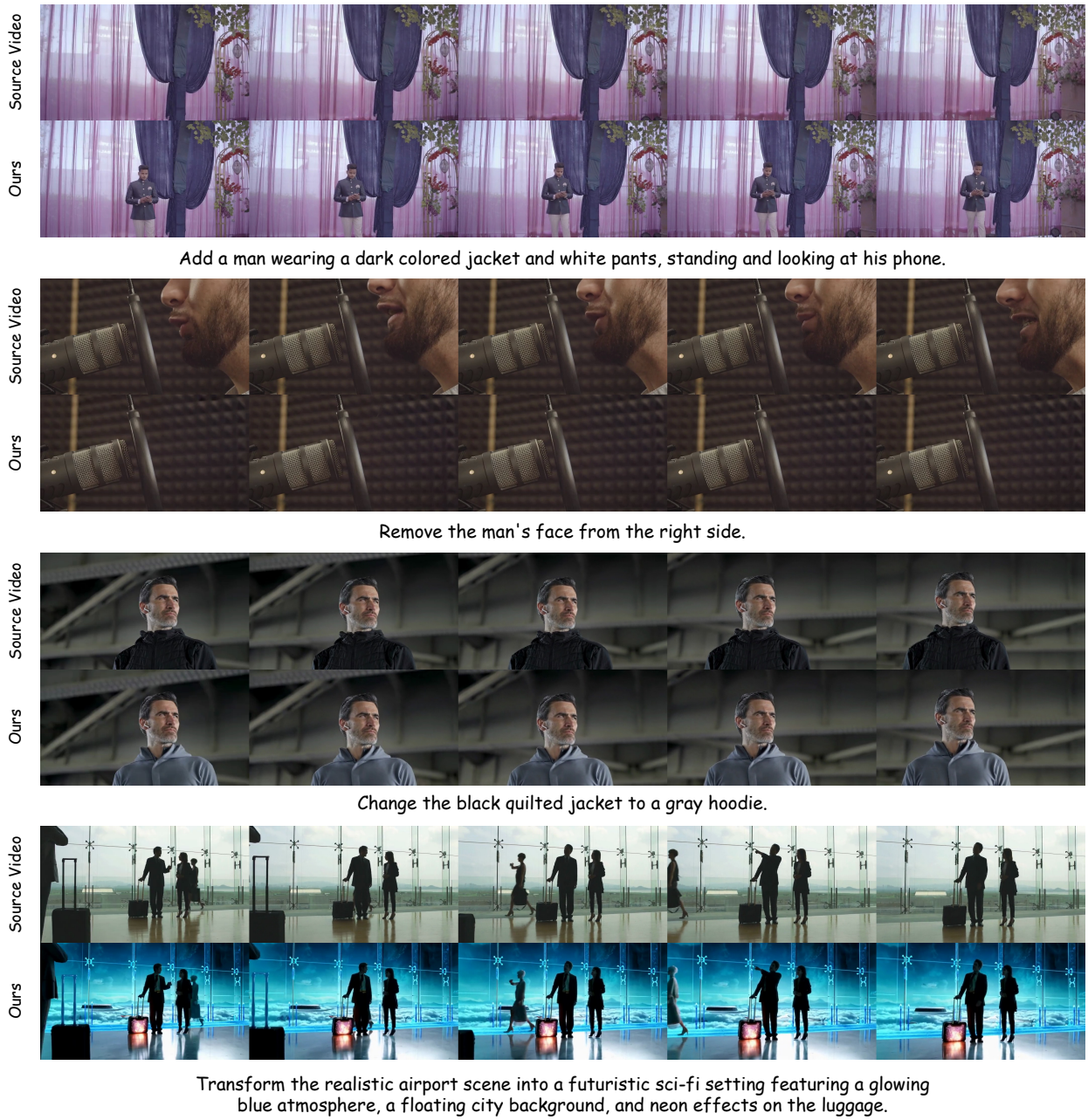

Figure 2. We visualize representative training samples spanning addition, removal, replacement, and stylization tasks, arranged from top to bottom.

