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

Yuzhou Huang[*1,2#] Liangbin Xie[*2,3,5#] Xintao Wang[2,4†] Ziyang Yuan[2,8#] Xiaodong Cun[4]
Yixiao Ge[2,4] Jiantao Zhou[3] Chao Dong[5,7] Rui Huang[6] Ruimao Zhang[1†] Ying Shan[2,4]

[1]School of Data Science, Shenzhen Research Institute of Big Data, The Chinese University of Hong Kong, Shenzhen, China
[2]**ARC Lab, Tencent PCG** [3]University of Macau [4]Tencent AI Lab
[5]Shenzhen Institute of Advanced Technology [6]School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, China
[7]Shanghai Artificial Intelligence Laboratory [8]Tsinghua University

https://github.com/TencentARC/SmartEdit

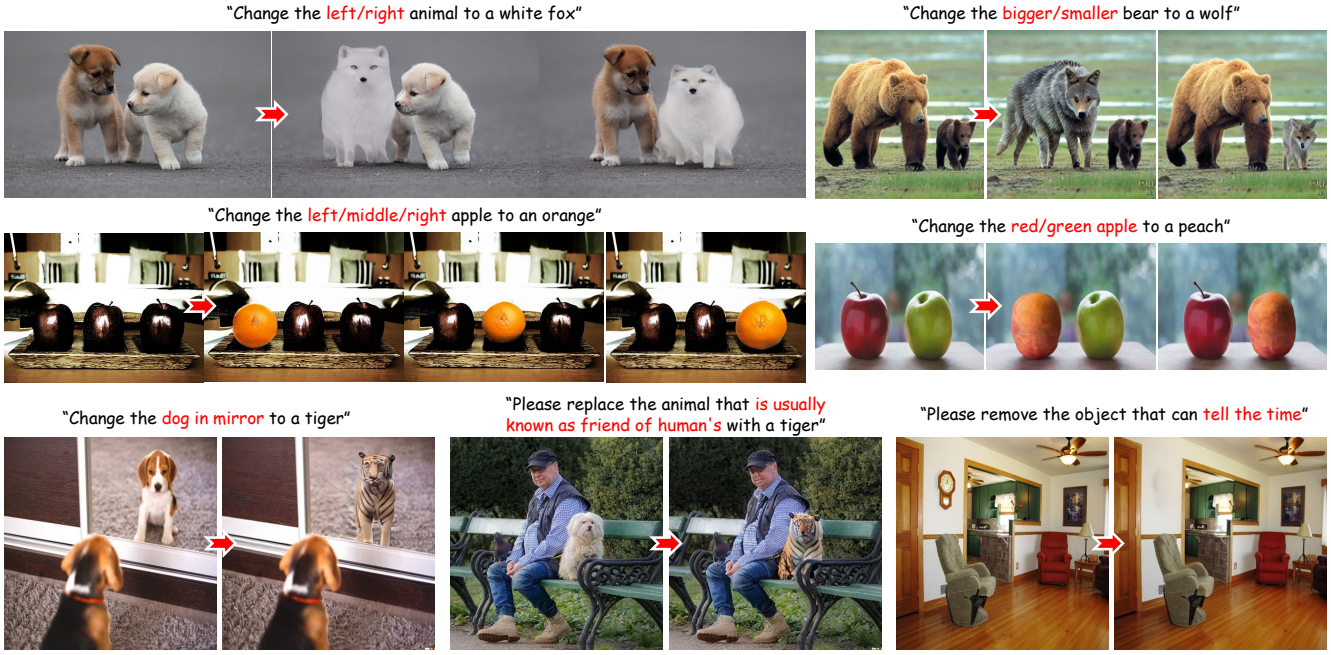

Figure 1. We propose SmartEdit, an instruction-based image editing model that leverages Multimodal Large Language Models (MLLMs) to enhance the understanding and reasoning capabilities of instruction-based editing methods. With the specialized design, our SmartEdit is capable of handling complex understanding (the instructions that contain various object attributes like location, relative size, color, and in or outside the mirror) and reasoning scenarios.

Abstract

Current instruction-based editing methods, such as InstructPix2Pix, often fail to produce satisfactory results in complex scenarios due to their dependence on the simple CLIP text encoder in diffusion models. To rectify this, this paper introduces SmartEdit, a novel approach to instruction-based image editing that leverages Multimodal Large Language Models (MLLMs) to enhance their understanding and reasoning capabilities. However, direct integration of these elements still faces challenges in situations requiring complex reasoning. To mitigate this, we propose a Bidirectional Interaction Module that enables comprehensive bidirectional information interactions between the input image and the MLLM output. During training, we initially incorporate perception data to boost the perception and understanding capabilities of diffusion models. Subsequently, we demonstrate that a small amount of complex

* Equal contribution † Corresponding author # Interns in ARC Lab, Tencent PCG

instruction editing data can effectively stimulate SmartEdit's editing capabilities for more complex instructions. We further construct a new evaluation dataset, Reason-Edit, specifically tailored for complex instruction-based image editing. Both quantitative and qualitative results on this evaluation dataset indicate that our SmartEdit surpasses previous methods, paving the way for the practical application of complex instruction-based image editing.

1. Introduction

Text-to-image synthesis [8, 14, 27, 29, 30, 32] has experienced significant advancements in recent years, thanks to the development of diffusion models. These methods have enabled the generation of images that are not only consistent with natural language descriptions but also align with human perception and preferences, marking a substantial leap forward in the field. Instruction-based image editing methods [1, 40], represented by InstructPix2Pix, leverage pre-trained text-to-image diffusion models as priors. This allows users to conveniently and effortlessly modify images through natural language instructions for ordinary users.

While existing instruction-based image editing methods can handle simple instructions effectively, they often fall short when dealing with complex scenarios, which require the model to have a more powerful understanding and reasoning capabilities. As depicted in Fig. 1, there are two common types of complex scenarios. The first is when the original image contains multiple objects, and the instruction modifies only one of these objects through certain attributes (such as *location, relative size, color, in or outside the mirror*). The other is when world knowledge is needed to identify the object to be edited (such as *the object that can tell the time*). We define these two types as complex understanding scenarios and complex reasoning scenarios, respectively. Handling these two scenarios is crucial for practical instruction editing, but existing instruction-based image editing methods probably fail in these scenarios (as shown in Fig. 2). In this paper, we attempt to identify the reasons why existing instruction-based image editing methods fail in these scenarios, and try to tackle the challenge in these scenarios.

The first reason why existing instruction-based image editing methods fail in these scenarios is that they typically rely on a simple CLIP text encoder [28] in diffusion models (*e.g.*, Stable Diffusion) to process the instructions. Under this circumstance, these models struggle to 1) understand and reason the instructions, and 2) integrate the image to comprehend the instructions. To address these limitations, we introduce the Multimodal Large Language Model (MLLM) (*e.g.*, LLaVA) [25, 42] into instruction-based editing models. Our method, SmartEdit, jointly optimizes MLLMs and diffusion models, leveraging the powerful reasoning capabilities of MLLMs to facilitate instruction-based image editing task.

While substituting the CLIP encoder in the diffusion model with MLLMs can alleviate some problems, this approach still falls short when it comes to examples that necessitate complex reasoning. This is because the input image to edit (original image) is integrated into the UNet of the Stable Diffusion model through a straightforward concatenation, which is further interacted with MLLM outputs through a cross-attention operation. In this setup, the image feature serves as the query, and MLLM outputs act as the key and value. This means that the MLLM outputs *unilaterally* modulate and interact with the image feature, which affects the results. To alleviate this issue, we further propose a Bidirectional Interaction Module (BIM). This module reuses the image information extracted by the LLaVA's visual encoder from the input image. It also facilitates a comprehensive bidirectional information interaction between this image and the MLLM output, enabling the model to perform better in complex scenarios.

即使把CLIP編碼器換成了MLLM，在處理需要複雜推理的情況下，效果仍然不夠好。

原因在於，圖像和MLLM的輸出之間的交互方式是單向的，即MLLM的輸出只影響圖像特徵，而沒有進行雙向的信息交換。

The second reason contributing to the failure of existing instruction-based editing methods is the absence of specific data. When solely training on editing datasets, such as the datasets used in Instructpix2pix and MagicBrush, SmartEdit also struggles to handle scenarios requiring complex reasoning and understanding. This is because SmartEdit has not been exposed to data from these scenarios. One straightforward approach is to generate a substantial amount of paired data similar to those scenarios. However, this method is excessively expensive because the cost of generating data for these scenarios is high. In this paper, we find that there are two keys to enhance SmartEdit's ability to handle complex scenarios. The first is to enhance the perception capabilities of UNet [31], and the second is to stimulate the model capacity in those scenarios with a few high-quality examples. Correspondingly, we 1) incorporate the perception-related data (*e.g.*, segmentation) into the model's training. 2) synthesize a few high-quality paired data with complex instructions to fine-tune our SmartEdit (similar to LISA [21]). In this way, SmartEdit not only reduces the reliance on paired data under complex scenarios but also effectively stimulates its ability to handle these scenarios.

