

# Supplementary for “Instruct-CLIP: Improving Instruction-Guided Image Editing with Automated Data Refinement Using Contrastive Learning”

Sherry X. Chen

Misha Sra

Pradeep Sen

University of California, Santa Barbara

{xchen774, sra, psen}@ucsb.edu

In this supplementary material, we first discuss the differences between our approach and CLIP directional similarity (Sec. A). Next, we provide additional implementation details in Sec. B. We then compare the performance of edit instruction refinement between our approach and vision-language models in Sec. D. Following that, we highlight the limitations of CLIP/DINO metrics in Sec. C. Finally, we present additional editing results, refined editing instructions, and further failure cases in Sec. E.

## A. CLIP Directional Similarity Comparison

One key difference between our I-CLIP and the CLIP directional similarity [2] used in InstructPix2Pix [1] is that I-CLIP leverages edit instructions rather than individual image prompts, which require two prompts per image pair. This makes I-CLIP readily applicable across instruction-guided image-editing datasets, even when image prompts are unavailable. Furthermore, individual prompts can often be lengthy and verbose, while edit instructions are typically more concise, reducing irrelevant information in the corresponding text embeddings.

For example, consider a prompt pair from the IP2P dataset: “Infinity walk by Marcelo Archila - Black & White Landscapes (contrast, monochrome, hdr, black and white, fine art, long exposure)” and “Infinity walk by Marcelo Archila - Black & White Landscapes (contrast, monochrome, hdr, black and white, commercial).” At first glance, it may be challenging to infer the edit instruction, which in this case is simply: “make it commercial.”

## B. Implementation and Dataset Refinement

LD-DINOv2 is initialized from a ViT-L/14 DINOv2 model [9]. To accommodate Stable Diffusion (SD) VAE [12] encoded images, the patch embedding projection layer is replaced. Additionally, the timestep embedding projection module is initialized to handle timestep inputs following the SD timestep encoding implementation.

The model is trained on the InstructPix2Pix (IP2P) [1] dataset, with all images resized to  $256 \times 256$ . Training is conducted with a learning rate of  $10^{-5}$ , a batch size of 32, and a total of 100K training steps.

During the first 10K steps, the timesteps are fixed to 0. This ensures that latent image inputs are not noisified, which allows the patch embedding projection layer to learn to encode latent images effectively. Then, the upper bound of the timestep is linearly increased in proportion to the training step number, reaching the maximum value of 1000 at the  $90K^{\text{th}}$  step. During this period, the timestep value is uniformly sampled between 0 and the current upper bound for each training step. This gradual increase in the timestep value sampling range helps preserve the knowledge learned by the patch embedding projection layer while simultaneously adapting the timestep embedding projection module.








	HIVE	I-Inp	WYS	ZONE	MB	IP2P	Ours	
CLIP-T	0.307	0.303	<b>0.325</b>	0.225	0.312	0.224	0.317	
CLIP-T	0.278	0.276	0.290	0.299	0.279	<b>0.304</b>	0.287	
CLIP-T	0.362	0.241	<b>0.369</b>	<b>0.369</b>	0.362	0.365	0.360	
CLIP-T	0.238	0.196	0.216	<b>0.280</b>	0.267	0.243	0.240	
CLIP-I	0.827	0.637	<b>0.890</b>	0.887	0.850	0.716	0.862	
DINO-I	0.267	0.326	<b>0.662</b>	0.081	0.070	0.393	0.572	
CLIP-T	0.375	0.284	0.381	0.375	<b>0.395</b>	0.351	0.384	
CLIP-I	0.891	0.875	0.903	0.879	<b>0.961</b>	0.826	0.930	
DINO-I	0.658	0.536	0.731	0.712	0.603	0.666	<b>0.860</b>	
CLIP-T	0.311	0.311	<b>0.335</b>	0.287	0.324	0.295	0.332	
CLIP-I	0.912	0.925	0.959	0.858	0.968	0.865	<b>0.971</b>	
DINO-I	0.933	0.816	<b>0.958</b>	0.605	0.940	0.851	0.940	
CLIP-T	0.318	0.315	<b>0.342</b>	0.273	0.306	0.217	0.305	
CLIP-I	0.943	0.953	0.946	0.750	<b>0.954</b>	0.703	0.937	
DINO-I	0.953	0.956	0.929	0.152	<b>0.960</b>	0.050	0.950	
CLIP-T	0.314	0.232	0.327	0.333	0.313	<b>0.337</b>	0.333	
CLIP-T	0.314	0.256	0.296	0.311	0.294	0.308	<b>0.315</b>	

Table 4. Metrics (CLIP-I/DINO-I if GT exists) for Fig. 5 outputs, with best bolded and shown.

