InsightEdit: Towards Better Instruction Following for Image Editing

Yingjing Xu[1,2*], Jie Kong[2*†], Jiazhi Wang[2], Xiao Pan[1], Bo Lin[1‡], Qiang Liu[2]

[1]Zhejiang University, [2]01.ai

{poppyxu, xiaopan, rainbowlin}@zju.edu.cn, {kongjie, wangjiazhi, liuqiang}@01.ai

Abstract

In this paper, we focus on the task of instruction-based image editing. Previous works like InstructPix2Pix, Instruct-Diffusion, and SmartEdit have explored end-to-end editing. However, two limitations still remain: First, existing datasets suffer from low resolution, poor background consistency, and overly simplistic instructions. Second, current approaches mainly condition on the text while the rich image information is underexplored, therefore inferior in complex instruction following and maintaining background consistency. Targeting these issues, we first curated the AdvancedEdit dataset using a novel data construction pipeline, formulating a large-scale dataset with high visual quality, complex instructions, and good background consistency. Then, to further inject the rich image information, we introduce a two-stream bridging mechanism utilizing both the textual and visual features reasoned by the powerful Multimodal Large Language Models (MLLM) to guide the image editing process more precisely. Extensive results demonstrate that our approach, InsightEdit, achieves state-of-the-art performance, excelling in complex instruction following and maintaining high background consistency with the original image. The project page is https://poppyxu.github.io/InsightEdit_web/.

1. Introduction

Image editing involves altering an image's appearance, structure, or content, encompassing a range of changes from subtle adjustments to major transformations [13]. It has witnessed significant advancements in image editing tasks alongside the development of diffusion models [2, 4, 9, 16, 31], leading to impressive results.

Existing works like InstructPix2Pix [2], MagicBrush [39], and InstructDiffusion [9] have investigated end-to-end image editing. Some recent efforts [8, 14]

explore using MLLM to comprehend instructions, thereby enhancing the ability to cope with complex editing tasks. However, two challenges still remain: (i) **Lack of high-quality datasets.** As illustrated in Table 1, current datasets suffer from low resolution, poor visual quality and background consistency, and overly simplistic, template-based instructions. Specifically, most datasets rely on Prompt2Prompt [10] method, which struggles to achieve precise control over the generated images and fails to maintain consistency in unedited regions. These limitations hinder the image editing model's capability in both complex instruction following and high-fidelity target image generation. (ii) **Lack of rich image condition.** Current methods primarily use the CLIP text encoder to provide conditions, yet it often exhibits limited ability to understand the instructions. Some approaches further leverage the more advanced MLLM to better comprehend instructions. However, they still mainly focus on understanding instructions at the textual level, while neglecting to capture the rich visual semantics of the image, therefore showing weak performance in complex instruction following and maintaining background consistency.

To address the dataset issue, we propose an automated data construction pipeline to generate editing pairs with complex instructions and strong background consistency. Leveraging the perceptive capabilities of MLLM, our approach extracts detailed object information from high-resolution images and utilizes an advanced mask-based editing model to produce realistic and controllable edits. Using this method, we introduce AdvancedEdit Dataset, a comprehensive collection comprising over 2,500,000 editing pairs with high visual quality, complex instructions, and good background consistency.

To address the lack of rich image condition, we employ a two-stream bridging mechanism to integrate both high-level textual and rich visual information into the denoising process of the diffusion model. This interaction effectively extracts the conditional information, enabling precise and controllable image editing.

Our method achieves state-of-the-art performance on

| Source Image | Target Image | Source Image | Target Image | Source Image | Target Image |

"Please locate and remove the person wearing a white shirt from the image."

"Locate the green items suspended from above and substitute them with pendant lighting."

"Please add a black park bench behind the large tree."

"What's hanging up in the image? Please remove it."

"Find the white container and change it to a wooden one."