Equipped with the model designs and the data utilization strategy, SmartEdit can understand complex instructions, surpassing the scope that previous instruction editing methods can do. To better evaluate the understanding and reasoning ability of instruction-based image editing methods, we collect the Reason-Edit dataset, which contains a total of 219 image-text pairs. Note that there is no overlap between the Reason-Edit dataset and the synthesized training data pairs. Based on the Reason-Edit dataset, we evaluate existing instruction-based image editing methods comprehensively. Both the quantitative and qualitative results

Figure 2. For more complex instructions or scenarios, InstructPix2Pix fails to follow the instructions.

on the Reason-Edit dataset indicate that SmartEdit significantly outperforms previous instruction-based image editing methods.

In summary, our contributions are as follows:

1. We analyze and focus on the performance of instruction-based image editing methods in more complex instructions. These complex scenarios have often been overlooked and less explored in past research.
2. We leverage MLLMs to better comprehend instructions. To further improve the performance, we propose a Bidirectional Interaction Module to facilitate the interaction of information between text and image features.
3. We propose a new dataset utilization strategy to enhance the performance of SmartEdit in complex scenarios. In addition to using conventional editing data, we introduce perception-related data to strengthen the perceptual ability of UNet in the diffusion process. Besides, we also add a small amount of synthetic editing data to further stimulate the model's reasoning ability.
4. An evaluation dataset, Reason-Edit, is specifically collected for evaluating the performance of instruction-based image editing tasks in complex scenarios. Both qualitative and quantitative results on Reason-Edit demonstrate the superiority of SmartEdit.

2. Related Work

2.1. Image Editing with Diffusion Models.

Pretrained text-to-image diffusion models [8, 14, 27, 29, 30, 32] can strongly assist image editing task. Instruction-based image editing task [1, 4, 12, 13, 17, 18, 35, 40] requires users to provide an instruction, which converts the original image to a newly designed image that matches the given instruction. Some methods can achieve this by utilizing a tuning-free approach. For example, Prompt-to-Prompt [13] suggests modifying the cross-attention maps by comparing the original input caption with the revised caption. MasaCtrl [4] converts existing self-attention in diffusion models into mutual self-attention, which can help query correlated local contents and textures from source images for consistency. In addition, due to the scarcity of paired image-instruction editing datasets, the pioneering work InstructPix2Pix [1] introduces a large-scale vision-language image editing datasets created by fine-tuned GPT-3 [2] and Prompt-to-Prompt with stable diffusion, and further fine-tunes the UNet [31], which can edit images by providing a simple instruction. To enhance the editing effect of InstructPix2Pix on real images, MagicBrush [40] further provides a large-scale and manually annotated dataset for instruction-guided real image editing.

The recent work, InstructDiffusion [12], also adopts the network design of InstructPix2Pix and focuses on unifying vision tasks in a joint training manner. By taking advantage of multiple different datasets, it can handle a variety of vision tasks, including understanding tasks (such as segmentation and keypoint detection) and generative tasks (such as editing and enhancement). Compared with InstructDiffusion, our primary focus is on the field of instruction-based image editing, especially for complex understanding and reasoning scenarios. In these scenarios, InstructDiffusion typically generates inferior results.

2.2. LLM with Diffusion Models

The exceptional open-sourced LLaMA [7, 34] significantly enhances the performance of vision tasks with the aid of Large Language Models (LLMs). Pioneering works such as LLaVA and MiniGPT-4 [25, 42] have improved image-text alignment through instruction-tuning. While numerous MLLM-based studies have demonstrated their robust capabilities across a variety of tasks, primarily those reliant on text generation (e.g., human-robot interaction, complex reasoning, science question answering, etc.), GILL [20] serves as a bridge between MLLMs and diffusion models. It learns to process images with LLMs and is capable of generating coherent images based on the input texts. SEED [10] presents an innovative image tokenizer to enable LLM to process and generate images and text concurrently. SEED-2 [11] further refines the tokenizer by aligning the generation embedding with the image embedding of unCLIP-SD, which allows for better preservation of rich visual semantics and reconstruction of more realistic images. Emu [33] can be characterized as a multimodal generalist, trained with the next-token-prediction objective. CM3Leon [39] proposes a multi-modal language model that is capable of executing text-to-image and image-to-text generation. It employs the CM3 multi-modal architecture that is fine-tuned on di-

verse instruction-style data, and utilizes a training method adapted from text-only language models.

3. Preliminary

The goal of instruction-based image editing is to make specific modifications to an input image x based on instructions c_T, resulting in the target image y. InstructPix2Pix, which is based on latent diffusion, is a seminal work in this field. For the target image y and an encoder $\mathcal{E}$, the diffusion process introduces noise to the encoded latent $z = \mathcal{E}(y)$, resulting in a noisy latent z_t, with the noise level increasing over timesteps $t \in T$. A UNet ϵ_δ is then trained to predict the noise added to the noisy latent z_t, given the image condition c_x and text instruction condition c_T, where $c_x = \mathcal{E}(x)$. The image condition is incorporated by directly concatenating c_x and z_t. The specific objective of latent diffusion is as follows:

$$L_{\text{diffusion}} = \mathbb{E}_{\mathcal{E}(y),\mathcal{E}(x),c_T,\epsilon\sim\mathcal{N}(0,1),t}[\|\epsilon - \epsilon_\delta(t, \text{concat}[z_t, \mathcal{E}(x)], c_T))\|_2^2 \quad (1)$$

where ϵ is the unscaled noise, t is the sampling step, z_t is latent noise at step t, $\mathcal{E}(x)$ is the image condition, and c_T is the text instruction condition. The concat corresponds to the concatenation operation.

Although InstructPix2Pix has some effectiveness in instruction editing, its performance is limited when dealing with complex understanding and reasoning scenarios. To address this issue, we introduce a Multimodal Large Language Model (MLLM) into the network architecture and propose a Bidirectional Interaction Module (BIM) to implement bidirectional information interaction between the MLLM output and image information. In addition, we also explore the data utilization strategy and find that perception-related data and a small amount of complex editing data are crucial for enhancing model's performance. We provide detailed descriptions of these aspects in the next section.

4. Method

In this paper, we introduce SmartEdit, specifically designed to handle complex instruction editing scenarios. In this section, we first provide a detailed overview of the framework of SmartEdit (Section 4.1). Then, we delve into the Bidirectional Interaction Module (Section 4.2). In Section 4.3, we discuss how to enhance the perception and understanding capabilities of UNet in the diffusion model and stimulate the ability of Multimodal Large Language Model (MLLM) to handle complex scenarios. Finally, We introduce Reason-Edit, which is primarily used to evaluate the ability of instruction-based image editing methods toward complex scenarios. (Section 4.4).

4.1. The Framework of SmartEdit

Given an image x and instruction c, which is tokenized as $(s_1, ..., s_T)$, our goal is to obtain the target image y based on c. As shown in Fig 3, the image x is first processed by the image encoder and FC layer, resulting in $v_\mu(x)$. Then $v_\mu(x)$ is sent into the LLM along with the token embedding $(s_1, ..., s_T)$. The output of the LLM is discrete tokens, which cannot be used as the input for subsequent modules. Therefore, we take the hidden states corresponding to these discrete tokens as the input for the following modules. To jointly optimize LLaVA and the diffusion model, following GILL [20], we expand the original LLM vocabulary with r new tokens $[\text{IMG}_1]$,...,$[\text{IMG}_r]$ and append the r [IMG] tokens to the end of instruction c. To be specific, we incorporate a trainable matrix $\mathbf{E}$ into the embedding matrix of the LLM, which represents the r [IMG] token embeddings. Subsequently, we minimize the negative log-likelihood of generated r [IMG] tokens, conditioned on tokens that have been generated previously:

$$L_{\text{LLM}}(c) = -\sum_{i=1}^{r} \log p_{/\theta\cup\mathbf{E}/}([\text{IMG}_i] \mid v_\mu(x), s_1, ..., s_T, [\text{IMG}_1], \ldots, [\text{IMG}_{i-1}]) \quad (2)$$

The majority of parameters θ in the LLM are kept frozen and we utilize LoRA [16] to carry out efficient fine-tuning. We take the hidden states h corresponding to the r [IMG] tokens as the input for the next module.