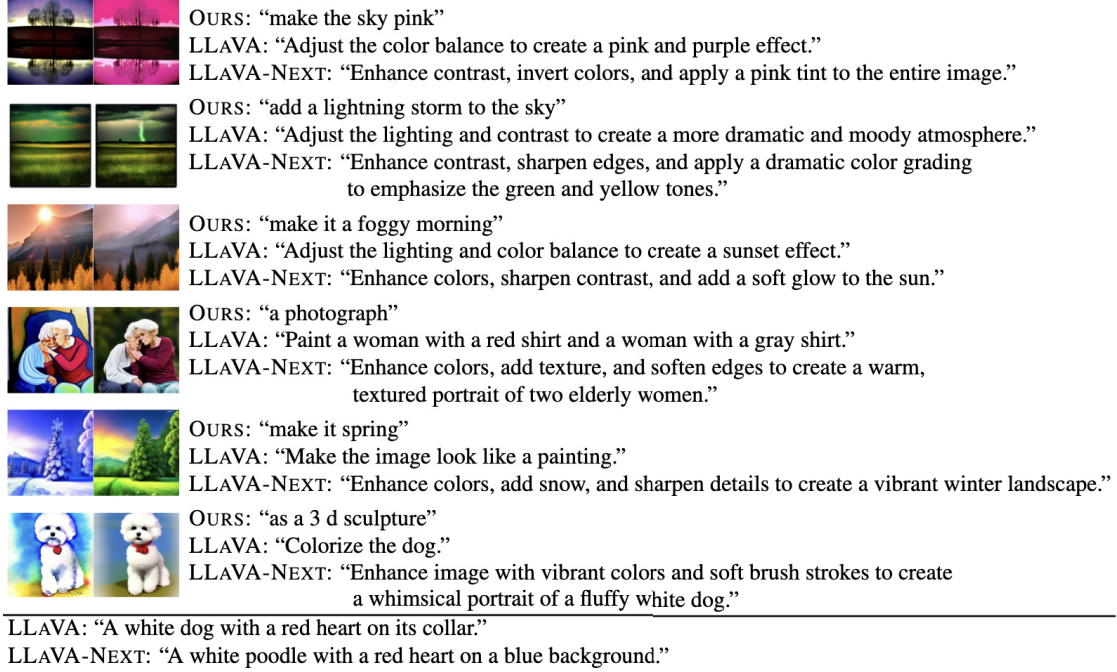


Figure 8. **Top:** VLM outputs with respect to prompt "Provide the edit instruction that can transform the source image to the target image in one phrase:" and image pairs in Fig. 2. **Bottom:** VLM outputs with respect to prompt "Describe the image in one phrase:" and the input (leftmost) image in the last row.

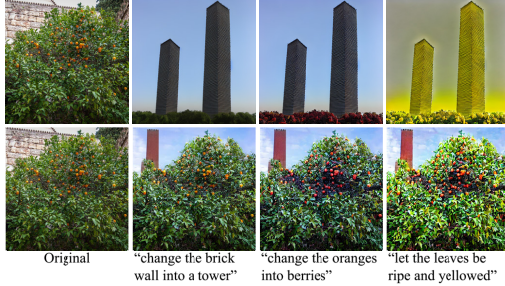


Figure 9. Multi-turn edit comparison with InstructPix2Pix (IP2P) [1]. Note how IP2P (top row) gradually diverges from the desired result more and more, unlike our approach (bottom row) which produces results more consistent with the original.

For the last 10K steps, timesteps are randomly sampled across the entire range. This strategy ensures that the model learns to handle the full distribution of timestep values.

Instruct-CLIP (I-CLIP in short) is initialized from a ViT-L/14 CLIP model [11]. We freeze the text decoder, the aforementioned LD-DINOv2, and finetune the CLIP image encoder. The instruction decoder follows the architecture of DeCap [6] with a pre-trained GPT-2 backbone [10] and is trained along with the image encoder on the IP2P dataset with a learning rate of  $10^{-5}$ , a batch size of 32, and a total of 100K training steps.