between the foreground subject and the edited new environment. This is vividly demonstrated in the example "Change the background from the ocean to a vast, snowy mountain range." In this scenario, our model successfully synthesizes realistic shadows cast by the subject onto the snowy terrain, thereby achieving a high degree of photorealism and subject-background harmony. Since these complex editing scenarios are difficult to synthesize through the data construction pipeline, they serve as a rigorous test of a model's generalization capabilities. We attribute our model's success to two key factors: First, our VLM instructor possesses an exceptional capacity to interpret the intricate correlation between textual instructions and visual contexts. Second, our strategy of mixing image editing data during training effectively transfers the robust generalization inherent in the image domain to the video editing domain.

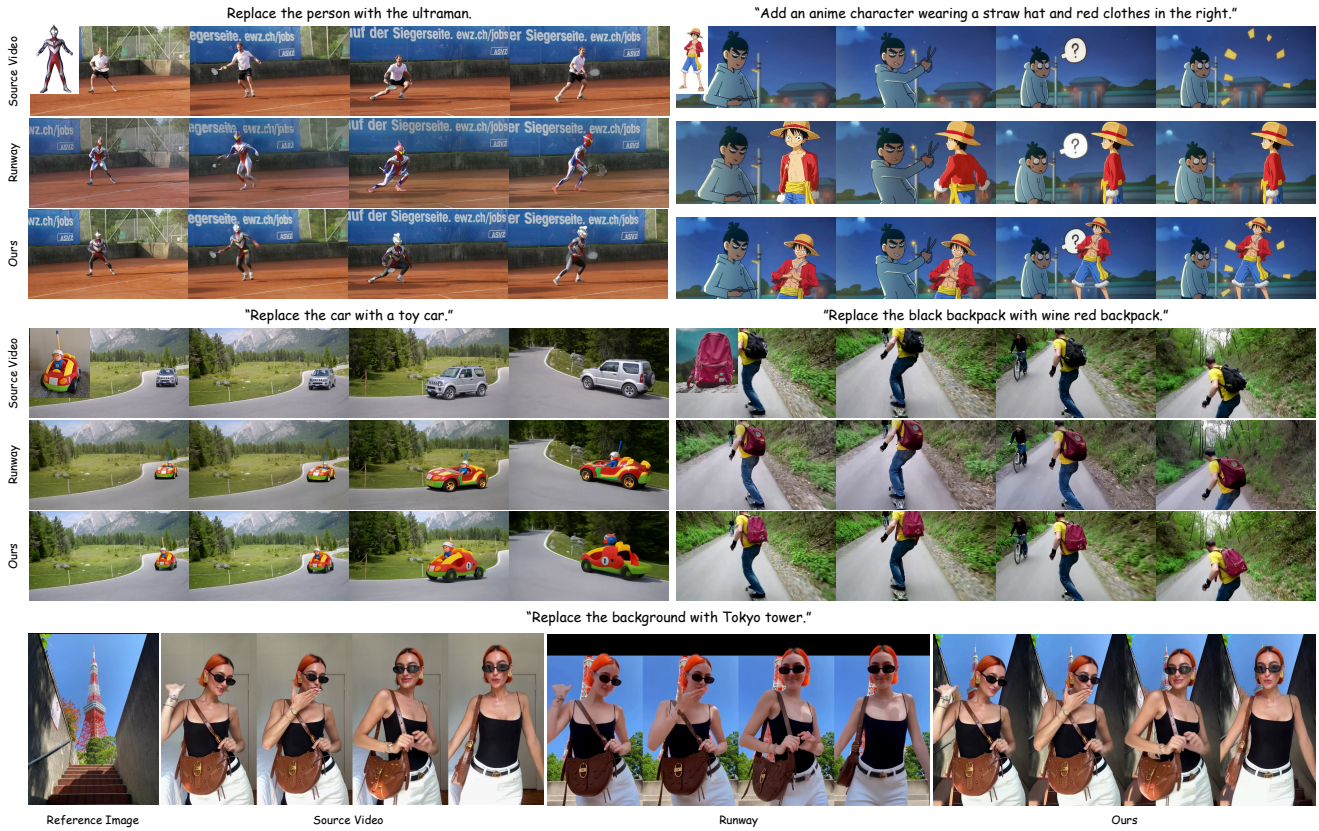

Figure 3. Qualitative comparison of the reference-based video editing on the VIE-Bench [8]. The editing instruction is shown at the top and the reference image is shown on the left for each group of results.