"Please locate the tallest building and add a clock tower in front of it."
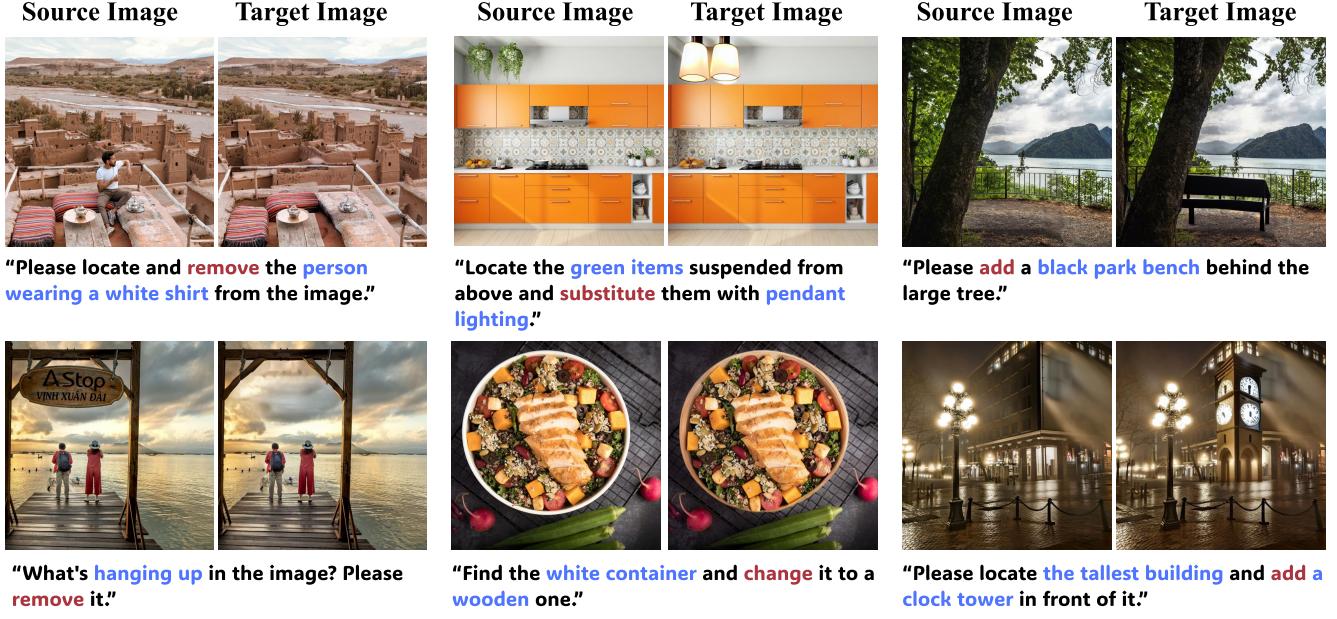
Figure 1. We propose **InsightEdit**, an end-to-end instruction-based image editing model, trained on high-quality data and designed to fully harness the capabilities of Multimodal Large Language Models (MLLM), achieving high-quality edits with strong instruction-following and background consistency.

Dataset	Editing Pairs	Resolution	Good Background Consistency	Complex Instructions	High Visual Quality	Automated Pipeline
InstructPix2Pix [2]	313,010	512^2	✗	✗	✗	✓
MagicBrush [39]	10,388	1024^2	✓	✗	✓	✗
HIVE [42]	1,100,000+	512^2	✗	✗	✗	✓
FaithfulEdit [6]	52,208	1024^2	✗	✗	✓	✓
UltraEdit [43]	4,108,262	512^2	✓	✓	✗	✓
HQEdit [15]	187,350	900^2	✗	✗	✓	✓
EditWorld [35]	10,000+	512^2	✗	✓	✓	✓
AdvancedEdit (Ours)	**2,536,674**	**1024^2**	✓	✓	✓	✓

Table 1. **Comparison between previous datasets and ours.** We provide more detailed comparisons in the appendix.

both the Reason-Edit [14] and AdvancedEdit-Eval, demonstrating its strong capability in complex instruction following and maintaining background consistency.

In summary, our contributions are as follows:

1. We propose an automated data construction pipeline for image editing, specifically engineered to facilitate the training of models for complex instruction-based image editing tasks.
2. We produce the AdvancedEdit Dataset, a large-scale, high-quality, and fine-grained image editing dataset with complex instructions and good background consistency. Additionally, we introduce the AdvancedEdit-Eval dataset to assess the model's ability to handle complex instructions.
3. We present InsightEdit, which utilizes a two-stream bridging mechanism to guide the image editing pro-

cess with both textual and image features reasoned by MLLM. Extensive experiments are conducted to prove the model's ability to follow complex instructions and maintain high background consistency.