The advantage of training LD-DINOv2 ahead of time is

that we can sample timesteps randomly within its maximum possible range without repeating the above training procedure, as LD-DINOv2 has already learned to "ignore" the noise added to the latent image.

After training, we refine the IP2P dataset. For each data sample  $(I^o, I^e, p)$  and its corresponding refined instruction  $p'$ , we update the sample if the I-CLIP cosine similarity between the visual changes in the original/edited image and the refined instruction differs significantly from that with the original instruction:

$$\begin{aligned} & \text{sim}(\text{Instruct-CLIP}_{\text{vis}}(L^o, 0, L^e, 0), \text{Instruct-CLIP}_{\text{text}}(p^R)) \\ & > \text{sim}(\text{Instruct-CLIP}_{\text{vis}}(L^o, 0, L^e, 0), \text{Instruct-CLIP}_{\text{text}}(p)) + \phi, \end{aligned} \quad (1)$$

where

$$\begin{aligned} L^o &= \text{VAE}_{\text{enc}}(I^o), \\ L^e &= \text{VAE}_{\text{enc}}(I^e), \end{aligned} \quad (2)$$

and  $\phi = 0.1$  is the margin. This results in over 120K new instructions out of 313,010 samples in the IP2P dataset. We retain the original instructions for the remaining samples.

The image editing model is initialized from the IP2P model [1], where the UNet [13] is fine-tuned using Low-Rank Adaptation (LoRA) [3] with parameters  $r = \alpha = 32$  on the newly generated samples. The training is performed with a learning rate of  $10^{-4}$ , a batch size of 64, and a total

of 10K training steps. The rest of the training configuration follows the original IP2P work.

### C. Limitation of CLIP/DINO Metrics

There are several reasons for the gap between our qualitative and quantitative results, which we include for completeness. First, while these metrics are widely used, they have well-documented limitations and do not align with human judgment, as highlighted by VIEScore [4]. Specifically, Yuksekgonul et al. [14] show that CLIP’s Bag-of-Words behavior is insensitive to word order, leading to weak correlations with human evaluations.

This issue is also evident in Table 4; despite the superior qualitative performance of our results in Fig. 5, our metrics are usually lower than the baselines. Additionally, the results presented in Table 1 (MagBr data) are computed on the MagBr test set, which has a distribution similar to their training set, giving MagBr an inherent advantage.

### D. Comparison with Vision Language Models

To compare our method with vision-language models (VLMs) in terms of edit instruction refinement, we evaluate LLaVA [7] and LLaVA-Next [8], two widely used open-source VLMs, as shown in Fig. 8 (top). Both VLMs fail to generate effective edit instructions compared to our method.

In Fig. 8 (bottom), we use these VLMs to generate a caption for the input image (last row), which serves as the input prompt for editing methods that require separate prompts for the input and target images. While the caption accurately describes the image, it fails to capture its watercolor style—a crucial detail needed for the intended style editing in this sample pair. Consequently, users still need to manually refine the input prompt and compose the target image prompt, which is significantly more cumbersome than using a single edit instruction.

### E. Additional Results

We include the multi-turn edit example (Fig. 9) mentioned in the paper. Additionally, we provide more instruction-guided image editing results in Figs. 10 and 11, as well as samples from our dataset with refined instructions in Figs. 12 and 13. Lastly, we present additional failure cases in Fig. 14.