Considering the discrepancy between the feature spaces of the hidden states in the LLM and the clip text encoder, we need to align the hidden states h to the clip text encoder space. Inspired by BLIP2 [22] and DETR [5], we adopt a QFormer Q_β with 6−layers of transformer [36] and n learnable queries, obtaining feature f. Subsequently, the image feature v output by the image encoder E_ϕ interacts with f through a bidirectional interaction module (BIM), resulting in f' and v'. The process mentioned above is represented as:

$$\begin{aligned} h &= \text{LLaVA}(x, c), \\ f &= Q_\beta(h), \\ v &= E_\phi(x), \\ f', v' &= \text{BIM}(f, v) \end{aligned} \quad (3)$$

For the diffusion model, following the design of Instructpix2pix, we concat the encoded image latent $\mathcal{E}(x)$ and noisy latent z_t. Unlike Instructpix2pix, we use f' as the key and value in UNet, and combine v' into the features before entering UNet in a residual manner.The specific process can be formulated as:

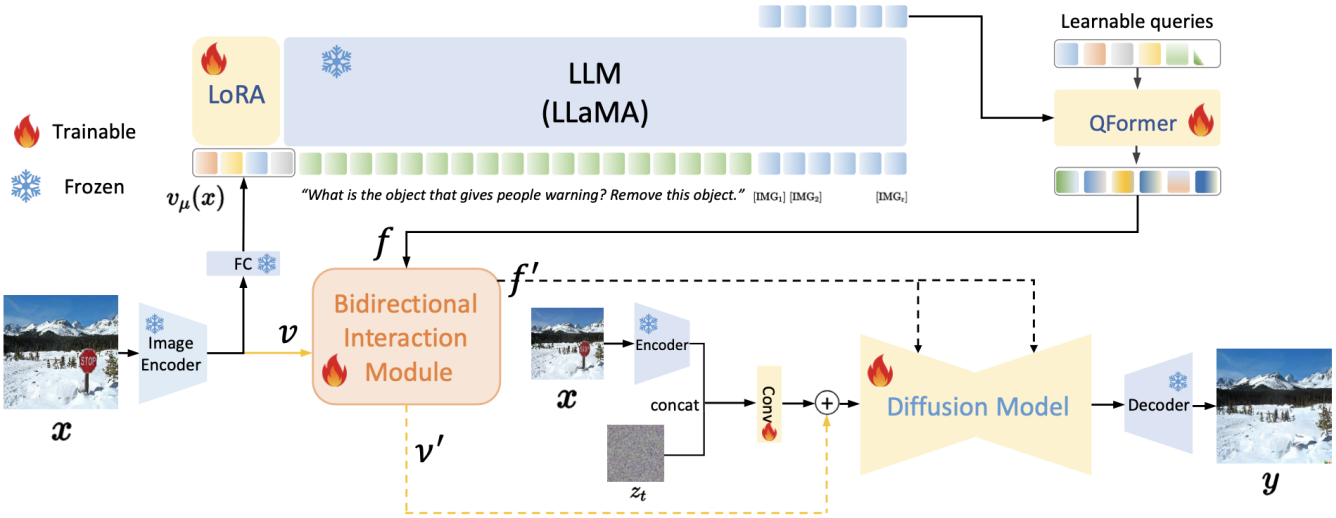

Figure 3. The overall framework of SmartEdit. For the instruction, we first append the r [IMG] tokens to the end of instruction c. Together with image x, they will be sent into LLaVA, which can then obtain the hidden states corresponding to these r [IMG] tokens. Then the hidden state is sent into the QFormer and gets feature f. Subsequently, the image feature v output by the image encoder E_ϕ interacts with f through a bidirectional interaction module (BIM), resulting in f' and v'. The f' and v' are input into the diffusion models to achieve the instruction-based image editing task.

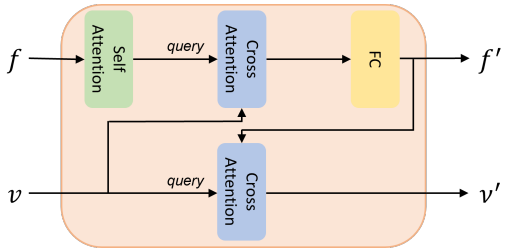

Figure 4. The network design of the BIM Module. In this module, the input information f and v will undergo bidirectional information interaction through different cross-attention.

$$L_{\text{diffusion}} = \mathbb{E}_{\mathcal{E}(y),\mathcal{E}(x),c_T,\epsilon\sim\mathcal{N}(0,1),t}[\|\epsilon - \epsilon_\delta(t, \text{concat}[z_t, \mathcal{E}(x)] + v', f'))\|_2^2 \quad (4)$$

To keep consistency with equation 1, we omit the Conv operation here.

4.2. Bidirectional Interaction Module

The design of BIM is depicted in Fig 4. It includes a self-attention block, two cross-attention blocks, and an MLP layer. The two inputs of BIM are the output f from QFormer and the output v from the image encoder. After bidirectional information interaction between f and v, BIM will eventually output f' and v'. In BIM, the process begins with f undergoing a self-attention mechanism. After this, f serves as a query to interact with the input v, which acts as both key and value, through a cross-attention block. This interaction results in the generation of f' via a point-wise MLP. Following the creation of f', it then serves as both key and value to interact with v, which now acts as a query. This second cross-attention interaction leads to the production of v'.

As discussed in the introduction, the proposed BIM module reuses the image feature and inputs it as supplementary information into UNet. The implementation of two cross-attention blocks in this module facilitates a robust bidirectional information interaction between the image feature and the text feature. Compared to not adopting the BIM module or only fusing the image feature and text feature in one direction, SmartEdit which is equipped with the BIM module yields better results. The experimental comparison of different designs is shown in Section 5.3.

4.3. Dataset Utilization Strategy

During the training process of SmartEdit, two primary challenges emerge when solely utilizing datasets gathered from InstructPix2Pix and MagicBrush as the training set. The first challenge is that SmartEdit has a poor perception of position and concept. The second challenge is that, despite being equipped with MLLM, SmartEdit still has limited capability in scenarios that require reasoning. In summary, the effectiveness of SmartEdit in handling complex scenarios is limited if it is only trained on conventional editing datasets. After analyses, we have identified the causes of these issues. The first issue stems from the UNet in the diffusion model which lacks an understanding of perception and concepts, leading to SmartEdit's poor perception of position and concept. The second issue is that SmartEdit has limited expo-

sure to editing data that requires reasoning abilities, which in turn limits its reasoning capabilities.

To tackle the first issue, we incorporate the segmentation data into the training set. Such modifications significantly enhanced the perception capabilities of the SmartEdit model. Regarding the second issue, we take inspiration from LISA [21] that a minimal amount of reasoning segmentation data can efficiently activate MLLM's reasoning capacity. Guided by this insight, we establish a data production pipeline and synthesize approximately 476 paired data (each sample contains an original image, instruction, and the synthetic target image) as a supplement to the training data. This synthetic editing dataset includes two major types of scenarios: complex understanding scenarios and reasoning scenarios. For complex understanding scenarios, the original image contains multiple objects and the corresponding instruction modifies the specific object based on various attributes (i.e., location, color, relative size, and in or outside the mirror). We specifically consider the mirror attribute because it is a typical example that requires a strong understanding of the scene (both inside and outside the mirror) to perform well. For reasoning scenarios, we involve complex reasoning cases that need world knowledge to identify the specific object. The effectiveness of this synthetic editing dataset and the impact of different datasets on the model's performance are detailed in Section 5.4. The details of the data production pipeline and some visual examples are described in the supplementary material.

4.4. Reason-Edit for Better Evaluation

To better evaluate existing instruction editing methods and SmartEdit's capabilities in complex understanding and reasoning scenarios, we collect an evaluation dataset, Reason-Edit. Reason-Edit consists of 219 image-text pairs. Consistent with the synthetic training data pairs, Reason-Edit is also categorized in the same manner. Note that there is no overlap between the data in Reason-Edit and the training set. With Reason-Edit, we can thoroughly test the performance of instruction-based image editing models in terms of understanding and reasoning scenarios. We hope more researchers will pay attention to the capabilities of instruction-based image editing models from these perspectives, thereby fostering the practical application of instruction-based image editing methods.

5. Experiments

5.1. Experimental Setting

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 [28] using the QFormer [22]. In the second stage, we optimize SmartEdit. To be specific, the weights of LLaVA are frozen and LoRA [16] 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 [31] in the diffusion model are fully optimized.

Network Architecture. For the Large Language Model with visual input (e.g., LLaVA), we choose LLaVA-1.1-7b and LLaVA-1.1-13b as the base model. During training, the weights of LLaVA are frozen and we add LoRA for efficient fine-tuning. In LoRA, the values of the two parameters, dim and alpha, are 16 and 27, respectively. We expand the original LLM vocabulary with 32 new tokens. The QFormer is composed of 6 transformer [36] layers and 77 learnable query tokens. In the BIM module, there is a self-attention block, two cross-attention blocks, and a Multilayer Perceptron (MLP) layer.

r = 32

Implementation Details. During the first stage of training, the AdamW optimizer [26] 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.

