SmartEdit: Exploring Complex Instruction-based Image Editing with Multimodal Large Language Models

Supplementary Material

In supplementary material, we provide the following contents:

1. Implementation Details of SmartEdit.
2. Details of the data production pipeline.
3. Discussion of the effectiveness of Bidirectional Interaction Module (BIM).
4. More quantitative comparisons on Reason-Edit.
5. More visual results on Reason-Edit.
6. Results of SmartEdit and other methods on MagicBrush.
7. Difference between SmartEdit, MGIE [4] and InstructDiffusion [5].

1. Implementation Details of SmartEdit

As we have already introduced the network architecture, training datasets and evaluation Metrics in the main body of the paper, we will additionally give out more implementation details of SmartEdit.

Training Process. The training process of SmartEdit is divided into two main stages. In the first stage, the MLLM is aligned with the CLIP text encoder [15] using the QFormer [10]. In the second stage, we optimize SmartEdit. To be specific, the weights of LLaVA are frozen and LoRA [7] is added for efficient fine-tuning. Since InstructDiffusion also trains on the segmentation dataset, for convenience, we directly use its weights as the initial weights for the diffusion model in SmartEdit. During the second stage, QFormer, BIM module, LoRA, and UNet [17] in the diffusion model are fully optimized.

Implementation Details. During the first stage of training, the AdamW optimizer [14] is used, and the learning rate and weight decay parameters are set to 2e-4 and 0, respectively. The training objectives at this stage are the combination of the mse loss between the output of LLaVA and clip text encoder, and the language model loss. The weights of both losses are 1. In the second stage, we also adopt the AdamW optimizer. The values of learning rate, weight decay, and warm-up ratio were set to 1e-5, 0, and 0.001, respectively. In this phase, the loss function is composed of two parts: the language model loss and the diffusion loss. The ratio of these two losses is 1:1.

Training Datasets. Since the training process of SmartEdit are divided into two phases. In the first stage, we utilize the extensive corpus CC12M [3] as our primary data source. In the second stage, the training data can be divided into 4 categories: (1) segmentation datasets, which include COCOStuff [2], RefCOCO [19], GRefCOCO [12], and the reasoning segmentation dataset from LISA [9]; (2) editing datasets, which involve InstructPix2Pix and MagicBrush; (3) visual question answering (VQA) dataset, which is the LLaVA-Instruct-150k dataset [13]; (4) synthetic editing dataset, where we collect a total of 476 paired data for complex understanding and reasoning scenarios.

2. Details of the Data Production Pipeline

As we mentioned in the main paper (Section 4.3), to effectively stimulate SmartEdit's editing capabilities for more complex instructions, we synthesize approximately 476 paired data as a supplement to the training data. This training dataset includes two major types of scenarios: complex understanding scenarios and reasoning scenarios.

For complex understanding scenarios, we establish a data production pipeline, which is illustrated in Fig. 1. To be specific, We begin with two images, x_1 and x_2, collected from the internet. Using the SAM [8] algorithm, we detect specific animals in these images. In image x_1, we identify a cat (mask_1) that we aim to replace, and in x_2, we identify a rabbit (mask_2) that we intend to use as a replacement. Following this, we apply the inpainting algorithm MAT [11] to x_1 and mask_1, creating a new image, y_1, where the cat has been seamlessly removed. To prepare the rabbit from x_2 for insertion into y_1, we apply resize and filter operations to mask_1, mask_2, and x_2, resulting in a new image, y_2. We then merge y_1 and y_2 to form y_3, which features the rabbit in the place of the cat. Due to potential differences in saturation, contrast, and other parameters between x_1 and x_2, the rabbit may not blend well with the rest of the image. To rectify this, we apply the harmonization algorithm PIH [18] to y_3 to obtain a more harmonious image, y_4. By utilizing some images in the entire process, we can obtain two pairs of training samples: where (y_1, x_1, *"Add a cat to the right of the cat"*) can form one pair of training samples, with y_1 as the original image and x_1 as the ground truth; (x_1, y_4, *"Replace the smaller cat with a rabbit"*) can also form a pair of training samples, with x_1 as the original image and y_4 as the ground truth. In Fig. 2, the first two rows show some complex understanding samples contained in the training data.

For reasoning scenarios, we first generate the corresponding object's mask through SAM [8], then adopt stable diffusion [16] to perform inpainting based on the provided instruction. Since the inpainting process can sometimes generate failure cases, we further manually filter the unsatisfied image. In the last row of Fig. 2, we illustrate some reasoning samples that are included in training data.