### References

- [1] Tim Brooks, Aleksander Holynski, and Alexei A Efros. InstructPix2Pix: Learning to follow image editing instructions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18392–18402, 2023. 1, 2
- [2] Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. StyleGAN-NADA: CLIP-guided domain adaptation of image generators. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022. 1
- [3] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2
- [4] Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhu Chen. VIEScore: Towards explainable metrics for conditional image synthesis evaluation. *arXiv preprint arXiv:2312.14867*, 2023. 3
- [5] Shanglin Li, Bohan Zeng, Yutang Feng, Sicheng Gao, Xiuhui Liu, Jiaming Liu, Lin Li, Xu Tang, Yao Hu, Jianzhuang Liu, et al. ZONE: Zero-shot instruction-guided local editing. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6254–6263, 2024. 4, 5, 8
- [6] Wei Li, Linchao Zhu, Longyin Wen, and Yi Yang. DeCap: Decoding CLIP latents for zero-shot captioning via text-only training. *arXiv preprint arXiv:2303.03032*, 2023. 2
- [7] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 3
- [8] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 3
- [9] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1
- [10] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 2
- [11] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR, 2021. 2
- [12] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 1
- [13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 2
- [14] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? *arXiv preprint arXiv:2210.01936*, 2022. 3
- [15] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. MagicBrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024. 4, 5, 8
- [16] Shu Zhang, Xinyi Yang, Yihao Feng, Can Qin, Chia-Chih Chen, Ning Yu, Zeyuan Chen, Huan Wang, Silvio Savarese,

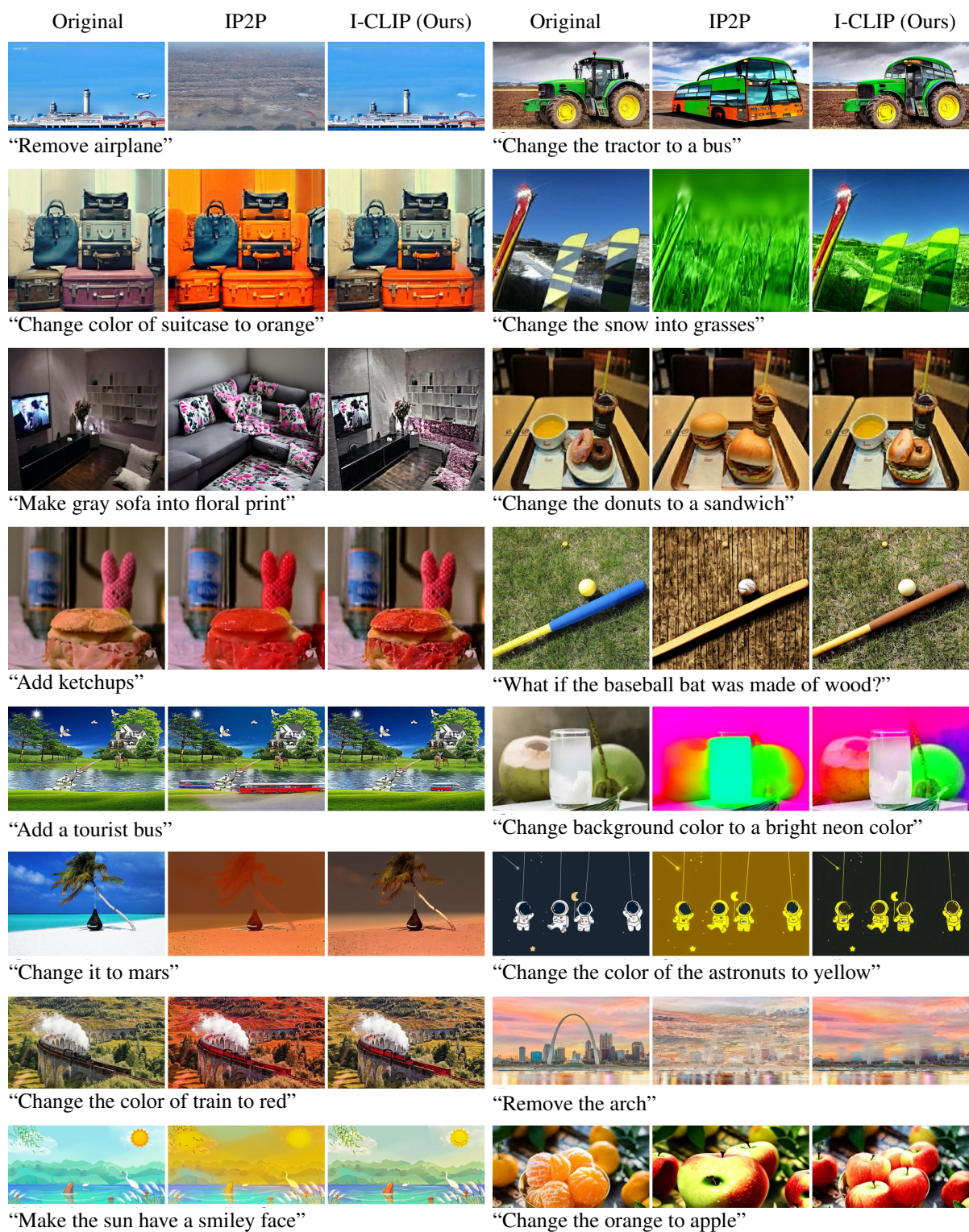


Figure 10. Additional results from our Instruct-CLIP image editing method on benchmarks [5, 15–17] (Part 1/2)