2. Related Work

2.1. Image Editing Methods

Recent advancements in image editing can be broadly categorized into mask-based and mask-free approaches. Mask-based image editing [1, 31–33] requires the original image, instructions, and mask image indicating the regions to be edited as input. The mask image typically needs to be manually specified by the user. Blended Latent Diffusion [1] extends Latent Diffusion Model [28] by integrating them into a localized image editing framework, enabling the tar-

geted redrawing of specific regions during the denoising process. BrushNet [16] is inspired by the architecture of ControlNet [40]. It employs a dual UNet [29] structure, where an additional branch is used to enhance feature extraction from the masked regions. Power-Paint [44] focuses on the distinct characteristics of various image editing tasks by learning a token to classify different editing tasks. Mask-based approaches enable localized redrawing and fine-grained control, resulting in a stable and high-visual-quality output. However, they require additional mask information and are sensitive to mask shape and the specific editing task.

Mask-free methods [2, 9, 14, 19] present a promising alternative by reducing the dependency on explicit masks and allowing for more flexible and intuitive image editing. InstructPix2Pix [2] utilizes GPT-3 [25] and a Prompt2Prompt [10] methodology to construct an image editing dataset. InstructDiffusion [9] builds upon the network design of InstructPix2Pix, aiming to unify various vision tasks through joint training. MGIE [19] enhances instruction-based image editing by learning to generate expressive instructions with MLLM. SmartEdit [14] utilizes MLLM to generate text embeddings, leveraging the MLLM's advanced reasoning and comprehension capabilities.

Although instruction-based image editing methods do not require explicit mask guidance, making them more intuitive and convenient, they often yield lower image quality than mask-based approaches. They also face challenges in following complex instructions and maintaining background consistency.

2.2. Image Editing Datasets

Several works have recently introduced image editing datasets. InstructPix2Pix [2] introduces a large-scale image editing dataset created through fine-tuned GPT-3 and a Prompt-to-Prompt methodology integrated with Stable Diffusion. MagicBrush [39] further provides a limited-scale, manually annotated dataset for instruction-guided real image editing. Subsequent work, HIVE [42] introduces more training triplets and human ranking results to provide stronger supervisory signals for improved model training. FaithfulEdits [6] employed inpainting techniques followed by a filtering process using Visual Question Answering (VQA) models. UltraEdit [43] explores region-based data generation, employing mask generation to facilitate object redrawing in specified areas. Some works explore the diversity of instructions, such as EditWorld [35], which defines and categorizes instructions grounded in various world scenarios.

As shown in Table 1, despite the notable contributions of these datasets, current image editing datasets still face several challenges, including the inability to scale the data con-

struction process, low dataset resolution, poor background consistency, and a lack of fine-grained instructions for complex understanding and reasoning tasks.

3. Dataset Construction

3.1. Automated Pipeline

We propose an automated data construction pipeline focused on generating high-fidelity, fine-grained image-editing pairs with detailed instructions that demonstrate advanced reasoning and understanding. We categorize the image editing tasks into three types: removal, addition, and replacement. Figure 2 presents our data preparation workflow.

Step 1: Caption & Object Extraction. We leverage the advanced comprehension capabilities of MLLM to generate a global caption that effectively conveys the image's content. Using this caption, we utilize LLM to extract a JSON list of objects, identifying those with physical significance. Each object is defined by a simple caption (e.g., *blue T-shirt*) and a detailed description (e.g., *A shirt worn by the man, typically made of cotton, featuring a blue color*).

Step 2: Mask Generation. For mask generation, we apply GroundedSAM [27] to extract local masks for each object, filtering out low-confidence masks based on a predefined threshold, thus obtaining accurate object and mask pairs for further processing.

Step 3: Editing Pair Construction. As mentioned in Section 2.2, mask-based image editing models exhibit superior image generation capabilities, offering better control over the generation process for specific tasks. In the process of constructing the edited images, we employed the state-of-the-art mask-based methods [16, 44] to generate the target image, enabling the creation of more fine-grained and controllable image editing pairs. We categorized the image editing generation tasks into three types: removal, addition, and replacement.