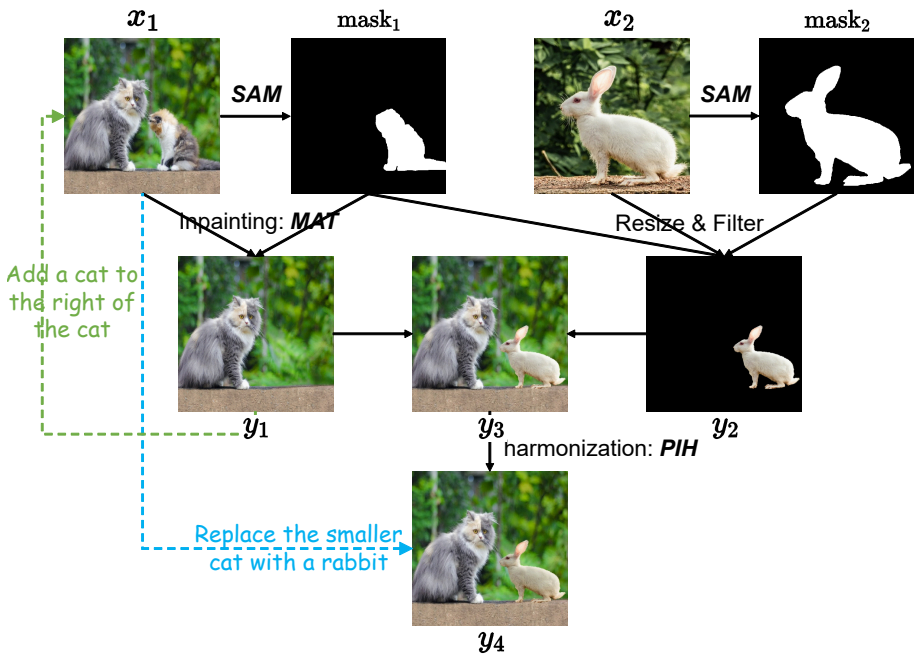

Figure 1. The data production pipeline of the synthetic paired training set (complex understanding scenarios). For x_1 and x_2, we first use SAM to generate mask_1 and mask_2. Then, we use MAT, combined with x_1 and mask_1, to get y_1. At the same time, by performing specific operations on mask_1, mask_2, and x_2, we can get y_2. By combining y_1 and y_2, we can get y_3. Finally, we use the harmonization algorithm PIH to get y_4. (y_1, x_1, *"Add a cat to the right of the cat"*) and (x_1, y_4, *"Replace the smaller cat with a rabbit"*) can form the training samples.

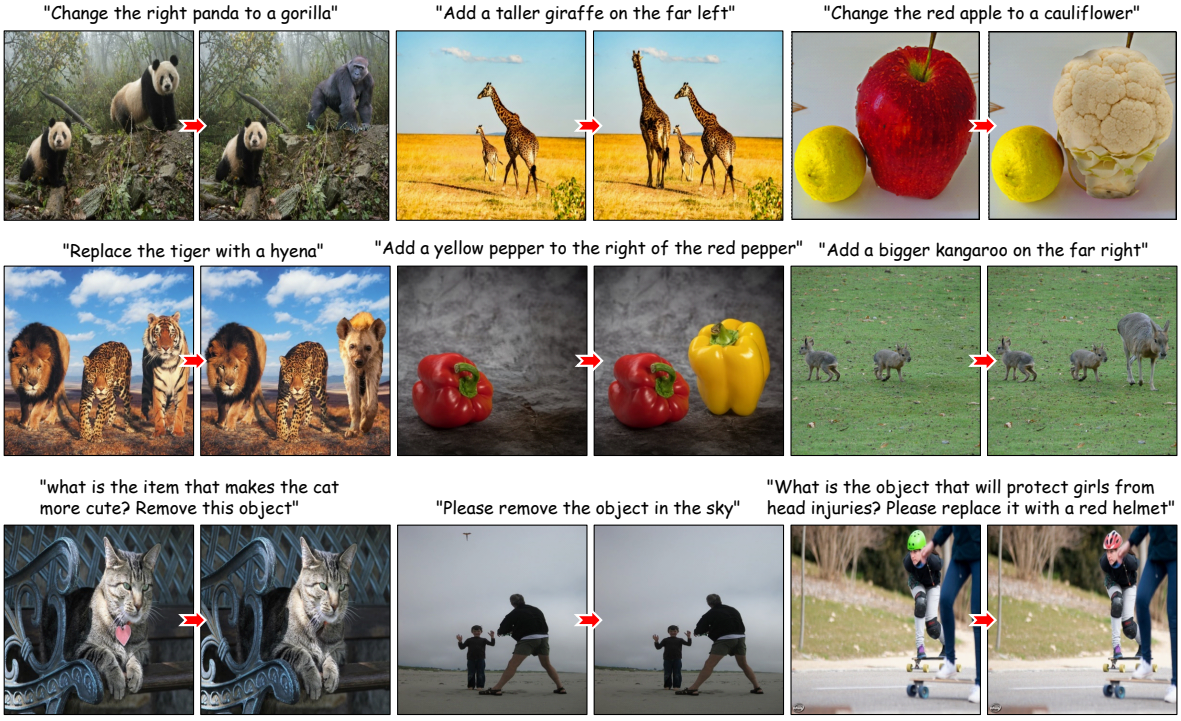

Figure 2. Samples of complex understanding and reasoning scenarios in our synthesized paired training data. For each sample, the image on the left is the input image, and the image on the right is the image edited according to the instructions above.

Exp	QFormer	BIM	Self-Attention	MLP	Understanding Scenarios					Reasoning Scenarios				
					PSNR(dB)↑	SSIM↑	LPIPS↓	CLIP Score↑	Ins-align↑	PSNR(dB)↑	SSIM↑	LPIPS↓	CLIP Score↑	Ins-align↑
1		✓			20.652	0.697	0.107	19.15	0.318	19.698	0.678	0.128	17.76	0.089
2	✓		✓		21.978	0.726	0.091	23.51	0.695	24.586	0.731	0.069	20.82	0.628
3	✓			✓	21.848	0.728	0.089	23.40	0.697	23.447	0.715	0.085	20.24	0.667
4 (SmartEdit-7B)	✓	✓			**22.049**	**0.731**	**0.087**	**23.61**	**0.712**	**25.258**	**0.742**	**0.055**	**20.95**	**0.789**
5	Finetuned GILL				20.567	0.705	0.116	22.34	0.532	20.734	0.656	0.147	20.05	0.533

Table 1. Quantitative comparison on Reason-Edit. Exp1 $\sim$ Exp4 are conducted based on the SmartEdit-7B.

