Supplementary Material for "InsightEdit: Towards Better Instruction Following for Image Editing"

1. Supplementary Materials of Dataset

1.1. Overview of Dataset

We propose an automated data construction pipeline to generate high-fidelity and fine-grained image editing pairs, accompanied by detailed instructions that exemplify advanced reasoning and comprehensive understanding. Figure S1 depicts the distribution of simple caption descriptions in the AdvancedEdit dataset, highlighting the frequency of keywords within various captions in the dataset. The upper pie chart shows the distribution of keywords within captions for removal and addition tasks, while the lower pie chart presents the distribution of the keywords in captions for the replacement task.

Figure S2 showcases some examples from the AdvancedEdit dataset. Each image editing pair consists of a source image, a target image, a simple instruction, and a complex instruction. These examples underscore the dataset's high visual fidelity and high-resolution image quality. The instructions can be broadly categorized into two types: simple instruction and advanced instruction. The simple instruction conveys the information to be modified in a straightforward manner. The advanced instruction contains better details and includes commands that require a certain level of understanding and reasoning ability.

Figure S3 compares existing image editing datasets and our proposed dataset. The comparison highlights the superior attributes of our dataset, including high resolution, exceptional visual quality, complex and detailed instructions, and robust background consistency.

1.2. Dataset Evaluation

We use VIEScore to evaluate image editing pairs, which encompasses two main aspects: **semantic consistency** and **perceptual quality**. Semantic consistency measures how well the image editing outcome adheres to the given instructions, ensuring that the edited content aligns with the specified task while preserving the integrity of the non-edited regions. Perceptual quality, on the other hand, evaluates how visually authentic the generated image appears and whether it conveys a sense of naturalness. The semantic consistency score comprises two key components: **instruction adherence**, which evaluates the degree to which the image aligns with the provided instructions, and **non-overediting**, which assesses whether the image has been altered appropriately without introducing excessive or unnatural modifications. Perceptual quality is defined by two factors: **naturalness**, which gauges the realism of the image, and **no-artifacts**, which identifies the presence of visual defects or distortions. In the dataset evaluation process, we use MLLM (*i.e.* GPT-4o) to evaluate the quality of the editing pair based on these four metrics, with an integer score ranging from 0 to 5, where 0 represents the worst and 5 represents excellent. Figure S4 shows examples of VIEScore.

In addition, for our experiment results evaluation, to better assess the model's ability to follow instructions and maintain background consistency, we compute the semantic consistency score using the following formula, let S_i represent the instruction-following score and S_n represent the non-overedit score. Each score is normalized by dividing it by 5, the maximum achievable score for each metric, thereby scaling the individual score to a range of 0 to 1. The final score is obtained by averaging these two normalized values. The computation of the VIEScore for benchmark evaluation is formally defined as follows:

$$VIEScore = \frac{1}{2}\left(\frac{1}{5} \times S_n + \frac{1}{5} \times S_i\right) \tag{1}$$

2. Supplementary Materials of Method

2.1. Training Stage 1: Embedding Space Alignment

The first training stage involves aligning the hidden states of the MLLM with the embedding space of the diffusion model. In this stage, we use CC12M, a dataset with 12 million image-text pairs, to train both the textual branch and image branch to explicitly align embedding space between MLLM and the diffusion model. For the textual branch, we apply Q-former to

align with the original CLIP text embedding of the diffusion model. In the image branch, the output of the mapper within the IAA module is trained to align with the CLIP image embedding of the source image. We extend the vocabulary of the LLM by introducing 32 special [MM] tokens to represent multi-modality comprehension of text and image information. The loss functions are as follow:

$$\begin{aligned}
\mathcal{L}_{\text{LLM}}(c) = -\sum_{i=1}^{r} \log p_{\{\omega \cup \theta\}} \Big([\text{MM}_i] \mid v, c, \\
[\text{MM}_1], \ldots, [\text{MM}_{i-1}] \Big), \\
\mathcal{L}_{\text{IB}} = \|\text{CLIP}(\mathbf{I}_{\text{src}}) - \text{Mapper}(h)\|_2^2, \\
\mathcal{L}_{\text{TB}} = \|\text{CLIP}(\mathbf{T}_{\text{c}}) - Q_\beta(q, h)\|_2^2, \\
\mathcal{L}_{\text{stage1}} = \mathcal{L}_{\text{LLM}}(c) + \mathcal{L}_{\text{IB}} + \mathcal{L}_{\text{TB}}.
\end{aligned} \tag{2}$$