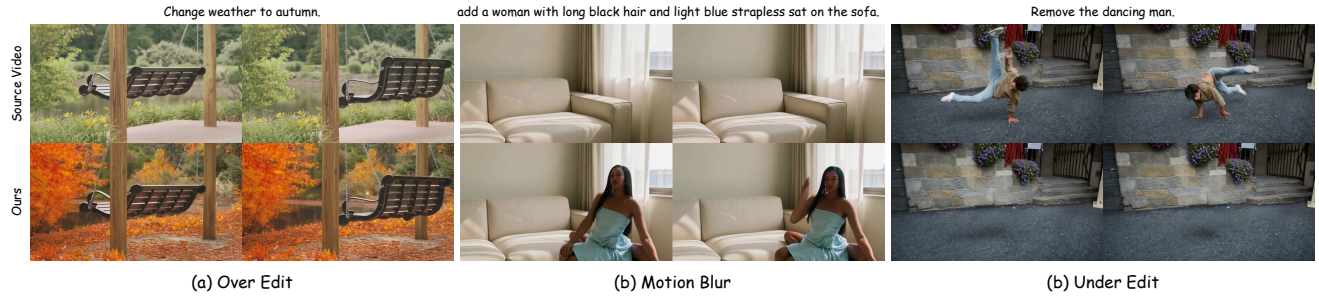

Figure 4. Failure cases. (a) Global transformations such as changing weather sometimes cause over-editing. (a) Rapid motion might occasionally lead to blurry results, such as the woman's hand. (c) Under-editing might be observed in removal tasks, where residual artifacts, such as cast shadows, remain.

4.4. VLM Evaluation

We employ Gemini-2.5-Pro [2] as the VLM evaluator. Figure 9 presents our VLM evaluation templates for the instruction-based video editing and reference-instruction-based video editing. The VLM evaluator offers a scalable, human-aligned assessment. It is provided with the source frame, the edited frame, the instruction and an optional reference image as inputs, and evaluates the editing performance on a scale of 0 (worst) to 10 (best) across three four critical dimensions: Instruction Following (instruction adherence), Source Video Preservation (consistency with the source video), Editing Quality (overall video aesthetics), and Subject Similarity (for reference-based edits).