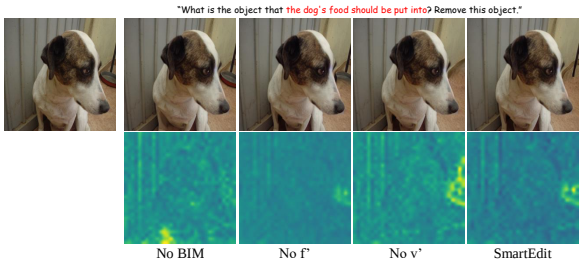

Figure 3. Cross-attention map visualization for the effectiveness of BIM.

3. Discussion of the effectiveness of Bidirectional Interaction Module (BIM)

3.1. Why BIM is effective?

BIM can assist SmartEdit in accurately locating the areas that need modification, and make the necessary changes without affecting other areas that do not require modification. Specifically, the f branch integrates the vision feature through the cross-attention block, and the v branch also incorporates the text feature through cross-attention, thereby enhancing both information mutually. According to the results shown in Tab. 1 and Fig. 3, the existence of BIM is crucial, and BIM surpasses all different architectural designs. In the specific case study of cross-attention map visualization shown in Fig. 3, f' can more precisely identify the entire object (i.e., the entire dog bowl) that needs modification. v' can better preserve the original information of areas that do not need to be modified (i.e., black pillar next to dog bowl).

3.2. Different architectural designs comparing with BIM

We conduct several different architectural design comparing with BIM. *1) QFormer utilization:* QFormer is often used to bridge the feature domain gap, and its effectiveness has been validated in BLIP2, BLIP-Diffusion, GILL, etc. Exp 1 and Exp 4 in Tab. 1 indicate the necessity of QFormer. *2) Comparison between BIM with Self-Attention or MLP layers:* As presented in Tab. 1 (Exp 2 $\sim$ Exp 4), SmartEdit with BIM achieves the best performance. *3) Comparison with GILL:* The performance of fine-tuned GILL is inferior to SmartEdit (Tab. 1, Exp 5). We attribute this to two factors: 1) the proposed BIM facilitates information exchange. 2) The performance of LLaVA in SmartEdit surpasses that of OPT-based MLLM in GILL.

4. More Quantitative Results on Reason-Edit

4.1. Instruction-Alignment Metric (Ins-align)

As mentioned in the main paper, PSNR/SSIM/LPIPS/CLIP-Score are the four most commonly used metrics in instruction-based image editing methods. For the foreground area, we calculate the CLIP Score [15] between the foreground area of the edited image and the GT label. For the background area, we calculate the PSNR/SSIM/LPIPS [6, 21] between the edited image and the original input image. While these metrics can reflect the performance to a certain extent, they are not entirely accurate. This can be confirmed in Fig. 4. Specifically, in the first row of results, SmartEdit successfully generates a chicken, while InstructDiffusion does not generate a real chicken well. However, the CLIP-Score metric ranks InstructDiffusion higher. In the second row of images, the CLIP-Score aligns more with visual judgment, ranking SmartEdit's results higher. This indicates that the CLIP-Score metric may not always match human visual assessment. Regarding the PSNR/SSIM/LPIPS metrics, there is a significant variation in the results between SmartEdit and InstructDiffusion. Visually, the images edited by these two methods (the first row and the second row) do not have much visual difference in the background area, which indicates that these three metrics also cannot always accurately reflect the effectiveness of the instruction-based image editing methods. To provide a more accurate evaluation of the effects of edited images, we propose a metric for assessing editing accuracy. Specifically, we hire four workers to manually evaluate the results of these different methods on Reason-Edit. The evaluation criterion is whether the edited image aligns with the instruction. After obtaining the evaluation results from each worker, we average all the results to get the final metric result, which is Instruction-Alignment (Ins-align).

For all the experimental results in the main paper, we include the results of the Ins-align indicator, as shown in Sec. 5.2, Sec. 5.3 and Sec. 5.4.

In Sec. 5.2, we compare the results of SmartEdit with different existing instruction editing methods. It can be observed that when we use a metric consistent with human visual perception (Ins-align), for complex understanding and reasoning scenarios, SmartEdit shows a significant improvement compared to previous instruction-based image editing methods. Also, when adopting a more powerful LLM model, SmartEdit-13B performs better than SmartEdit-7B on the Ins-align metric.

Sec. 5.3 and Sec. 5.4 present the results of the Ablation studies for BIM module and Dataset Usage, respectively. Sec. 5.3, based on the results from the Ins-align metric, the introduction of the BIM module and its bidirectional information interaction capability indeed enhance SmartEdit's instruction editing performance in complex understanding and reasoning scenarios. As shown in Sec. 5.4, the joint utilization of editing data, segmentation data, and synthetic editing data enables SmartEdit to deliver better results in complex understanding and reasoning scenarios.

4.2. User Study

To further verify the effectiveness of SmartEdit, we perform a user study. Specifically, we randomly select 30 images from Reason-Edit, of which 15 images belong to complex understanding scenarios, and the other 15 belong to reasoning scenarios. For each image, we obtain the results of InstructPix2Pix, MagicBrush, InstructDiffusion, and SmartEdit, and randomly shuffle the order of these method results. As we mentioned in the main paper, for fairness, all comparison methods undergo fine-tuning on the same dataset as SmartEdit. In the end, we get 30 groups of images with shuffled order. For each set of images, we ask participants to independently select the two best pictures. The first one is the best picture corresponding to the instruction (i.e., Instruct-Alignment), and the second one is the picture with the highest visual quality under the condition of having editing effects (i.e., Image Quality). A total of 25 people participate in the user study. The result is shown in Fig. 5. We can find that over 67% of participants think that the effect of SmartEdit corresponds better with the instructions and more than 72% of participants prefer the results generated by SmartEdit. This further suggests that SmartEdit is superior to other methods.