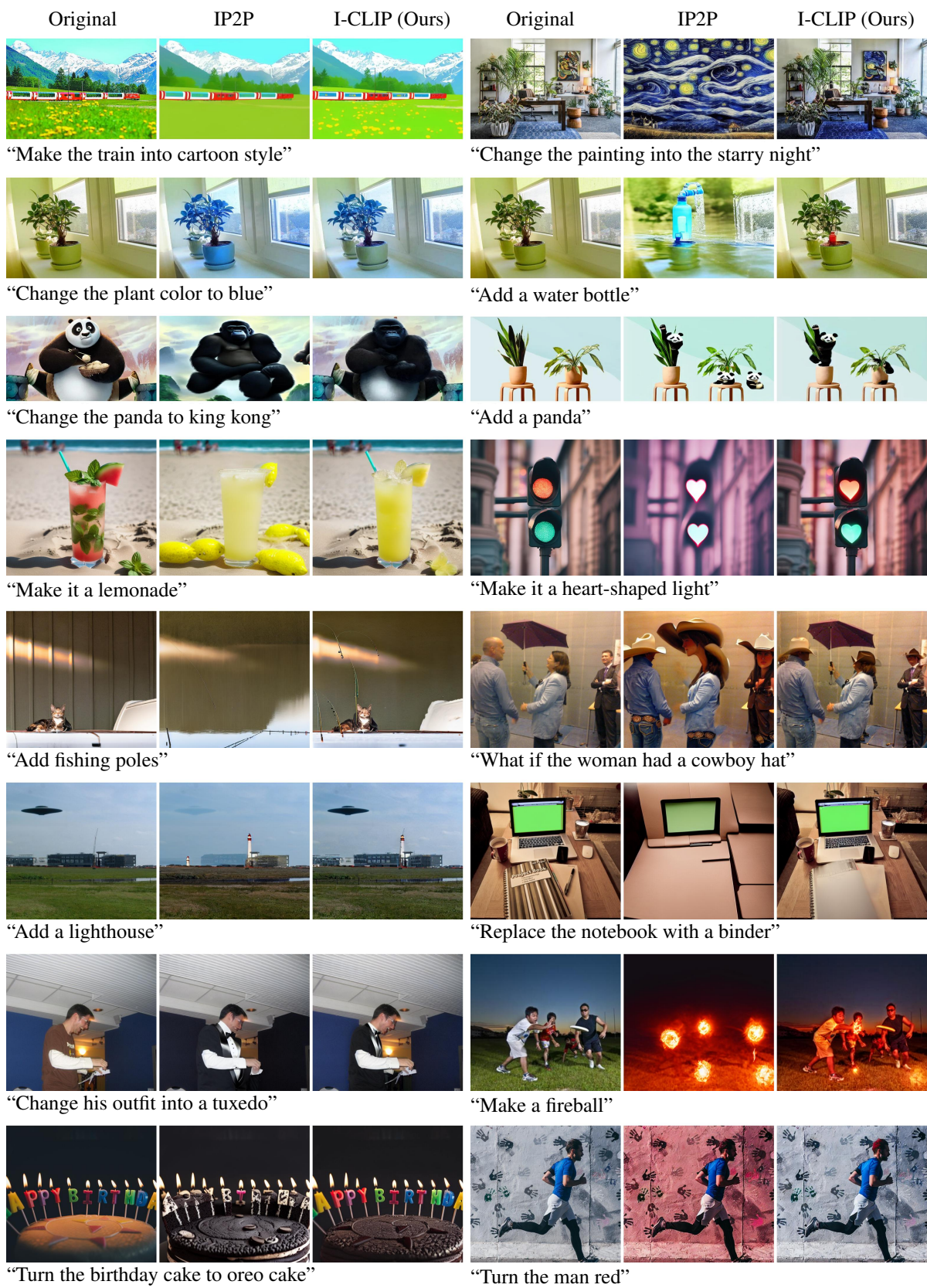


Figure 11. Additional results from our Instruct-CLIP image editing method on benchmarks [5, 15–17] (Part 2/2)