The loss function consists of three components. The first loss is next token prediction loss of LLM, to learn the token embeddings of the extended vocabulary. The second loss $\mathcal{L}_{\text{IB}}$ is the Mean Squared Error (MSE) loss between the output embedding of the mapper (a Multi-Layer Perceptron layer) within the IAA module with the CLIP image encoder embedding derived from the source image. The third loss $\mathcal{L}_{\text{TB}}$ is the MSE loss between the output embedding of the Qformer and the CLIP text encoder embedding corresponding to the instruction.

2.2. Training Stage 2: Textual Branch Training

In the second training stage, the trainable components include the MLLM, the Q-former, the mapper in IAA module, the BIM, and the Unet. We finetune the MLLM in this stage using low-rank adaption(LORA). The BIM is trained to obtain the textual information better. Specifically, an explicit constraint is introduced within the IAA module to guide the visual embedding to match the features of the target image. The loss functions are as below:

$$\begin{aligned}
\mathcal{L}_{\text{MLLM}}(c) = -\sum_{i=1}^{r} \log p_{\{\omega \cup \theta\}} \Big([\text{MM}_i] \mid v, c, \\
[\text{MM}_1], \ldots, [\text{MM}_{i-1}] \Big), \\
\mathcal{L}_{\text{IAA}} = \|\text{CLIP}(\mathbf{I}_{\text{tar}}) - \text{Mapper}(h)\|_2^2, \\
\mathcal{L}_{\text{diff1}} = \mathbb{E}_{\mathcal{E}(y), \mathcal{E}(x), c_T, \epsilon \sim \mathcal{N}(0,1), t}, \\
\left[\|\epsilon - \epsilon_\delta \left(t, \text{concat}\left[z_t, \mathcal{E}(x)\right] + v_t, f_{txt}\right) \|_2^2 \right] \\
\mathcal{L}_{\text{stage2}} = \mathcal{L}_{\text{MLLM}}(c) + \mathcal{L}_{\text{IAA}} + \mathcal{L}_{\text{diff1}}.
\end{aligned} \tag{3}$$

There are three loss functions in this stage, where $\mathcal{L}_{\text{IAA}}$ is the MSE loss between the output embedding of the mapper in IAA with the CLIP image encoder embedding derived from the target image. $\mathcal{L}_{\text{diff1}}$ is the stable diffusion loss. We use InstructPix2Pix, MagicBrush, COCO, RefCOCO, GRefCOCO, COCOStuff, LISA and our AdvancedEdit dataset to train the model in this stage.

2.3. Training Stage 3: Image Branch Training

In the third training stage, we apply the de-coupled cross attention mechanism to exert both text and image feature conditions in image editing. In this stage, only the IAA module and the decoupled cross-attention parameters in the image branch of the UNet are trained. The loss function of the third training stage is as below:

$$\begin{aligned}
\mathcal{L}_{\text{diff2}} = & \mathbb{E}_{\mathcal{E}(y), \mathcal{E}(x), c_T, \epsilon \sim \mathcal{N}(0,1), t} \Big[\\
& \|\epsilon - \epsilon_\delta \left(t, \text{concat}\left[z_t, \mathcal{E}(x)\right] + v_t, f_{txt}, f_{img}\right) \|_2^2 \Big], \\
\mathcal{L}_{\text{stage3}} = & \mathcal{L}_{\text{IAA}} + \mathcal{L}_{\text{diff2}}.
\end{aligned} \tag{4}$$

In the third training stage, we apply $\mathcal{L}_{\text{diff2}}$ and $\mathcal{L}_{\text{IAA}}$. The $\mathcal{L}_{\text{diff2}}$ includes the de-coupled cross attention mechanism to integrate both text and image features. The dataset we use in this stage is the same as in the training stage 2.

3. Supplementary of Experiment

Figure S5 presents additional showcases of InsightEdit, demonstrating its robust capabilities in instruction following and maintaining background consistency. InsightEdit excels in accurately interpreting a wide range of instructions, including specific details related to location, color, quantity, and complex descriptions. Furthermore, it effectively handles instructions requiring deeper understanding and reasoning, such as recognizing object that can sit on in the image is a bench. These examples highlight InsightEdit's ability to comprehend nuanced instructions and preserve critical contextual elements while performing image editing tasks.