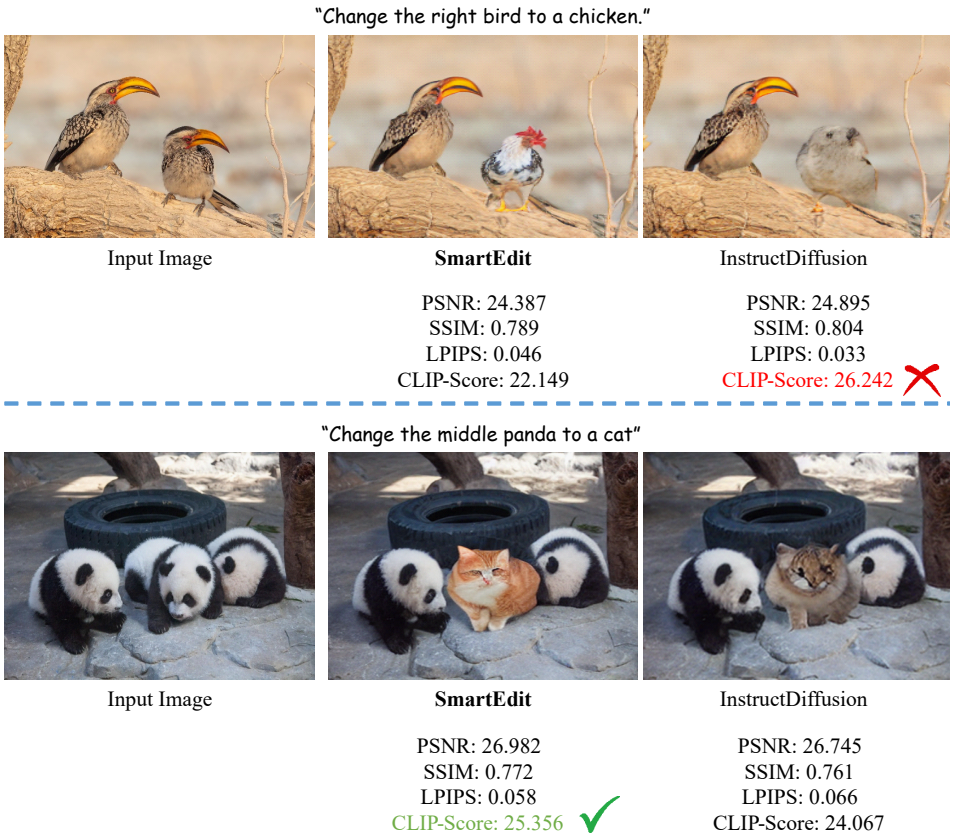

Figure 4. The evaluation of the outputs generated by SmartEdit and InstructDiffusion.

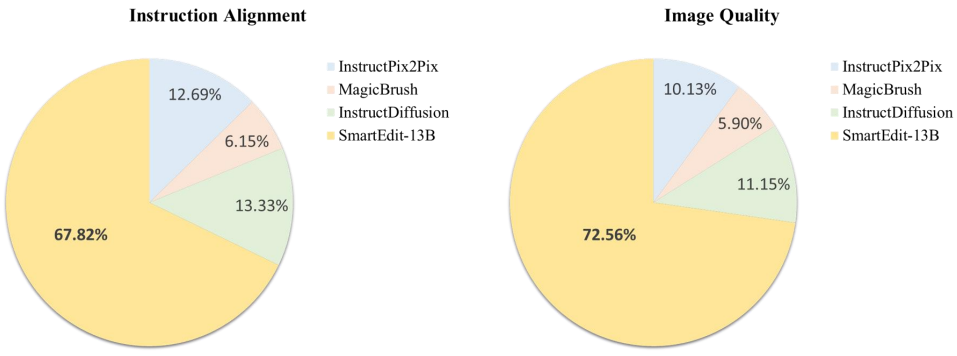

Figure 5. The results of user studies, comparing the results generated by InstructPix2Pix, MagicBrush, InstructDiffusion, and SmartEdit-13B. Based on the results from both the Instruction Alignment and Image Quality perspectives, SmartEdit demonstrates superior effectiveness.

5. More Visual Results on Reason-Edit

For complex understanding scenarios, we show more editing results of SmartEdit in Fig. 6. For the various object attributes, SmartEdit can understand the image and instructions well and can correctly edit the specified object accordingly. In addition, we compare the qualitative results of different methods for complex understanding scenarios, as shown in Fig. 7. From the first and second rows, it can be seen that InstructDiffusion can also edit specified objects according to instructions, but the quality of its edited images is much worse than that of SmartEdit. For the middle two rows of images, only MagicBrush among the existing methods understands the instructions and makes some modifications, but the image quality after editing is poor. For the last two rows of images, existing methods struggle to understand the instructions. SmartEdit, on the other hand, exhibits a superior ability to accomplish this task.

For reasoning scenarios, we provide a qualitative comparison of different methods on Reason-Edit, as shown in Fig. 8. In the first row, although MagicBrush and InstructDiffusion can remove the fork, the part of the cake in the original image also gets modified accordingly. In contrast, SmartEdit not only removes the fork but also effectively protects other areas from being modified. For the second row, other methods do not find the food with the most vitamins (i.e., orange), but SmartEdit successfully identifies the orange and replaces it with an apple. From the third to the sixth rows, SmartEdit can understand the instructions and reason out the objects that need to be edited while keeping other areas unchanged. However, other methods struggle with understanding complex instructions and identifying the corresponding objects, leading to a poor editing effect. In summary, even though the existing methods use the same training data as SmartEdit for fine-tuning, the introduction of LLaVA and BIM modules enables the model to comprehend more complex instructions, thus yielding superior results.