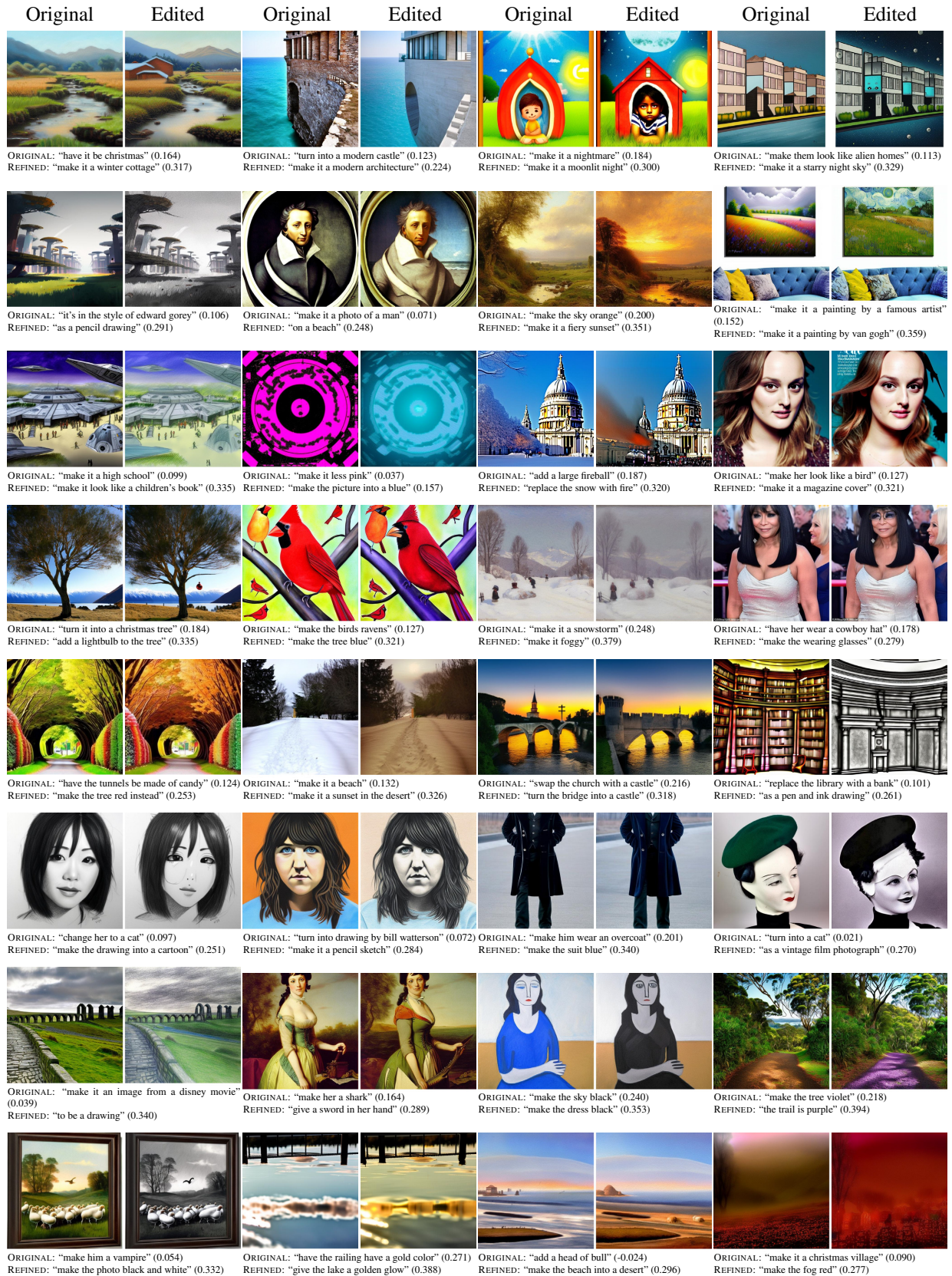


Figure 12. Additional refined instruction from our dataset (Part 1/2)