Figure S6 presents qualitative comparisons between InsightEdit and existing state-of-the-art instruction-based image editing methods. The results demonstrate that InsightEdit outperforms other methods in complex instruction following. In the third example, other methods fail to interpret the instruction *"object capable of flight"* while InsightEdit, trained on the AdvancedEdit dataset, successfully executes the task. Similarly, our method is the only one that correctly identifies and removes the specified objects in the last two examples. Additionally, InsightEdit excels at understanding fine-grained instructions, such as the description *"a butterfly featuring orange and yellow wings"*, which is accurately reflected in the generated image. Our approach also outperforms others in background preservation, maintaining consistency in non-edited regions. In the first two examples, all other methods fail to preserve the original content, while InsightEdit ensures a seamless integration of edited and non-edited areas.

Figure S1. **Statistics for the Advanced dataset.** The upper pie chart shows the keyword distribution in simple captions for addition and removal tasks, while the lower chart illustrates the distribution for replacement tasks, highlighting the rich diversity of edited object types in our dataset.

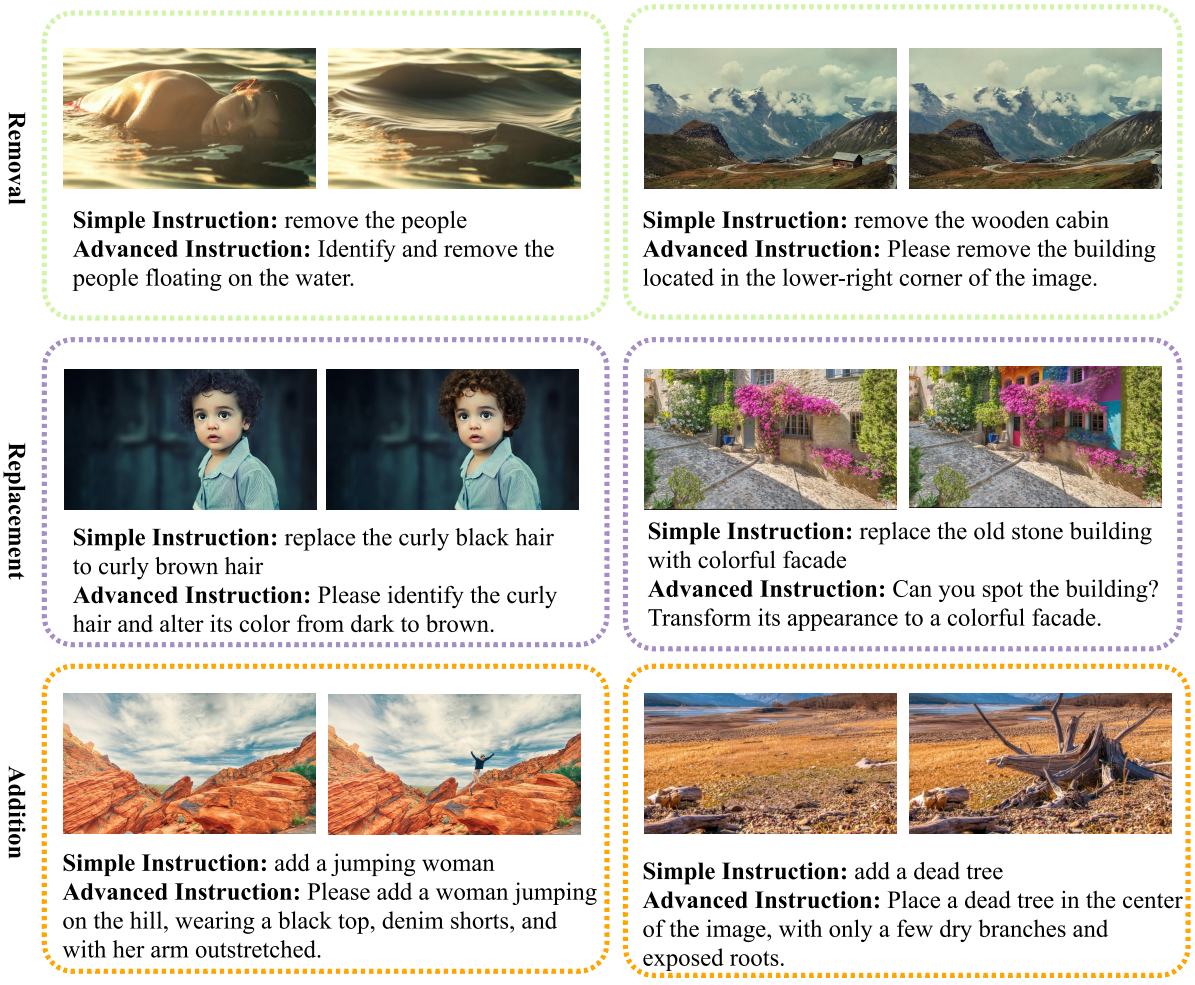


Figure S2. **Examples of the Advanced Dataset.** The top column illustrates removal tasks, the center column showcases replacement tasks, and the bottom column presents addition tasks, each with both simple and advanced instructions.

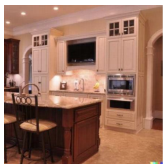