Figure 6. Visual effects of SmartEdit on Reason-Edit dataset (mainly on the complex understanding scenarios). It can be seen that for complex understanding scenarios (the instruction that contains various object attributes like location, relative size, color, and in or outside the mirror), SmartEdit has good instruction-based editing effects.

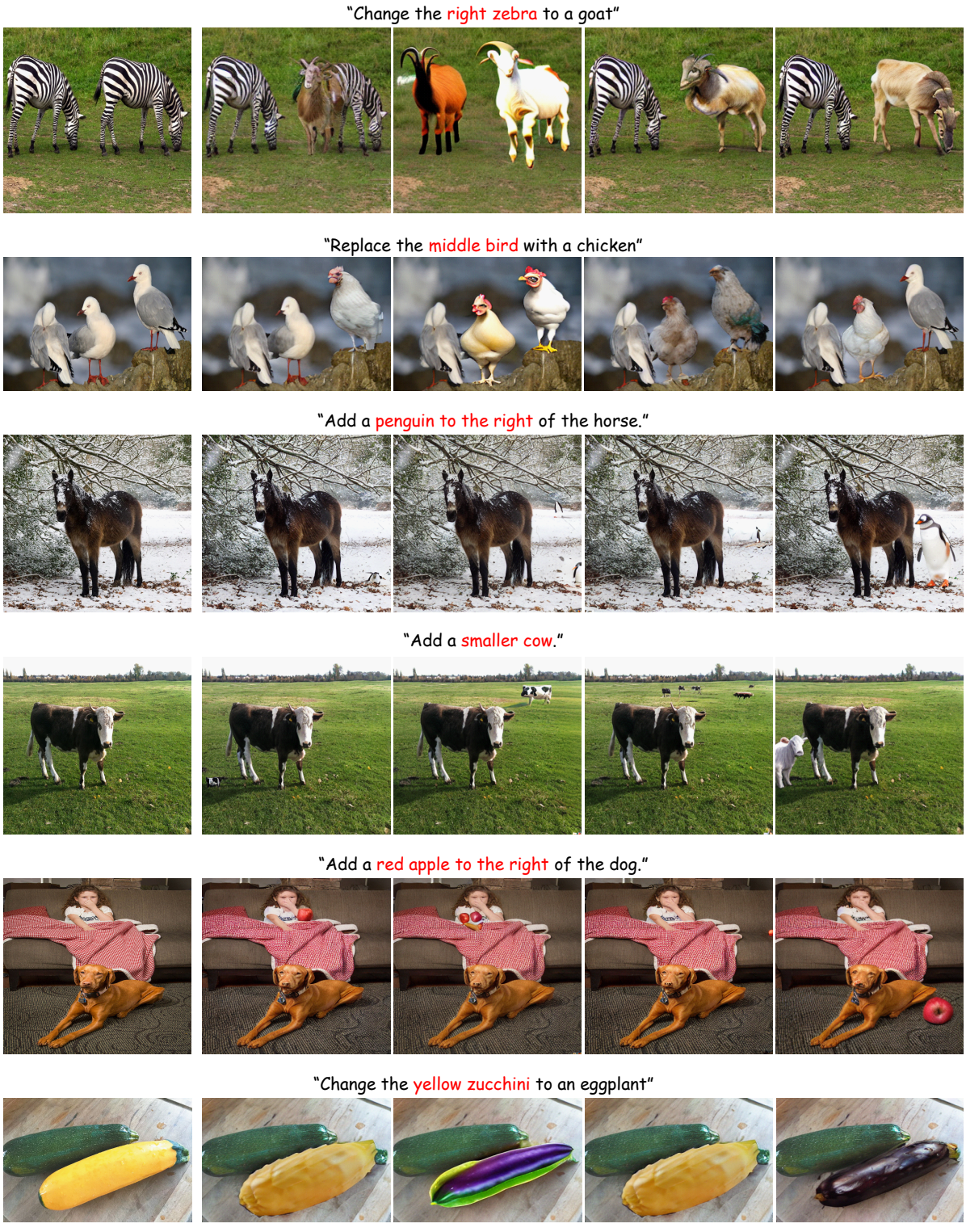

Figure 7. Qualitative comparison on Reason-Edit dataset (mainly on the complex understanding scenarios). Compared to other methods, SmartEdit can precisely edit specific objects in images according to instructions, while keeping the content in other areas unchanged.