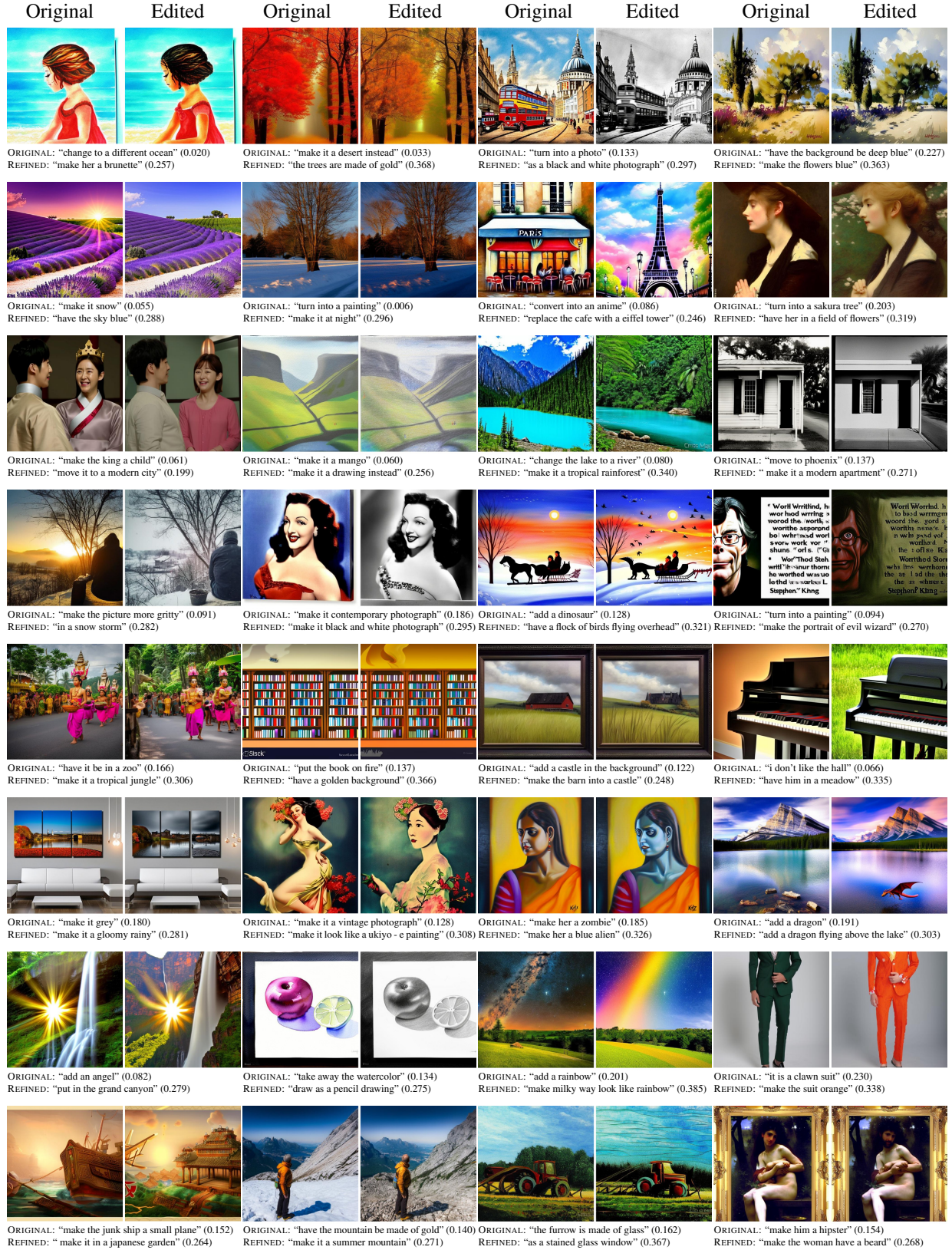


Figure 13. Additional refined instruction from our dataset (Part 2/2)