QFormer Image-Text aligned

LLaVa predict [IMG]

Training Datasets. In the first stage, we utilize the extensive corpus CC12M [6] as our primary data source. In the second stage, the training data can be divided into 4 categories: (1) segmentation datasets, which include COCOStuff [3], RefCOCO [38], GRefCOCO [24], and the reasoning segmentation dataset from LISA [21]; (2) editing datasets, which involve InstructPix2Pix and MagicBrush; (3) visual question answering (VQA) dataset, which is the LLaVA-Instruct-150k dataset [25]; (4) synthetic editing dataset, where we collect a total of 476 paired data for complex understanding and reasoning scenarios.

Evaluation Metrics. As we hope to only change the foreground of the image while keeping the background unchanged during the editing process, we adopt three metrics for the background area: PSNR, SSIM, and LPIPS [15, 41]. For the foreground area, we calculate the CLIP Score [28] between the foreground area of the edited image and the GT label. The GT label is annotated manually. Among these four metrics, except for LPIPS where lower is better, the other three metrics are higher the better. While these metrics can reflect the performance to a certain extent, they are not entirely accurate. To provide a more accurate evaluation of the effects of edited images, we propose a metric for assessing editing accuracy. Specifically, we hire four work-

ers 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).

5.2. Comparison with State-of-the-Art Methods

We compare SmartEdit with existing state-of-the-art instruction-based image editing methods, namely InstructPix2Pix, MagicBrush, and InstructDiffusion. Considering that these released models are trained on specific datasets, they would inevitably perform poorly if directly evaluated on Reason-Edit. To ensure a fair comparison, we fine-tune these methods on the same training set used by SmartEdit, and evaluate the fine-tuned models on Reason-Edit. The experimental results are shown in Tab 1. From the quantitative results of the Reasoning Scenarios in the table, it can be observed that when we replace the clip text encoder in the diffusion model with LLaVA and adopt the proposed BIM module, both SmartEdit-7B and SmartEdit-13B achieve better results on these five metrics. This suggests that in scenarios requiring reasoning from instructions, a simple clip text encoder may struggle to understand the meaning of the instructions. However, the MLLM can fully utilize its powerful reasoning ability and world knowledge to correctly identify the corresponding objects and perform edits.

The qualitative results further illustrate this point. As shown in Fig. 5, the first three examples are reasoning scenarios. In the first example, both SmartEdit-7B and SmartEdit-13B successfully identify the tool used for cutting fruit (knife) and remove it, while keeping the rest of the background unchanged. The second example can also be handled well by both of them. However, in the third example, we observe a difference in performance. Only SmartEdit-13B can accurately locate the object and perform the corresponding edits without altering other background areas. This suggests that in instruction-based image editing tasks that require reasoning, a more powerful MLLM model can effectively generalize its reasoning ability to this task. This observation aligns with the findings from LISA.

However, for understanding scenarios, we observe a difference in performance between SmartEdit-7B and SmartEdit-13B when compared to InstructDiffusion on the three metrics of PSNR/SSIM/LPIPS. Specifically, SmartEdit-7B performs worse than InstructDiffusion, while SmartEdit-13B outperforms InstructDiffusion on these metrics. Upon further analysis of the qualitative results, as shown in the 4_{th} and 5_{th} rows of Fig. 5, we find that from a visual perspective, both SmartEdit-7B and SmartEdit-13B appear superior to InstructDiffusion. This suggests that the three metrics do not always align with human visual perception. We confirm this phenomenon in the supplementary material (Section 8.1). From the result of the Ins-align metric, it can be observed that SmartEdit shows a significant improvement compared to previous instruction-based image editing methods. Also, when adopting a more powerful MLLM model, SmartEdit-13B performs better than SmartEdit-7B on the Ins-align metric.

5.3. Ablation Study on BIM

To validate the effectiveness of the bidirectional information interaction in our proposed BIM module, we conduct comparative experiments on the SmartEdit-7B model. The details are presented in Tab. 2. The first experiment, denoted as Exp 1, aims to verify the necessity of the information interaction proposed in the BIM module. In this experiment, we remove the BIM module from the SmartEdit-7B model and directly apply the text feature output from QFormer to the diffusion model. The second experiment, denoted as Exp 2, aims to verify the necessity of the bidirectional information interaction proposed in the BIM module. Specifically, all blocks are discarded except for the cross-attention block on the image feature branch. Therefore, the information from the text feature of QFormer is unidirectionally applied to the image feature. These two experiments are designed to test the impact of removing or altering the BIM module on the performance of SmartEdit-7B in complex understanding and reasoning scenarios. As shown in Tab. 2, if the BIM module is removed, there is a significant decline in all metrics for both understanding and reasoning scenarios. When the BIM module is replaced with the SimpleCA module, we observe a noticeable decline in all metrics, except for the clip score in understanding scenarios. Further comparison of the qualitative results in Fig. 6 confirms that the introduction of the BIM indeed enhances SmartEdit's instruction editing performance. To be specific, when we do not use the BIM module (i.e., plain), the dog bowl (first row) turns into other objects (marked with a red circle), and the fork (second row) does not change at all. After using SimpleCA, it can be found that the dog bowl and fork have been partially removed. When SmartEdit is equipped with BIM, the dog bowl and fork can be well removed.

5.4. Ablation Study on Dataset Usage

In Section 4.3, we explore an efficient strategy for data utilization, aiming to enhance SmartEdit's capabilities in handling complex understanding and reasoning scenarios. During the training process of SmartEdit, we employ the common editing dataset, segmentation dataset, and the synthetic editing dataset. To validate the significance of these different data types in boosting SmartEdit's performance, we conduct a series of ablation studies, as detailed in Tab. 3. These experiments are based on the SmartEdit-7B model. In Exp 1, we train the model using only the editing data. In

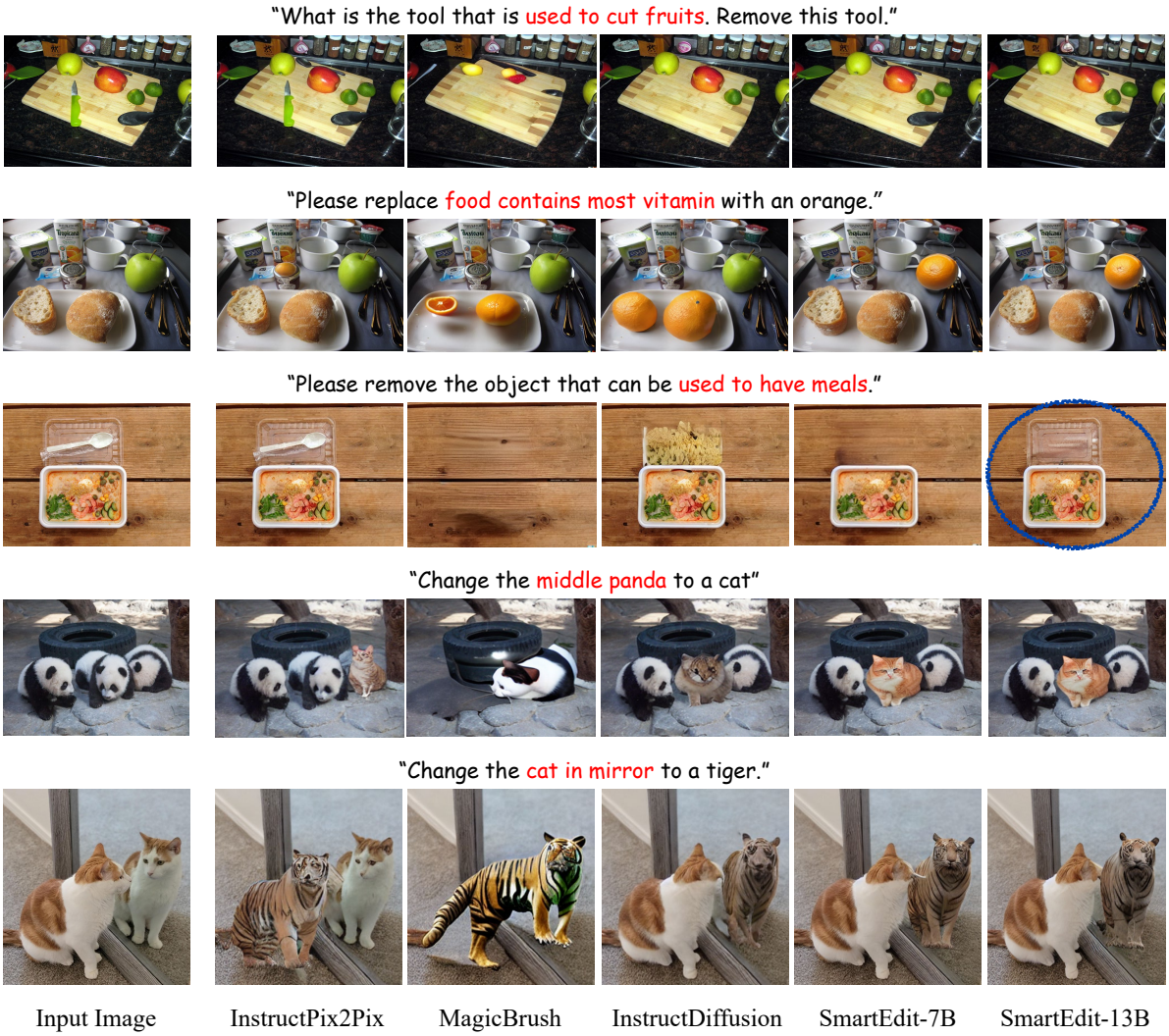

Figure 5. Qualitative comparison on Reason-Edit. When compared to several existing instruction-based image editing methods that have undergone further fine-tuning on our synthetic editing dataset, our approach demonstrates superior editing capabilities in complex scenarios.