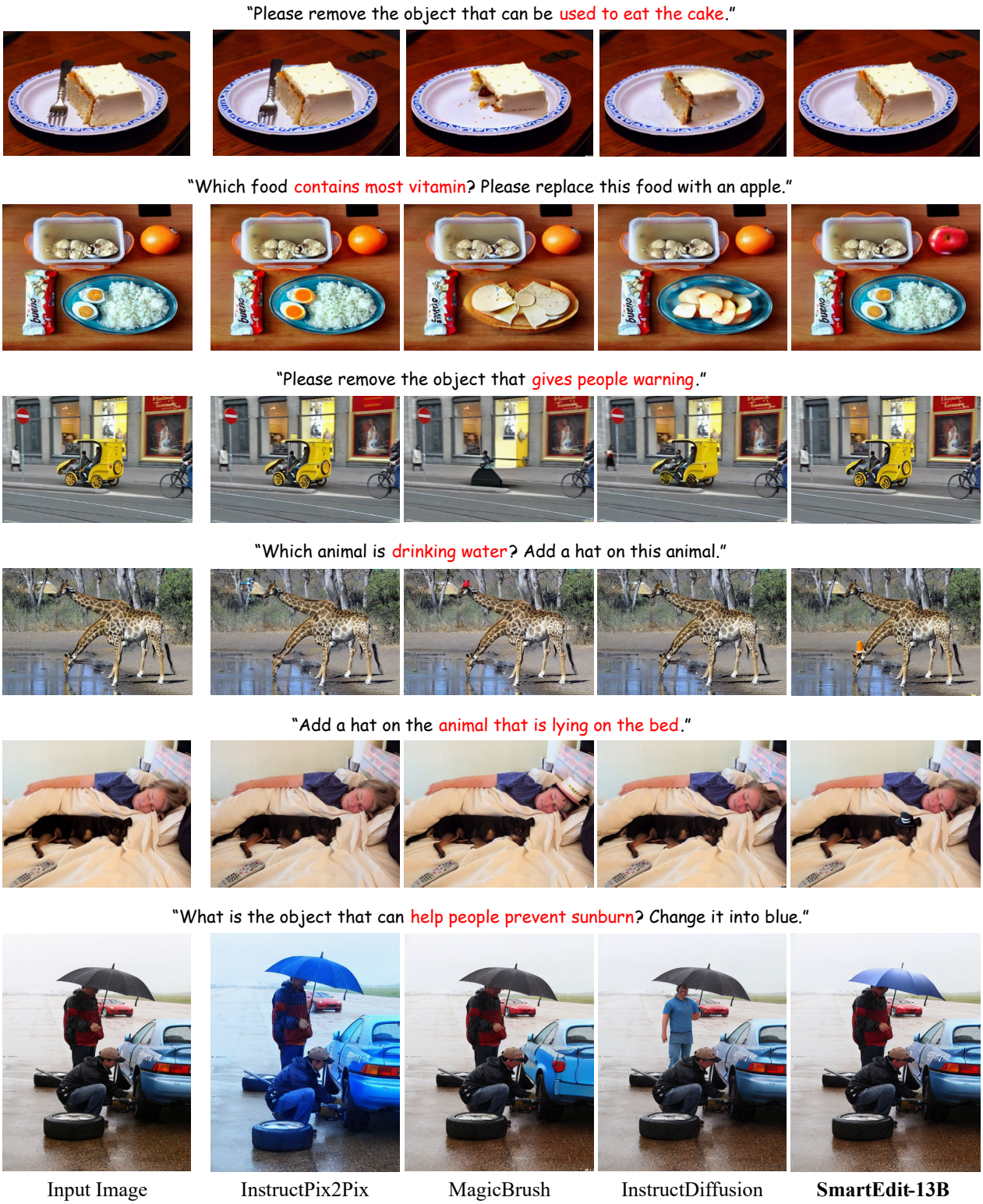

Figure 8. Qualitative comparison on Reason-Edit dataset (mainly on the complex reasoning scenarios). For reasoning scenarios, SmartEdit can effectively utilize the reasoning capabilities of the LLM to identify the corresponding objects, and then edit the objects according to the instructions. Other methods perform poorly in these scenarios.

6. Results of SmartEdit and Other Methods on MagicBrush

In Fig. 9, we demonstrate the performance of SmartEdit on the MagicBrush [20] test dataset. The first 2 rows are the editing results for single-turn, the middle 2 rows are for two-turn, and the last row is for three-turn. These results indicate that SmartEdit also has good editing effects on the MagicBrush test dataset, not only for single-turn, but also for multi-turn.

We further compare SmartEdit with other methods such as InstructPix2Pix [1], MagicBrush [20], and InstructDiffusion [5] on the MagicBrush test dataset. The quantitative results are presented in Tab. 2. It's important to note that MagicBrush releases two distinct checkpoints, MagicBrush-52[1] (trained for 52 epochs) and MagicBrush-168[2] (trained for 168 epochs). In the main paper of MagicBrush, the author utilizes MagicBrush-52 for qualitative results, while MagicBrush-168 is designed for quantitative results. As shown in Tab. 2, MagicBrush-168 significantly outperforms MagicBrush-52 and other methods, including SmartEdit, in terms of metrics. However, upon further analysis of these metrics (as shown in Fig. 10), we find that the L_1, CLIP-I, and DINO-I metrics may not be reliable. For instance, in the first set of images, SmartEdit effectively replaces the animal stickers with a smiley face sticker, while MagicBrush-168 adds multiple face stickers without completely removing the original animal stickers. Visually, SmartEdit's results appear superior to those of MagicBrush-168. A similar pattern is observed in the second set of images where SmartEdit successfully changes the hats of the two men in the original image to white, whereas MagicBrush-168 shows minimal changes. Despite this, the L_1, CLIP-I, and DINO-I metrics indicate that MagicBrush-168's results are significantly better than SmartEdit's, suggesting that these metrics may not be a reliable measure of performance. In contrast, the CLIP-T metric seems to align more closely with the actual editing results, making it a potentially more reliable performance indicator. From Tab. 2, it can be seen that SmartEdit performs better than MagicBrush-168 on the CLIP-T metric, while it is comparable to the results of MagicBrush-52.

The comparative analysis of the qualitative results is illustrated in Fig. 11. InstructPix2Pix, which has not been trained on the MagicBrush dataset, demonstrates subpar performance. MagicBrush-168, in most cases, either tends to retain the original image (as seen in the first, second, third, and fifth rows) or exhibits poor editing results (as evident in the fourth and sixth rows). Although MagicBrush-52 shows better results than MagicBrush-168, the results after editing do not correspond well with the instructions (notably in the second and fourth rows). InstructDiffusion sometimes generates artifacts, as observed in the fourth and fifth rows. In contrast, SmartEdit effectively adheres to the instructions, showcasing superior results.