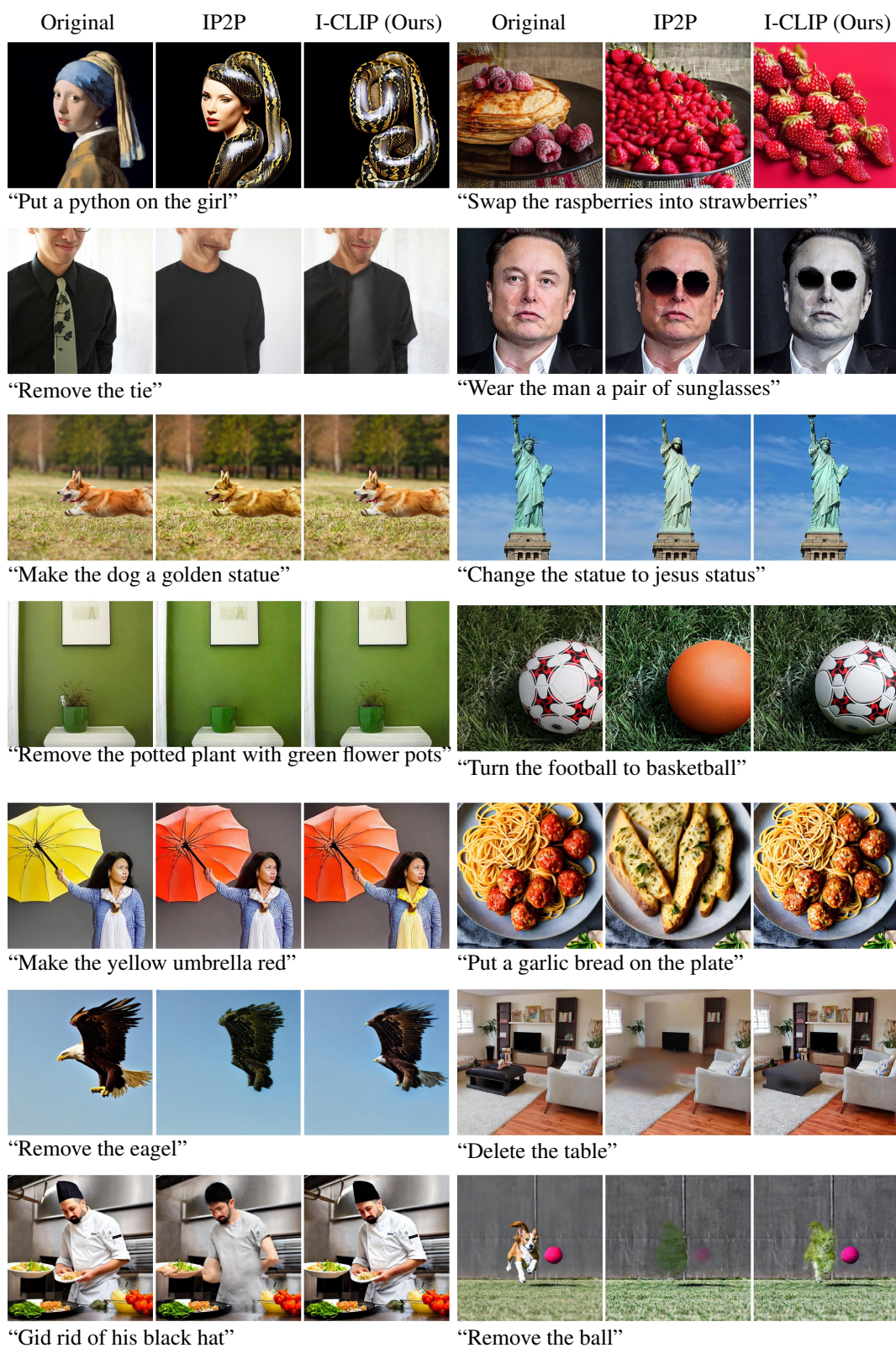


Figure 14. Additional failure cases from our Instruct-CLIP image editing method on benchmarks [5, 15–17])

Stefano Ermon, Caiming Xiong, and Ran Xu. HIVE: Harnessing human feedback for instructional visual editing. *arXiv preprint arXiv:2303.09618*, 2023.

- [17] Ruoyu Zhao, Qingnan Fan, Fei Kou, Shuai Qin, Hong Gu, Wei Wu, Pengcheng Xu, Mingrui Zhu, Nannan Wang, and Xinbo Gao. InstructBrush: Learning attention-based instruction optimization for image editing. *arXiv preprint arXiv:2403.18660*, 2024. [4](#), [5](#), [8](#)