For removal, we apply a mask-based approach, using both the original image and the object mask to generate the post-removal image, simultaneously generating and storing the template instruction: *"remove the [object]."* For addition, we swap the source and target images. The template instruction is: *"add the [object]."*

For the replacement task, we leverage MLLM's creative capabilities to propose an alternative object for the scene, often resulting in plausible and innovative outcomes. Simultaneously, we generate and store the template instruction: *"replace the [source object] with the [target object]."*.

Step 4: Instruction Recaptioning. To diversify the complexity of the instructions, we recaption the instructions into both simple and advanced versions. The simple instructions are generated through synonym replacement and task template modifications. For instance, the instruction *"replace*
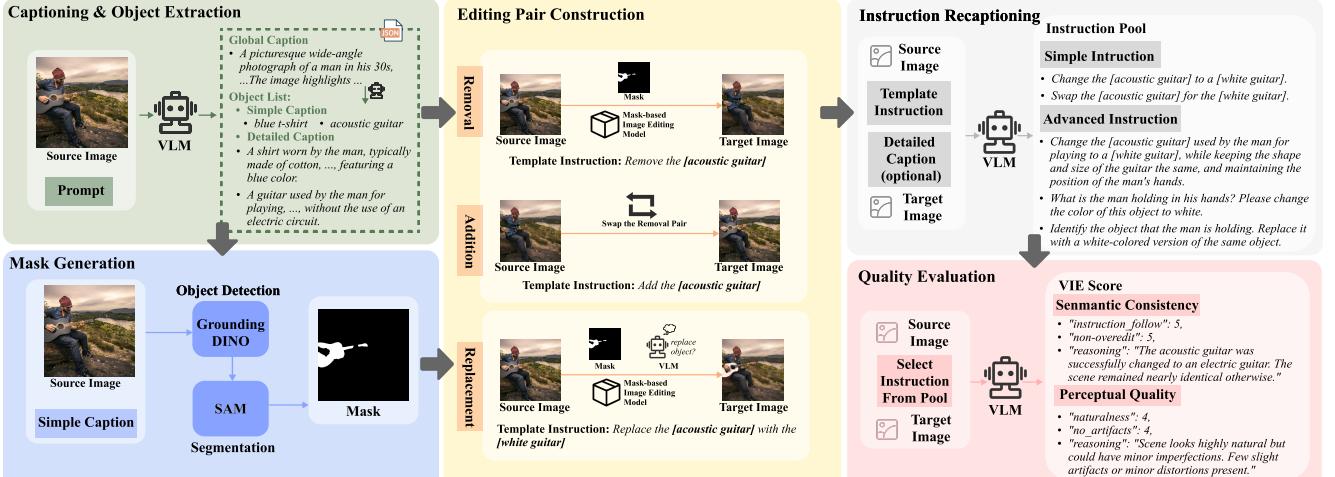
Figure 2. The overall data construction pipeline. (1) Captioning & Object Extraction: Utilizing VLM to generate a global caption from the source image, and further get an object JSON list contains both simple caption and detailed caption. **(2) Mask Generation:** Utilizing GroundedSAM to obtain the corresponding mask of each object. **(3) Editing Pair Construction:** Utilizing mask-based image editing model to construct target image and templated instruction. **(4) Instruction Recaptioning:** Utilizing VLM to rewrite instruction to gain diverse instructions. **(5) Quality Evaluation:** Filtering the datasets using VIEScore.

the [acoustic guitar] with [white guitar]" is rewritten as *"Swap the [acoustic guitar] for the [white guitar]."*

In contrast, the advanced instructions are designed to increase complexity in two key ways. First, we replace simple object descriptions with detailed ones prepared in Step 1 and apply different task templates. Second, we recaption instructions to gain reasoning capabilities, such as *"What is the man holding in his hands? Please change the color of this object to white."* This approach challenges the editing model to generate images with more intricate details and greater precision.