Methods	$L_1 \downarrow$	CLIP-I $\uparrow$	CLIP-T $\uparrow$	DINO-I $\uparrow$
InstructPix2Pix	0.113	0.854	0.292	0.698
MagicBrush-52	0.076	0.907	0.306	0.806
MagicBrush-168	0.062	0.934	0.302	0.868
InstructDiffusion	0.097	0.892	0.302	0.777
SmartEdit-7B	0.089	0.904	0.303	0.797
SmartEdit-13B	0.081	0.914	0.305	0.815

Table 2. Quantitative comparison (L_1/CLIP-I/CLIP-T/DINO-I) on the MagicBrush test set.

7. Difference between SmartEdit, MGIE and InstructDiffusion

Recently, we have noticed a concurrent work: MGIE [4]. This method mainly uses MLLMs (i.e., LLaVA) to generate expressive instructions and provides explicit guidance for the following diffusion model. Compared with MGIE, there are three main differences. First, SmartEdit primarily targets complex understanding and reasoning scenarios, which are rarely mentioned in the MGIE paper. Secondly, in terms of network structure, we propose a Bidirectional Interaction Module (BIM) that enables comprehensive bidirectional information interactions between the image and the LLM output. Thirdly, we explore how to enhance the perception and reasoning capabilities of SmartEdit and propose a synthetic editing dataset. From both quantitative and qualitative results, it can be demonstrated that Our Smart has the ability to handle complex understanding and reasoning scenarios.

Compared with InstructDiffusion, which proposes a unifying and generic framework for aligning computer vision tasks with human instructions, our primary focus is the field of instruction-based image editing. In our experiments, we find that the perceptual ability of the diffusion model is crucial for instruction editing methods. Since InstructDiffusion also trains on the segmentation dataset, for convenience, we directly use its weights as the initial weights for the diffusion model in SmartEdit. However, as can be seen from Fig. 7 and Fig. 8, despite InstructDiffusion utilizing a large amount of perception datasets for joint training, its performance in complex understanding and reasoning scenarios is somewhat standard. By integrating LLaVA and BIM module, and supplementing the training data with segmentation data and synthetic editing data, the final SmartEdit can achieve satisfactory results in complex understanding and reasoning scenarios.

[1] https://huggingface.co/vinesmsuic/magicbrush-jul7
[2] https://huggingface.co/vinesmsuic/magicbrush-paper

Figure 9. The performance of SmartEdit on the MagicBrush test dataset. SmartEdit has good editing effects on the MagicBrush test dataset, not only for single-turn but also for multi-turn.

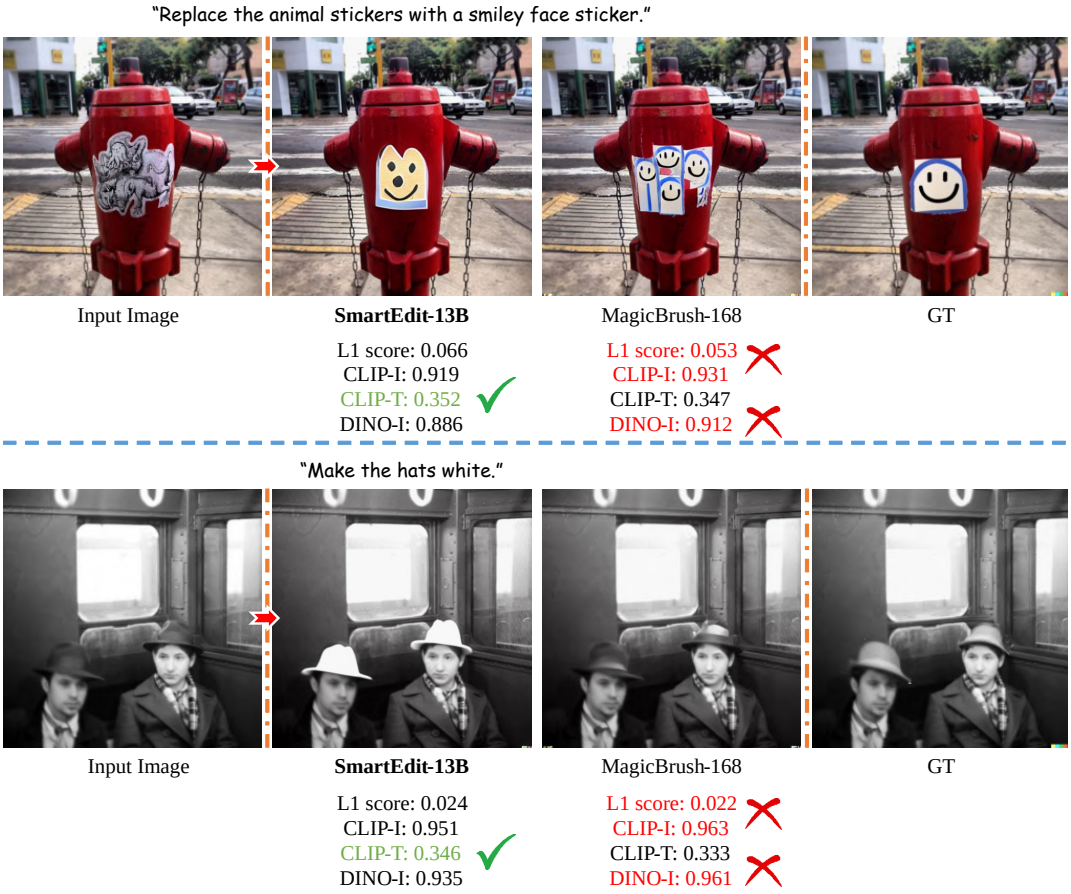

Figure 10. The evaluation of the outputs generated by SmartEdit and MagicBrush-168. Here we adopt these four metrics: L_1, CLIP-I, CLIP-T, and DINO-I metrics. The results indicate that SmartEdit performs better than MagicBrush-168. However, it's important to note that the L_1, CLIP-I, and DINO-I metrics may not correspond well with these results.