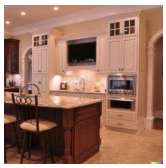
Datasets	Source Image Example 1	Target Image Example 1	Source Image Example 2	Target Image Example 2	Problems
InstructPix2Pix					• **Low Resolution** • **Bad Background Consistency** • **Simple Instructions** • **Low Visual Quality**
	Make her a farmer.		Swap the cyclist for a biker.		
MagicBrush					• **Simple Instructions** • **Manually Pipeline**
	Add a dolphin jumping out of the water.		Turn on the faucet.		
HIVE					• **Low Resolution** • **Bad Background Consistency** • **Simple Instructions** • **Low Visual Quality**
	Change retro to futuristic.		Make the man a woman.		
UltraEdit					• **Low Resolution** • **Low Visual Quality**
	Replace the cows with flamingos.		Add a rainbow in the sky above them.		
HQ-Edit					• **Bad Background Consistency** • **Simple Instructions**
	Alter her hair color to black.		Change the weather to rainy.		
EditWorld					• **Bad Background Consistency**
	What appears when the clock strikes midnight?		What happens when the door creaks open?		
AdvancedEdit					
	Simple Instruction: remove the hat **Advanced Instruction:** What is present on the top of the head in the original image? Please remove it.		**Simple Instruction:** change the black jacket to a brown jacket **Advanced Instruction:** Replace the black textured leather jacket worn by the person in the picture with a brown suede jacket, keeping the style of the jacket unchanged.		

Figure S3. **Comparison examples between previous datasets and ours.** We can see that in these examples, AdvancedEdit outperforms others in terms of good background consistency, complex instructions, and high visual quality.