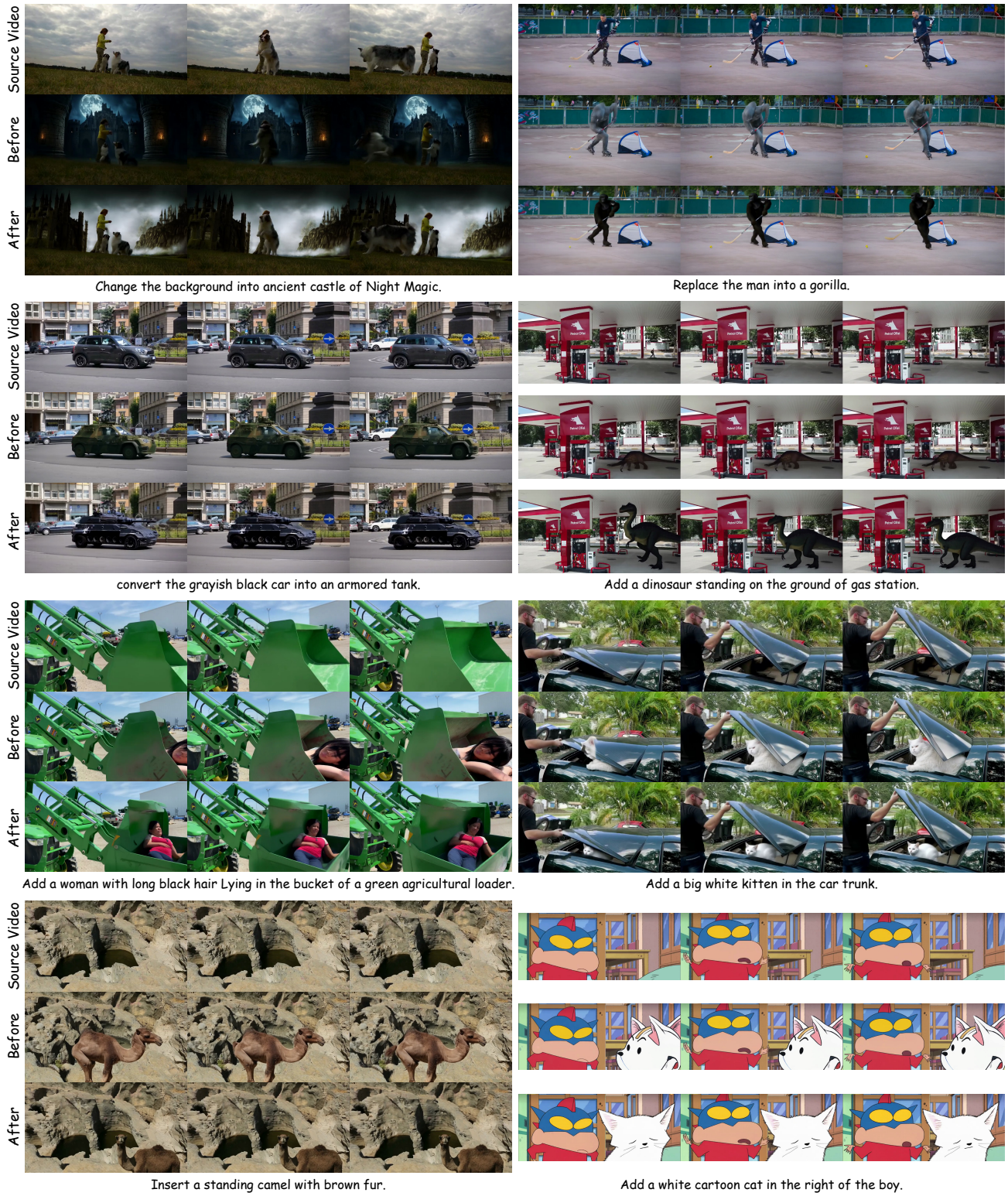

Figure 5. Ablation studies on Edit-GRPO. Before: without Edit-GRPO; After: with Edit-GRPO. The editing instruction is shown at the bottom.