Methods	Understanding Scenarios					Reasoning Scenarios				
	PSNR(dB)↑	SSIM↑	LPIPS↓	CLIP Score↑	**Ins-align↑**	PSNR(dB)	SSIM	LPIPS	CLIP Score	**Ins-align↑**
InstructPix2Pix	21.576	0.721	0.089	22.762	0.537	24.234	0.707	0.083	19.413	0.344
MagicBrush	18.120	0.68	0.143	22.620	0.290	22.101	0.694	0.113	19.755	0.283
InstructDiffusion	23.258	0.743	0.067	23.080	0.697	21.453	0.666	0.117	19.523	0.483
SmartEdit-7B	22.049	0.731	0.087	23.611	**0.712**	25.258	0.742	0.055	20.950	**0.789**
SmartEdit-13B	23.596	0.751	0.068	23.536	**0.771**	25.757	0.747	0.051	20.777	**0.817**

Table 1. Quantitative comparison (PSNR↑/SSIM↑/LPIPS↓/CLIP Score↑ (ViT-L/14)/Ins-align↑) on Reason-Edit. All the methods we compared have been fine-tuned using the same training data as that used by SmartEdit.

Exp 2, we incorporate segmentation data into the training process, building upon Exp 1. In Exp 3, we further add the synthetic editing data to the basis established in Exp 1. The quantitative results of these experiments reveal that segmentation data and synthetic editing data play complementary roles in enhancing the model's performance. This is further corroborated by the visual comparison in Fig. 7. For reasoning scenarios, when adopting only the editing dataset or combining the editing dataset and the segmentation dataset, the performance of SmartEdit is inferior. When the syn-

Exp ID	Plain	SimpleCA	BIM	Understanding Scenarios					Reasoning Scenarios				
				PSNR(dB)↑	SSIM↑	LPIPS↓	CLIP Score↑	**Ins-align↑**	PSNR(dB)	SSIM	LPIPS	CLIP Score	**Ins-align↑**
1	✓			20.975	0.713	0.108	23.36	0.695	23.848	0.725	0.074	20.33	0.694
2		✓		19.557	0.692	0.126	23.66	0.692	23.508	0.716	0.081	20.17	0.722
3			✓	22.049	0.731	0.087	23.61	**0.712**	25.258	0.742	0.055	20.95	**0.789**

Table 2. Quantitative comparison (PSNR↑/SSIM↑/LPIPS↓/CLIP Score↑ (ViT-L/14)/Ins-align↑) on Reason-Edit. These comparative experiments are conducted based on the SmartEdit-7B.

Exp ID	Edit	Segmentation	Synthetic editing dataset	Understanding Scenarios					Reasoning Scenarios				
				PSNR(dB)↑	SSIM↑	LPIPS↓	CLIP Score↑	**Ins-align↑**	PSNR(dB)	SSIM	LPIPS	CLIP Score	**Ins-align↑**
1	✓			17.568	0.664	0.171	22.79	0.201	22.400	0.706	0.102	19.22	0.233
2	✓	✓		18.960	0.690	0.143	22.83	0.361	21.774	0.693	0.116	19.82	0.311
3	✓		✓	19.562	0.702	0.111	22.32	0.440	23.595	0.715	0.079	20.43	0.567
4	✓	✓	✓	22.049	0.731	0.087	23.61	**0.712**	25.258	0.742	0.055	20.95	**0.789**

Table 3. Quantitative comparison (PSNR↑/SSIM↑/LPIPS↓/CLIP Score↑ (ViT-L/14)/Ins-align↑) on Reason-Edit. These comparative experiments are conducted based on the SmartEdit-7B.

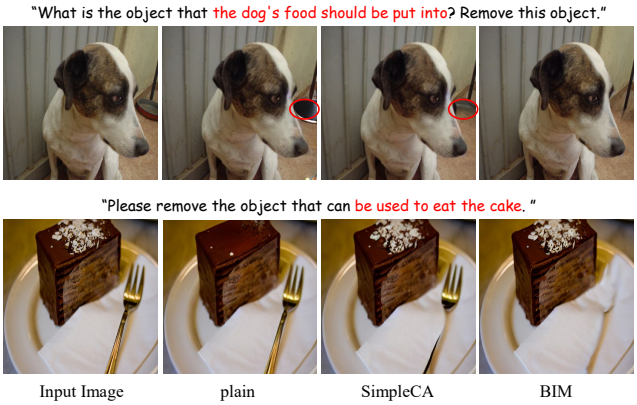

Figure 6. The effectiveness of the BIM Module.

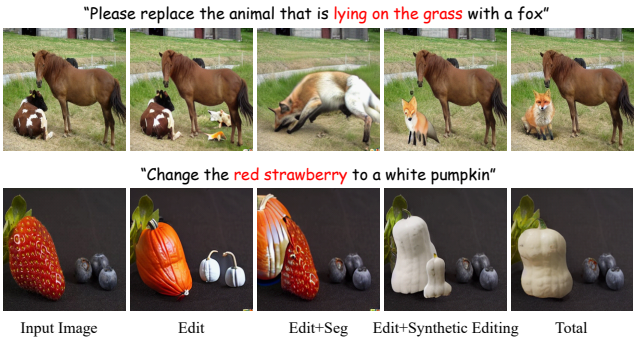

Figure 7. The significance of joint training with multiple datasets.

thetic editing data is incorporated into the editing dataset, SmartEdit can accurately locate the specific objects. However, the output of SmartEdit is also mediocre (the generated fox has obvious artifacts, and two pumpkins are generated). When all these datasets are combined as the training set, the results generated by SmartEdit have a further significant improvement in visual effects.

6. Conclusion

In conclusion, this paper presents SmartEdit, a novel approach to instruction-based image editing that enhances understanding and reasoning capabilities by incorporating the Large Language Models (LLMs) with visual inputs. By introducing the Bidirectional Interaction Module (BIM), we have overcome challenges associated with the direct integration of LLMs and diffusion models in complex reasoning scenarios. Our data utilization strategy, which incorporates perception data and complex instruction editing data, effectively enhances SmartEdit's capabilities in handling complex understanding and reasoning scenarios. Evaluation on our newly constructed dataset, Reason-Edit, shows that SmartEdit outperforms previous methods, marking a significant step towards practical applications of complex instruction-based image editing.

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. 2, 3, 9

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 3

[3] 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. 6

[4] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. *arXiv preprint arXiv:2304.08465*, 2023. 3

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-

end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 4

[6] 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. 6

[7] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. 3

[8] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 2, 3

[9] 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, 9

[10] Yuying Ge, Yixiao Ge, Ziyun Zeng, Xintao Wang, and Ying Shan. Planting a seed of vision in large language model. *arXiv preprint arXiv:2307.08041*, 2023. 3

[11] Yuying Ge, Sijie Zhao, Ziyun Zeng, Yixiao Ge, Chen Li, Xintao Wang, and Ying Shan. Making llama see and draw with seed tokenizer. *arXiv preprint arXiv:2310.01218*, 2023. 3

[12] 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. 3, 1, 9

[13] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 3

[14] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2, 3

[15] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE, 2010. 6, 3

[16] 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. 4, 6

[17] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Direct inversion: Boosting diffusion-based editing with 3 lines of code. *arXiv preprint arXiv:2304.04269*, 2023. 3

[18] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023. 3

[19] 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

[20] Jing Yu Koh, Daniel Fried, and Ruslan Salakhutdinov. Generating images with multimodal language models. *arXiv preprint arXiv:2305.17216*, 2023. 3, 4

[21] 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. 2, 6

[22] 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. 4, 6

[23] 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

[24] 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. 6

[25] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 2, 3, 6

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[27] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 2, 3

[28] 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. 2, 6, 3

[29] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 2, 3

[30] 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. 2, 3, 1

[31] 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, 3, 6

[32] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 2, 3

[33] Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative pretraining in multimodality. *arXiv preprint arXiv:2307.05222*, 2023. 3

[34] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 3

[35] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1921–1930, 2023. 3

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4, 6

[37] 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

[38] 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. 6

[39] Lili Yu, Bowen Shi, Ramakanth Pasunuru, Benjamin Muller, Olga Golovneva, Tianlu Wang, Arun Babu, Binh Tang, Brian Karrer, Shelly Sheynin, et al. Scaling autoregressive multimodal models: Pretraining and instruction tuning. *arXiv preprint arXiv:2309.02591*, 2023. 3

[40] 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. 2, 3, 9

[41] 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. 6, 3

[42] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 2, 3

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

Supplementary Material

In this supplementary file, we provide the following materials:

1. Details of the data production pipeline.
2. More quantitative comparisons on Reason-Edit.
3. More visual results on Reason-Edit.
4. Results of SmartEdit and other methods on MagicBrush.
5. Difference between SmartEdit, MGIE [9] and InstructDiffusion [12].