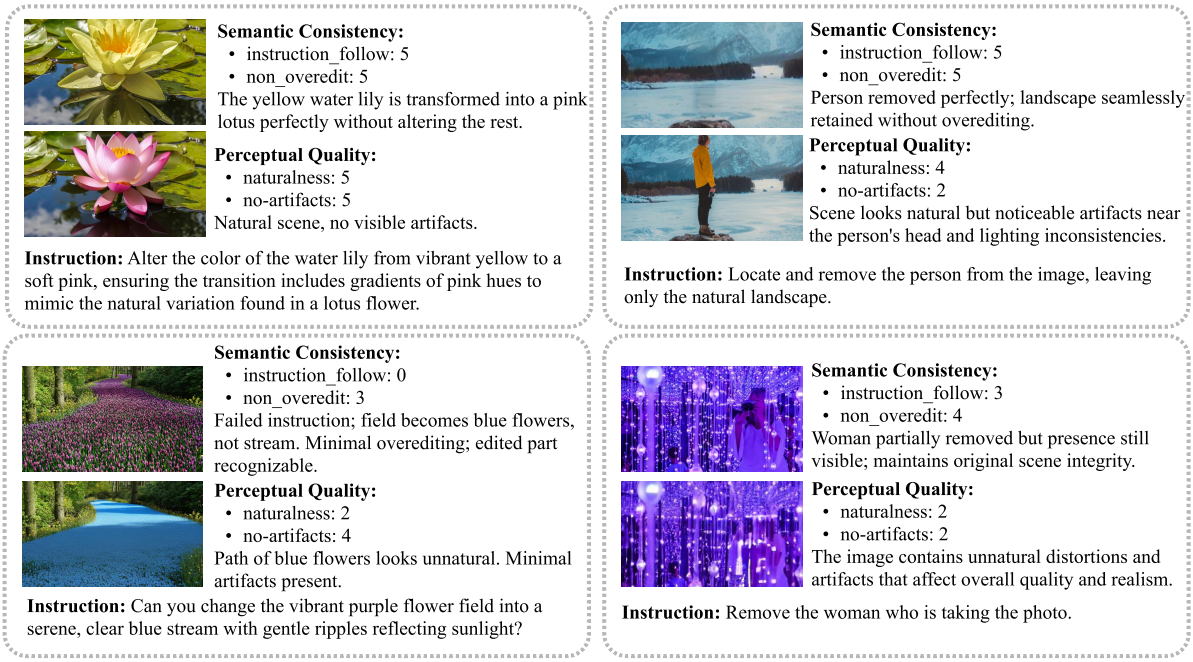


Figure S4. **Examples of VIEScore.** To assess the semantic consistency of an image editing pair, we use MLLM, which assigns integer scores ranging from 0 to 5 based on instruction-following ability and background consistency, along with a detailed explanation of the scoring criteria. Similarly, we apply MLLM to evaluate the perceptual quality of the image editing pair, using a 0-5 scale to assess naturalness and the presence of artifacts, accompanied by an explanation of the scoring process.

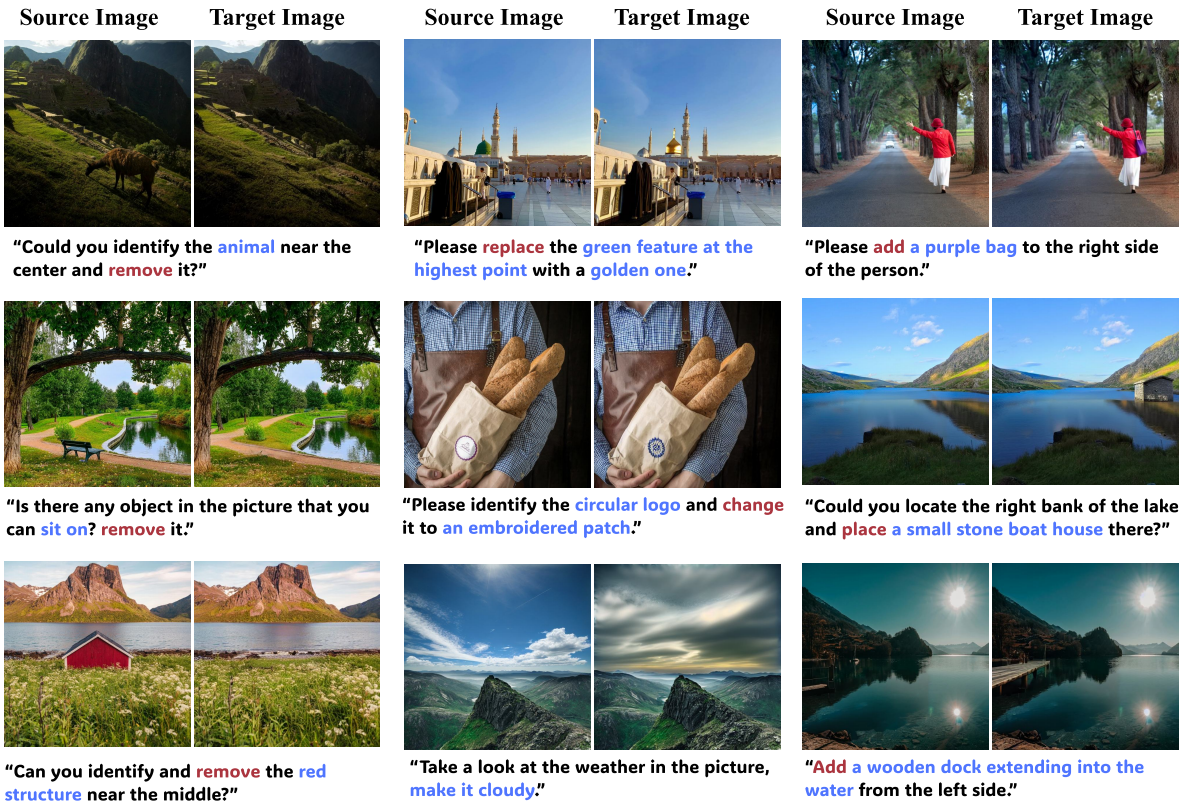


Figure S5. **Additional showcases of InsightEdit.** The left illustrates the removal task, the center illustrates the replacement task, and the right illustrates the addition task.