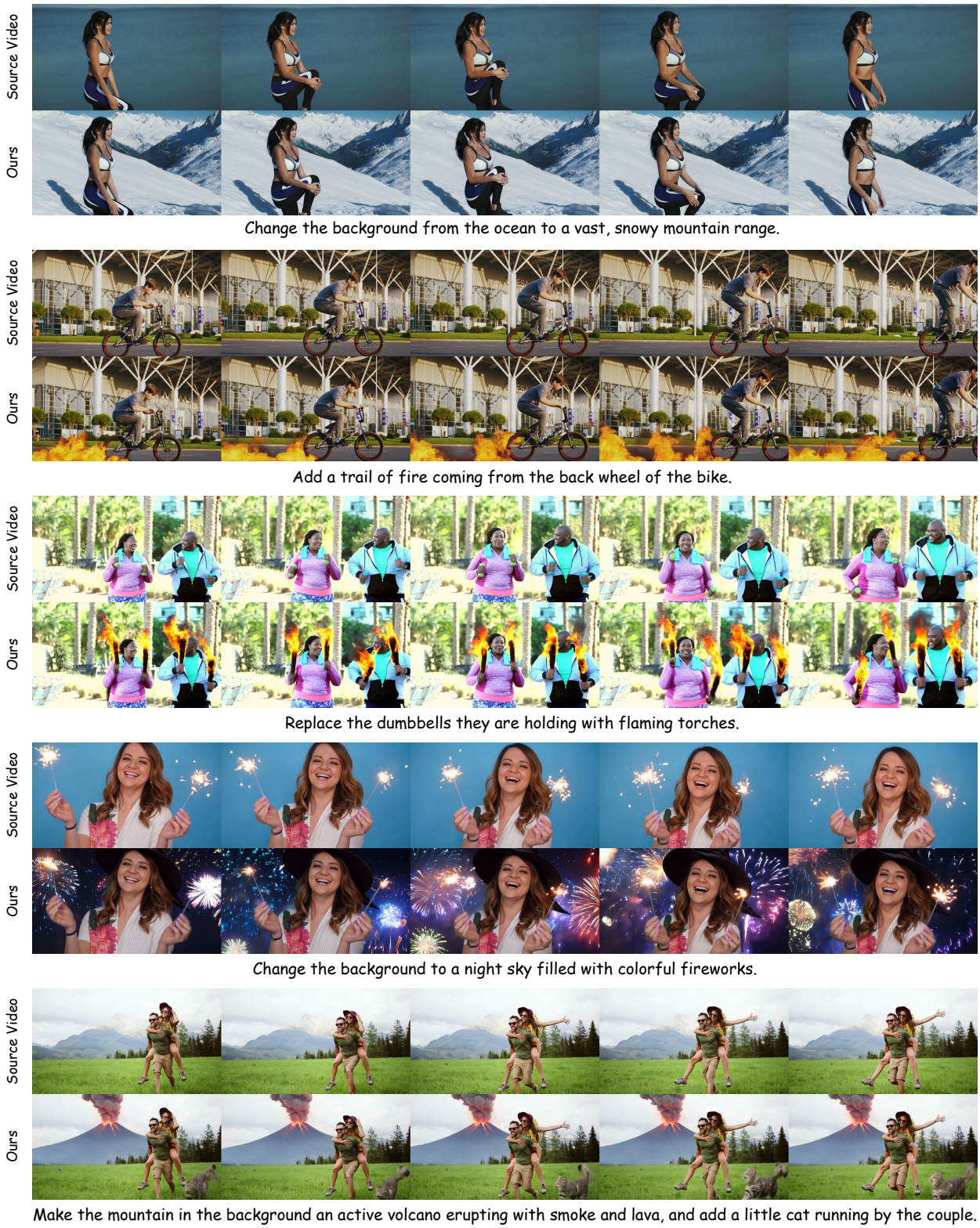

Figure 6. Qualitative results. We present more results of our method on complex instructions that are non-trivial and challenging to be synthesized by the data construction pipeline. The editing instruction is shown at the bottom.