7. 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. 8. To be specific, We begin with two images, x_1 and x_2, collected from the internet. Using the SAM [19] 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 [23] 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 [37] 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. 9, 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 [19], then adopt stable diffusion [30] 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. 9, we illustrate some reasoning samples that are included in the training data.

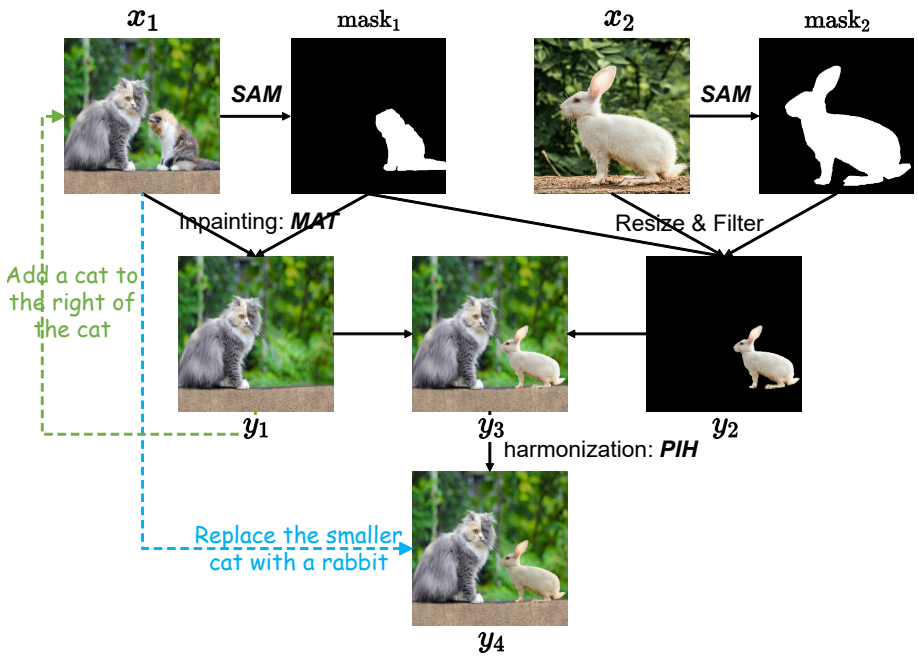

Figure 8. The data production pipeline of the synthetic paired training set (complex understanding scenarios). For x_1 and x_2, we first use SAM to generate mask_1 and mask_2. Then, we use MAT, combined with x_1 and mask_1, to get y_1. At the same time, by performing specific operations on mask_1, mask_2, 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 9. 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.

8. More Quantitative Results on Reason-Edit

8.1. 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 [28] between the foreground area of the edited image and the GT label. For the background area, we calculate the PSNR/SSIM/LPIPS [15, 41] 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. 10. 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 Tab. 1, Tab. 2, and Tab. 3. In Tab. 1, 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.

Tab. 2 and Tab. 3 present the results of the Ablation studies for BIM module and Dataset Usage, respectively. In Tab. 2, 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 Tab. 3, the joint utilization of editing data, segmentation data, and synthetic editing data enables SmartEdit to deliver better results in complex understanding and reasoning scenarios.

8.2. 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. 11. 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.

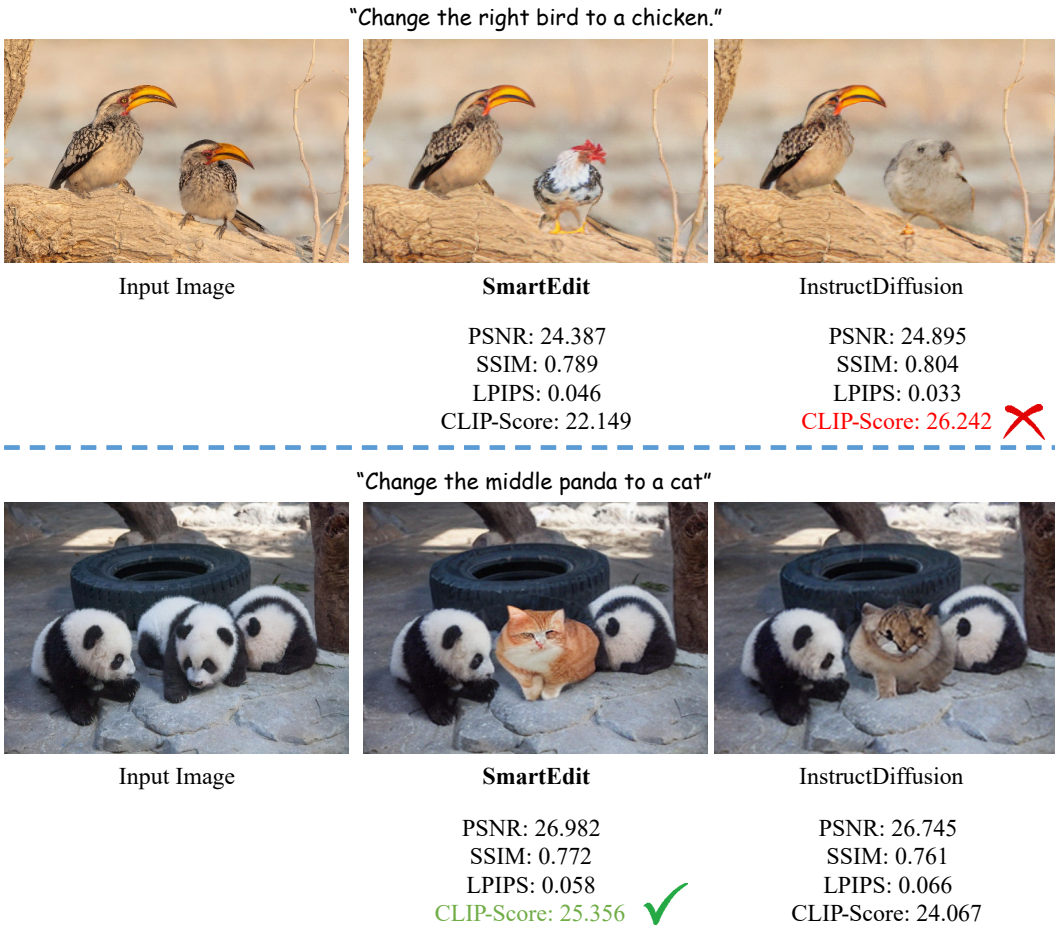

Figure 10. The evaluation of the outputs generated by SmartEdit and InstructDiffusion.

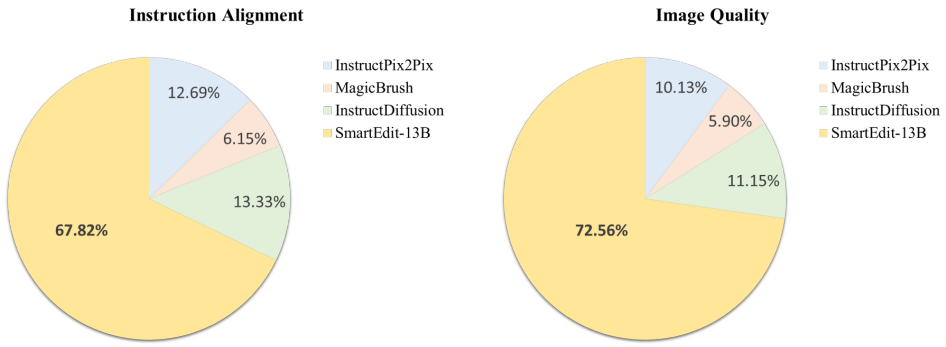

Figure 11. The results of user studies, comparing the results generated by InstructPix2Pix, MagicBrush, InstructDiffusion, and SmartEdit-13B. Based on the results from both the Instruction Alignment and Image Quality perspectives, SmartEdit demonstrates superior effectiveness.