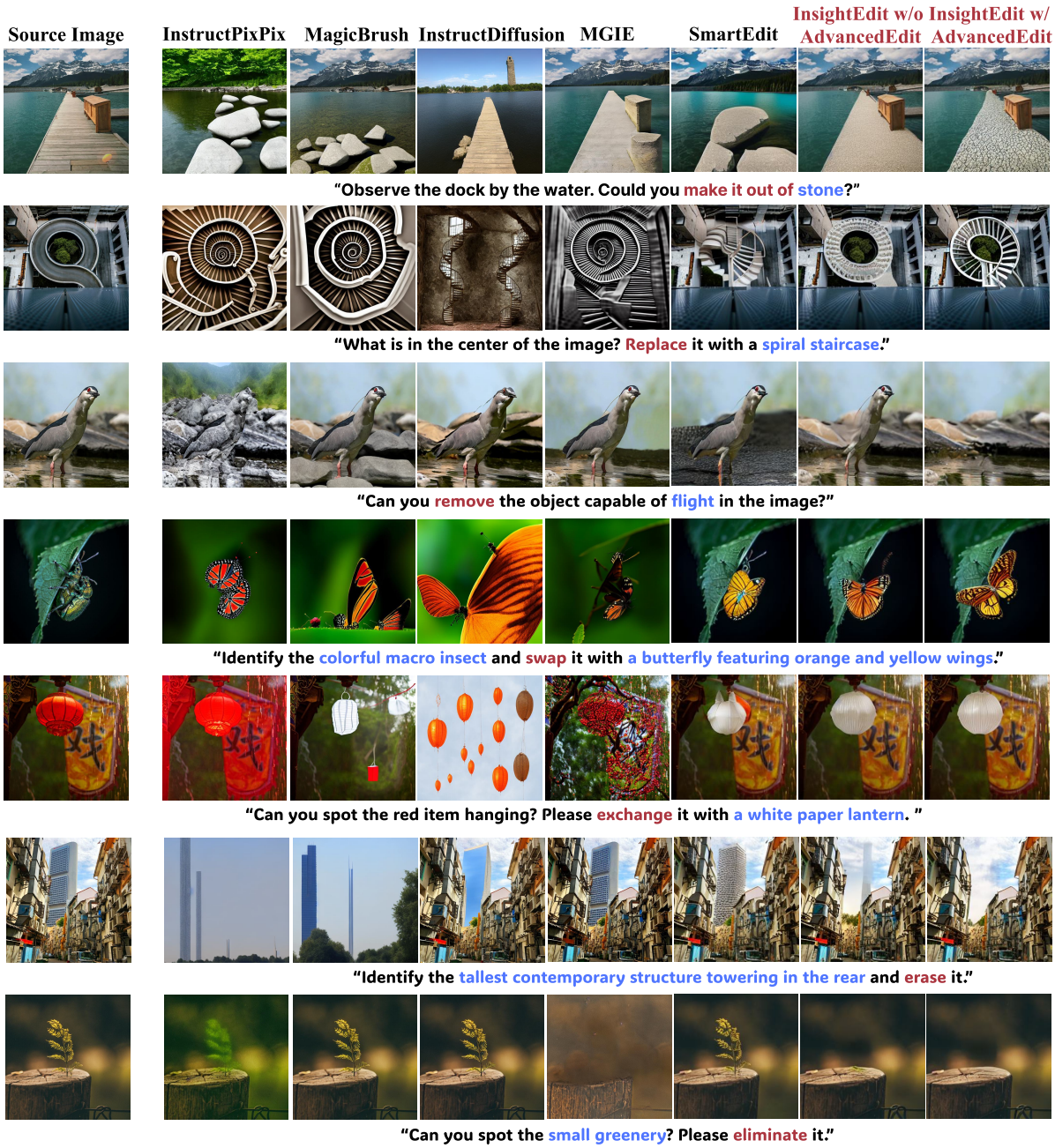


Figure S6. **Additional comparison on AdvancedEdit-Eval.** InsightEdit shows superior instruction following and background consistency capability.