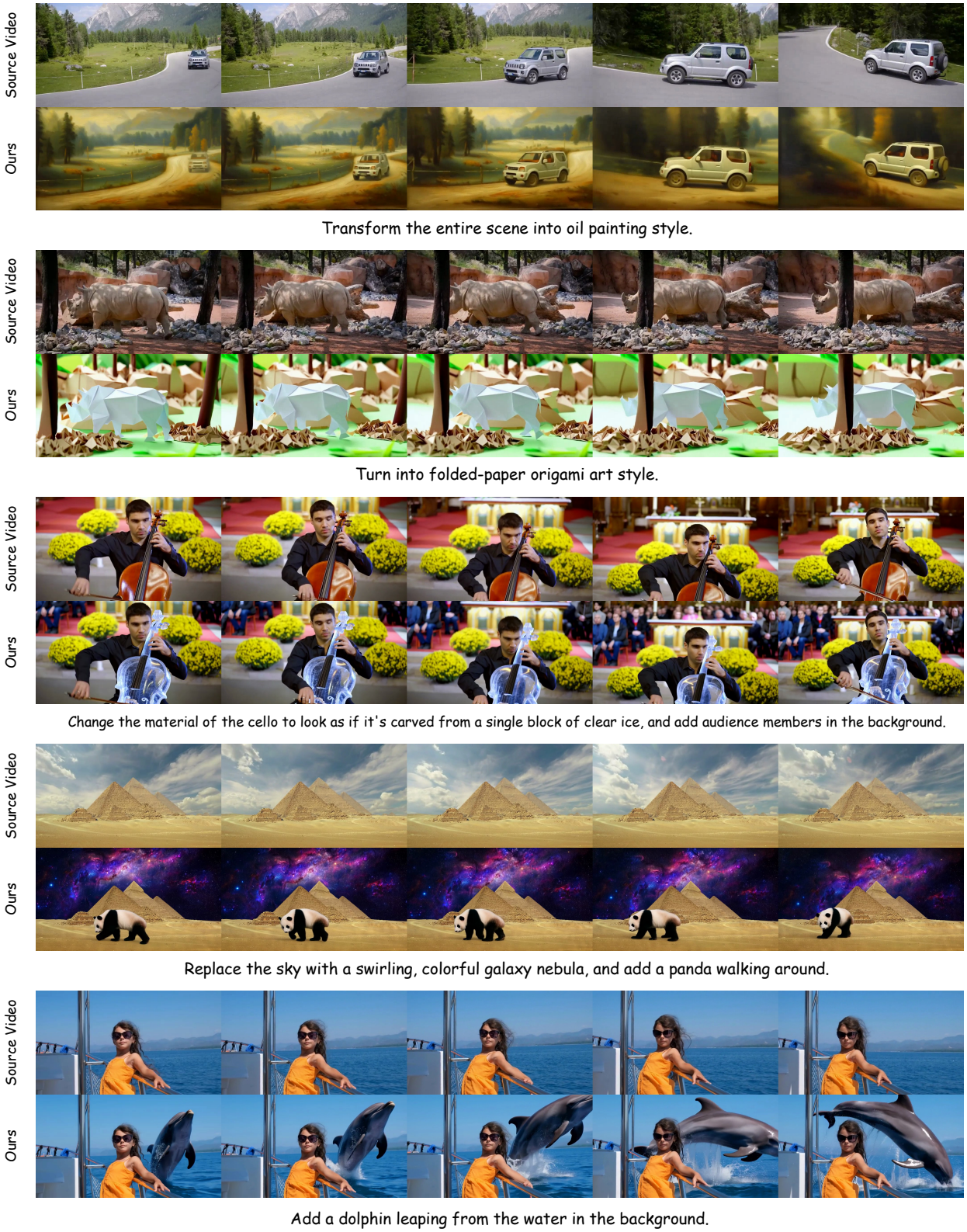

Figure 7. Qualitative results. We present more results of our method on complex instructions that are non-trivial and challenging to be synthesized by the data construction pipeline. The editing instruction is shown at the bottom.