9. More Visual Results on Reason-Edit

For complex understanding scenarios, we show more editing results of SmartEdit in Fig. 12. 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. 13. 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. 14. 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 12. 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.

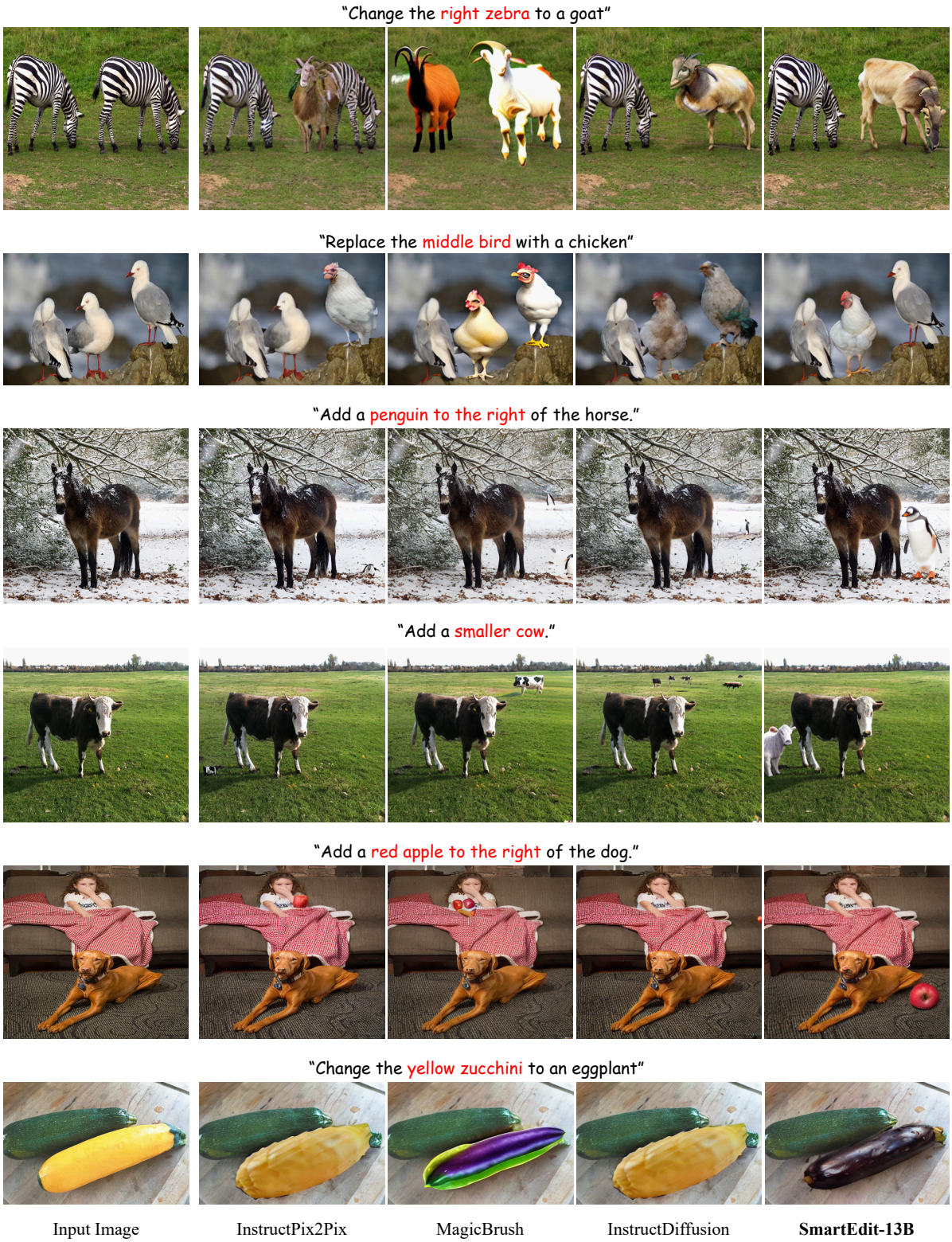

Figure 13. Qualitative comparison on Reason-Edit dataset (mainly on the complex understanding scenarios). Compared to other methods, SmartEdit can precisely edit specific objects in images according to instructions, while keeping the content in other areas unchanged.

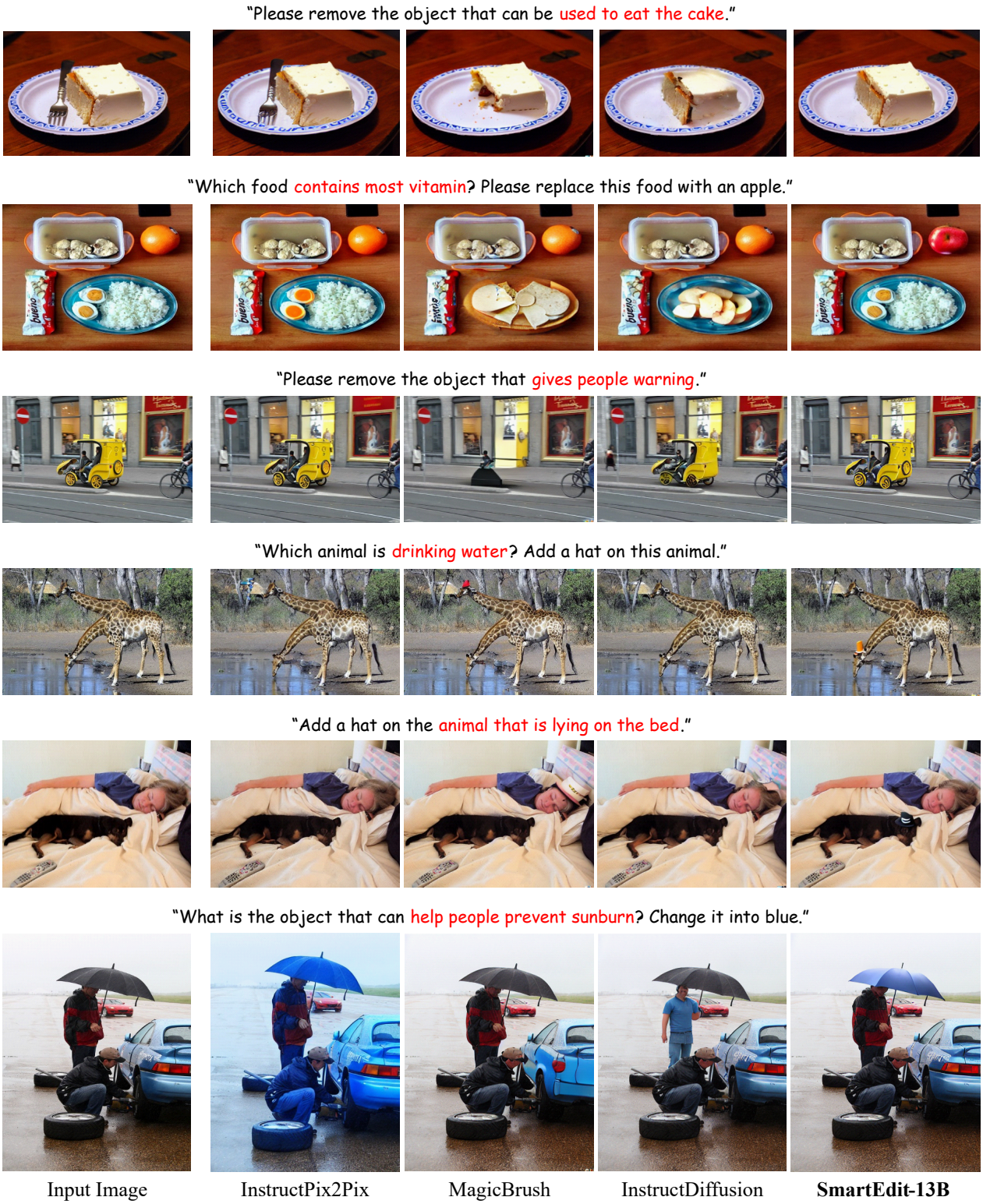

Figure 14. Qualitative comparison on Reason-Edit dataset (mainly on the complex reasoning scenarios). For reasoning scenarios, SmartEdit can effectively utilize the reasoning capabilities of the LLM to identify the corresponding objects, and then edit the objects according to the instructions. Other methods perform poorly in these scenarios.

10. Results of SmartEdit and Other Methods on MagicBrush

In Fig. 15, we demonstrate the performance of SmartEdit on the MagicBrush [40] 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 [40], and InstructDiffusion [12] on the MagicBrush test dataset. The quantitative results are presented in Tab. 4. It's important to note that MagicBrush releases two distinct checkpoints, MagicBrush-52 (trained for 52 epochs) and MagicBrush-168 (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. 4, 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. 16), 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. 4, 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. 17. 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 4. Quantitative comparison (L_1/CLIP-I/CLIP-T/DINO-I) on the MagicBrush test set.