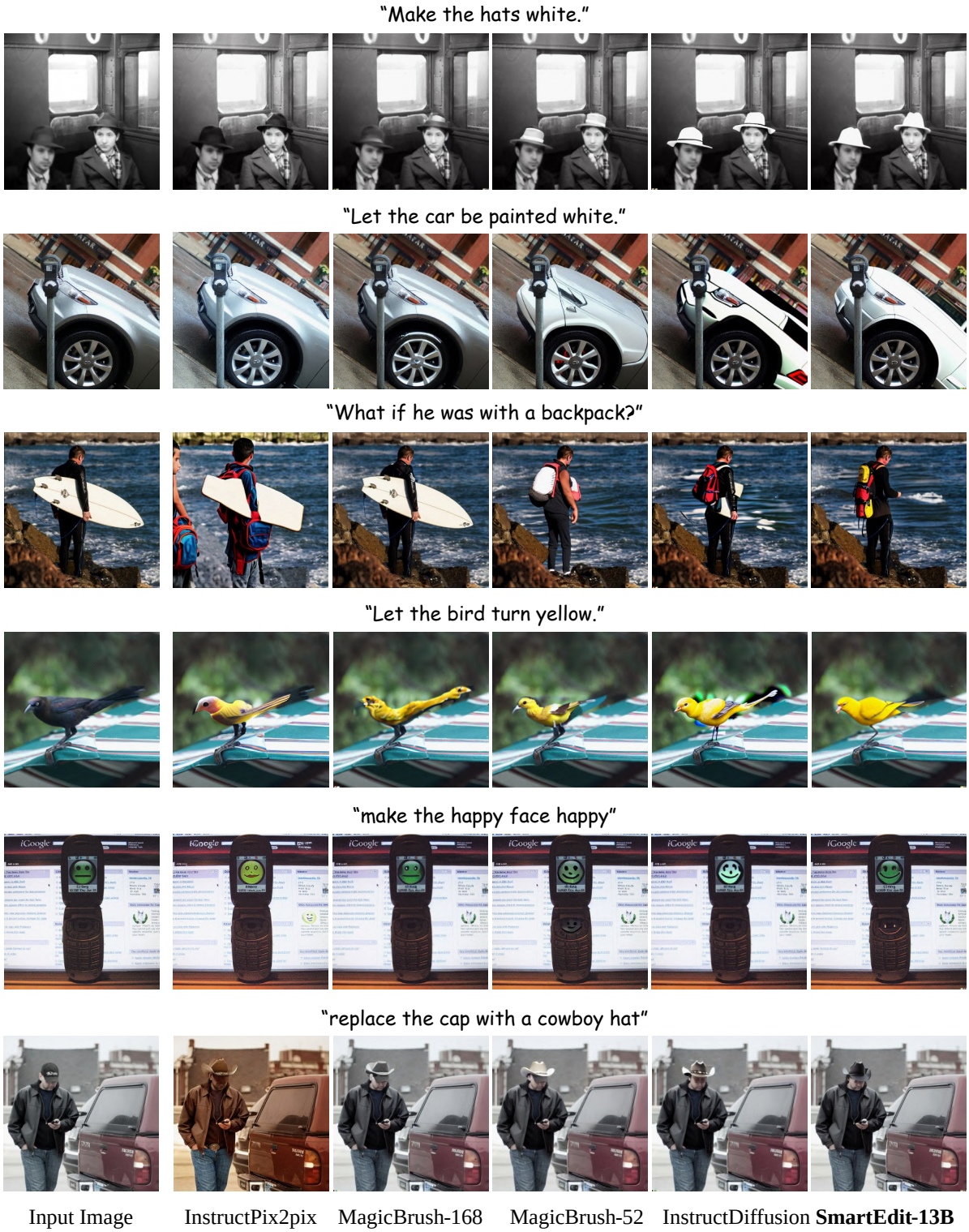

Figure 11. Qualitative comparison between our SmartEdit, MagicBrush-168, MagicBrush-52, InstructDiffusion, and InstructPix2Pix. Compared against other methods, SmartEdit effectively adheres to the instructions, showcasing superior results.

References

[1] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023. 10

[2] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018. 1

[3] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3558–3568, 2021. 1

[4] Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Yang Wang, Yinfei Yang, and Zhe Gan. Guiding instruction-based image editing via multimodal large language models. *arXiv preprint arXiv:2309.17102*, 2023. 1, 10

[5] Zigang Geng, Binxin Yang, Tiankai Hang, Chen Li, Shuyang Gu, Ting Zhang, Jianmin Bao, Zheng Zhang, Han Hu, Dong Chen, et al. Instructdiffusion: A generalist modeling interface for vision tasks. *arXiv preprint arXiv:2309.03895*, 2023. 1, 10

[6] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE, 2010. 4

[7] 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. 1

[8] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 1

[9] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*, 2023. 1

[10] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 1

[11] Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. Mat: Mask-aware transformer for large hole image inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10758–10768, 2022. 1

[12] Chang Liu, Henghui Ding, and Xudong Jiang. Gres: Generalized referring expression segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23592–23601, 2023. 1

[13] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 1

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[15] 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*, pages 8748–8763. PMLR, 2021. 1, 4

[16] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1

[17] 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. 1

[18] Ke Wang, Michaël Gharbi, He Zhang, Zhihao Xia, and Eli Shechtman. Semi-supervised parametric real-world image harmonization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1

[19] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 69–85. Springer, 2016. 1

[20] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *arXiv preprint arXiv:2306.10012*, 2023. 10

[21] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 4