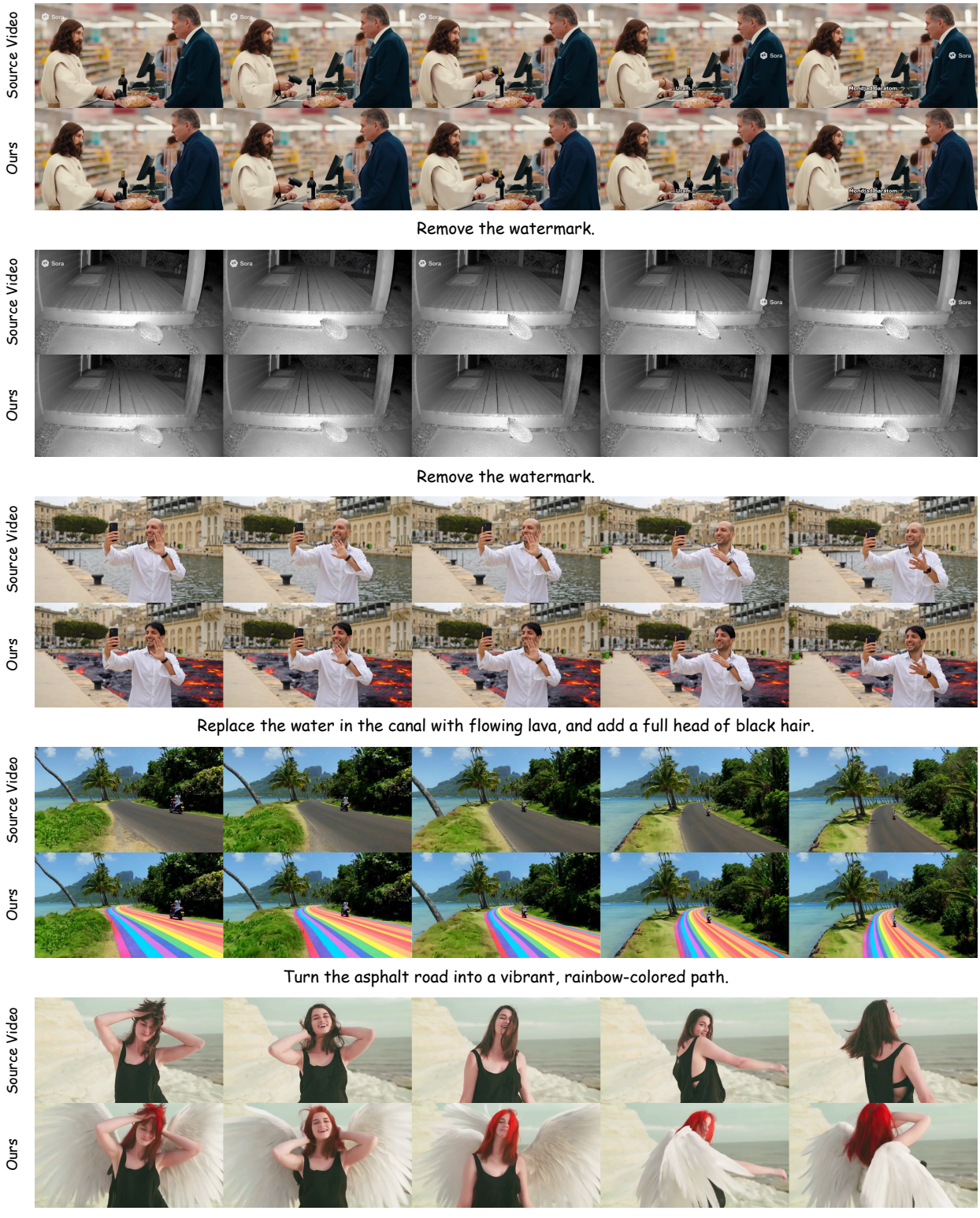

Figure 8. Qualitative results. We present more results of our method on complex instructions that are non-trivial and challenging to be synthesized by the data construction pipeline. The editing instruction is shown at the bottom.