11. Difference between SmartEdit, MGIE and InstructDiffusion

Recently, we have noticed a concurrent work: MGIE [9]. 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. 13 and Fig. 14, 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.

https://huggingface.co/vinesmsuic/magicbrush-jul7
https://huggingface.co/vinesmsuic/magicbrush-paper

Figure 15. 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.

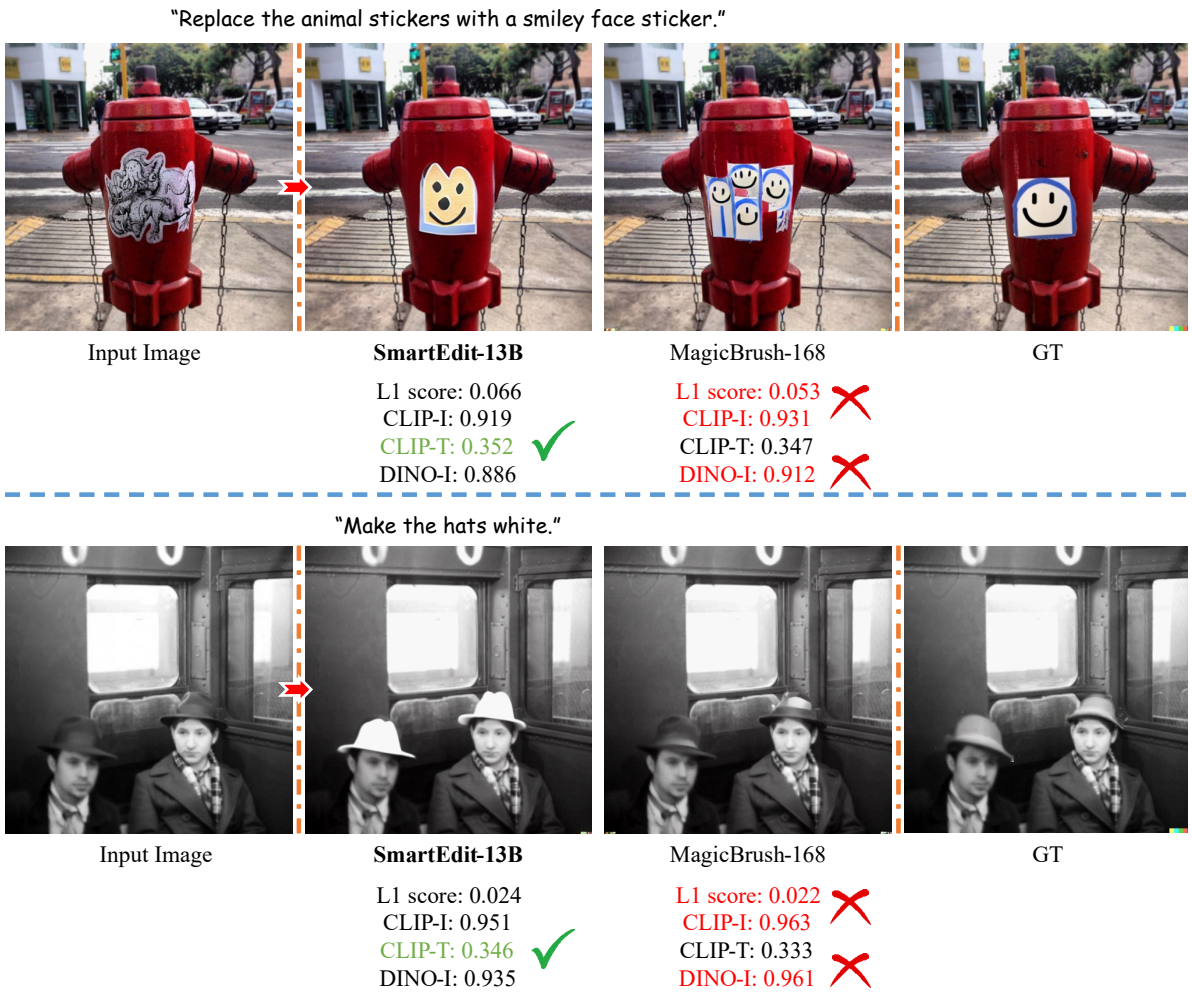

Figure 16. The evaluation of the outputs generated by SmartEdit and MagicBrush-168. Here we adopt these four metrics: L_1, CLIP-I, CLIP-T, and DINO-I metrics. The results indicate that SmartEdit performs better than MagicBrush-168. However, it's important to note that the L_1, CLIP-I, and DINO-I metrics may not correspond well with these results.

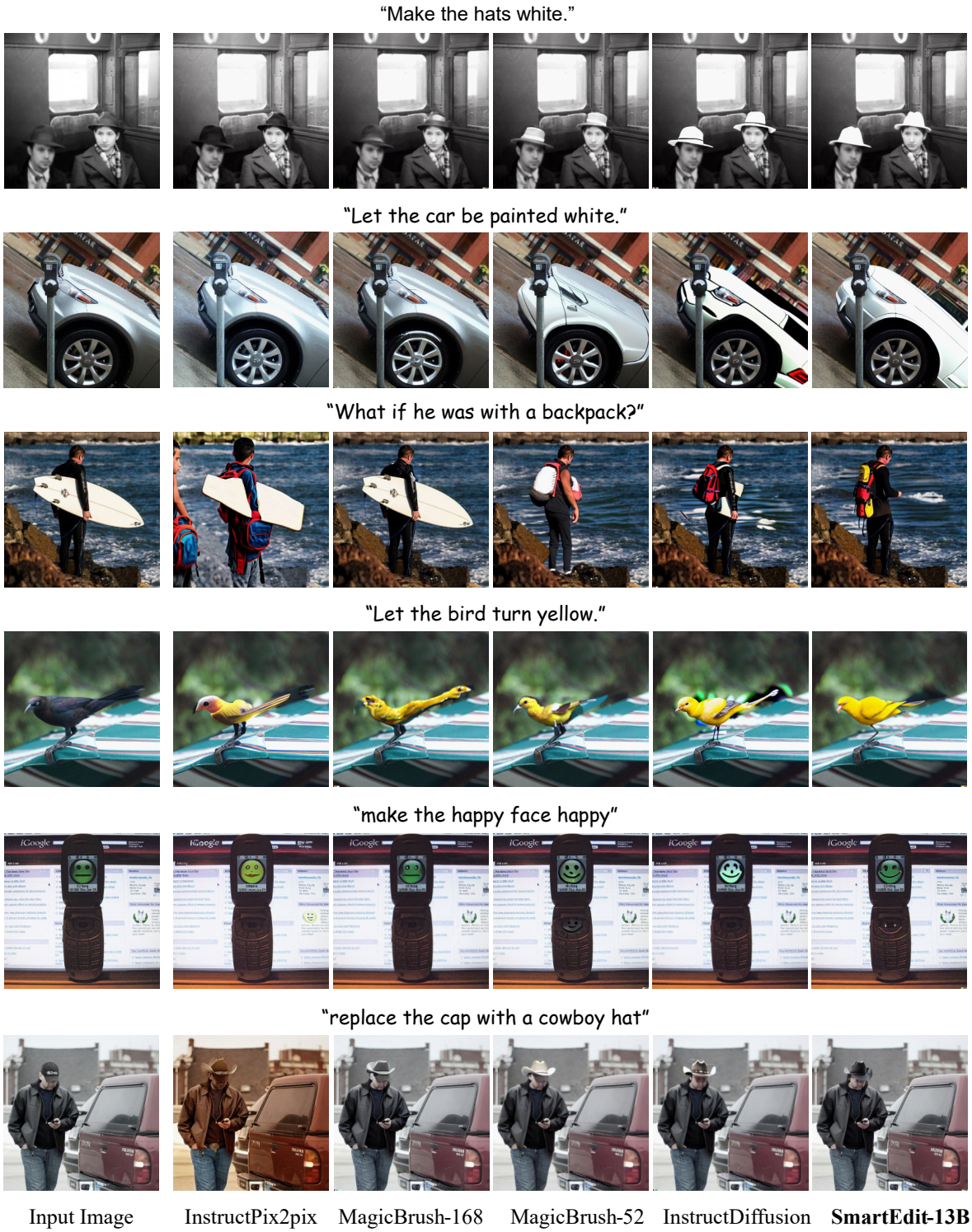

Figure 17. Qualitative comparison between our SmartEdit, MagicBrush-168, MagicBrush-52, InstructDiffusion, and InstructPix2Pix. Compared against other methods, SmartEdit effectively adheres to the instructions, showcasing superior results.