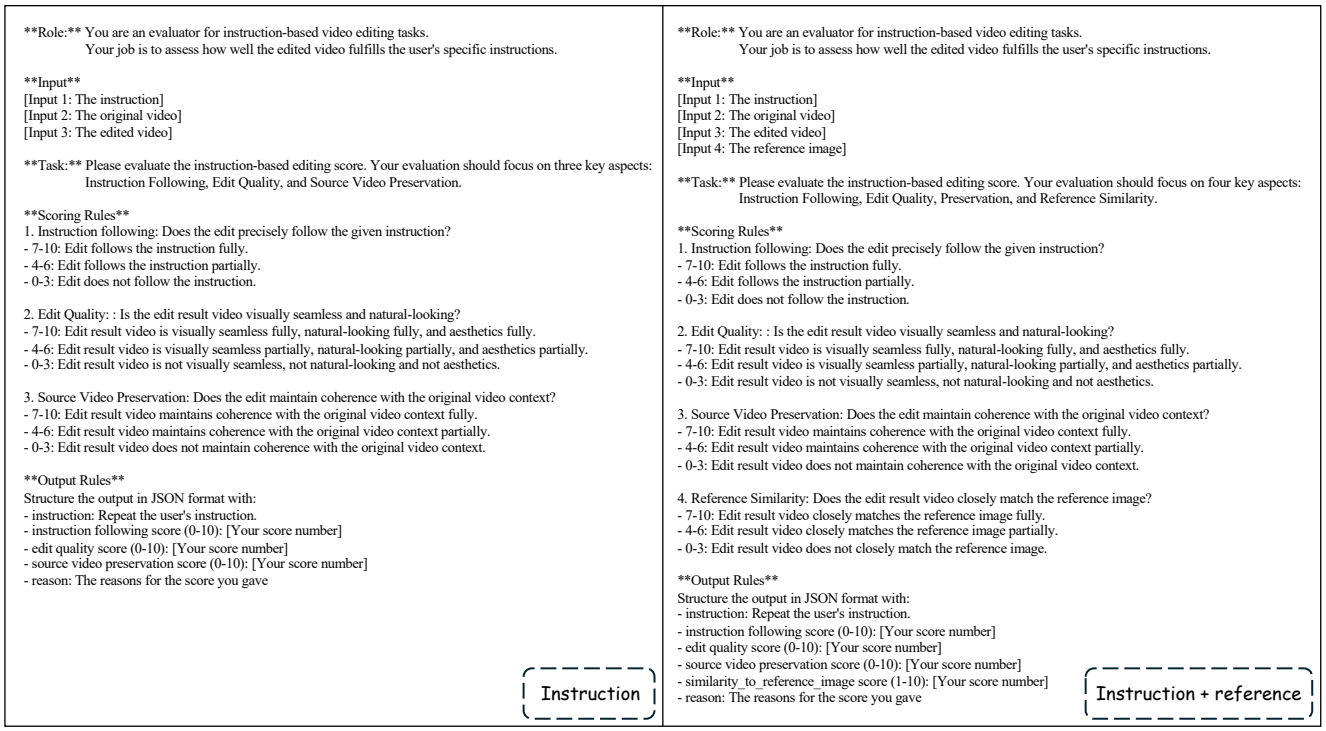

Figure 9. VLM templates for the instruction-based video editing and reference-instruction-based video editing.

5. Limitations

Figure 4 presents three failure cases of VIVA. Despite the exceptional generalization capabilities of VIVA, it encounters challenges in specific cases. Rapid motion might occasionally lead to blurry outputs, such as the woman's hand. Furthermore, VIVA sometimes struggles to balance editing intensity: it tends to exhibit over-editing in global transformations (such as weather or style changes) while showing under-editing in removal tasks, where residual artifacts—such as cast shadows—often remain.

References

[1] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 2

[2] Google. Gemini 2.5: Our most intelligent ai model, 2025. 2, 4

[3] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 1

[4] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Dax Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. Hunyuanvideo: A systematic framework for large video generative models, 2025. 1

[5] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 1

[6] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025. 1

[7] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. 1

[8] Chong Mou, Qichao Sun, Yanze Wu, Pengze Zhang, Xinghui Li, Fulong Ye, Songtao Zhao, and Qian He. Instructx: Towards unified visual editing with mllm guidance. *arXiv preprint arXiv:2510.08485*, 2025. 2, 4

[9] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 1

[10] Zihan Shao, Tianyang Cai, Yichang Zhou, et al. DeepSeek-

Math: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 1

[11] Feng Wang and Zihao Yu. Coefficients-preserving sampling for reinforcement learning with flow matching. *arXiv preprint arXiv:2509.05952*, 2025. 1