Step 5: Quality Evaluation. To construct a high-quality dataset, we assess and filter the image-editing pairs using VIEScore [20], which aligns evaluations with human preferences via MLLM. The evaluation framework consists of two components: semantic consistency, which measures instruction following and evaluates whether the image has been over-edited, and perceptual quality, which evaluates image fidelity, including the presence of artifacts or blurriness. The rating example of VIEScore can be seen in Figure 2.

3.2. AdvancedEdit Dataset

We choose Pexels [1], a high-quality real-world photographic image dataset, as our source data. Pexels is a popular platform offering a large collection of high-resolution, royalty-free images contributed by photographers around the world. The images in the Pexels dataset generally exhibit an average resolution of approximately 2K or higher. This dataset contains over 1 million photographic images and encom-

---
[1]https://www.pexels.com/

passes a broad range of subjects, covering various world scenes. Given that each image contains multiple objects and is associated with various instruction tasks, our dataset comprises a total of $2,536,674$ editing pairs. A more detailed analysis of the proposed dataset is provided in the appendix.

We define the image editing pairs with simple instructions as **SimpleEdit**, and image editing pairs with complex instructions as **AdvancedEdit**. We will discuss the influence of instructions with varying complexity in Section 5.3. Additionally, we introduce the **AdvancedEdit-Eval**, comprising 300 curated image pairs, covering a range of tasks including removal, addition, and replacement. The evaluation dataset includes a variety of intricate and nuanced image editing scenarios, necessitating that the model possesses a certain level of understanding and reasoning capabilities.

4. Method

Overview. The overall architecture of InsightEdit is depicted in Figure 3. It mainly consists of a comprehension module, a bridging module, and a generation module. Specifically, the comprehension module leverages MLLM to comprehend the image editing task; the bridging module integrates both text and image features into the denoising process of the diffusion model; and the generation module receives editing guidance via the diffusion model to generate the target image.

4.1. Comprehension Module

Multimodal large language model receives the original image and the editing instruction to comprehend the image editing task. In the comprehension module, we use LLaVA-
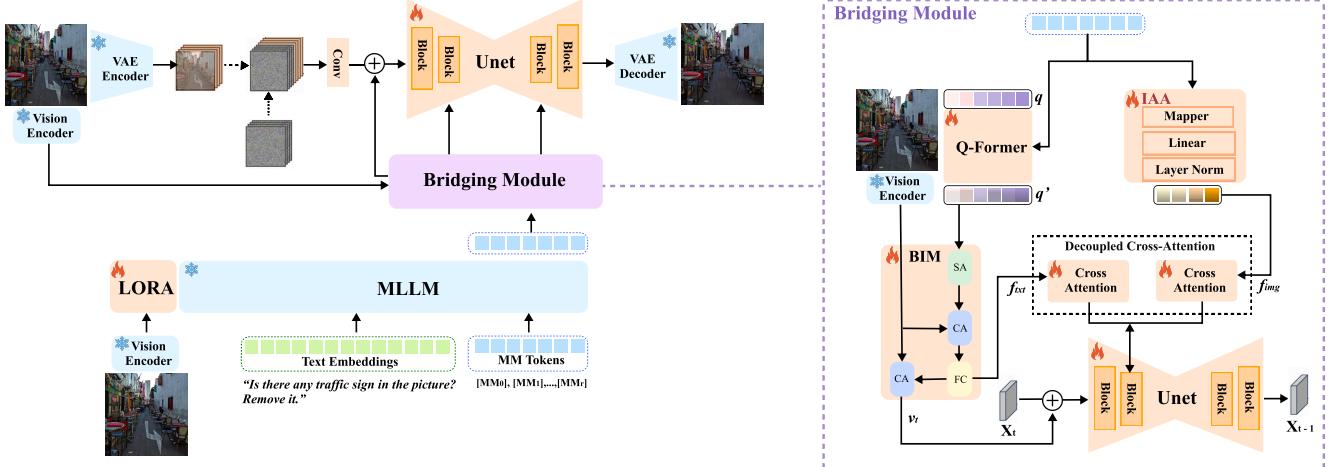
Figure 3. **The overall architecture of InsightEdit.** It mainly consists of three parts: (1) **Comprehension Module:** A comprehension module that leverages MLLM to perceive and comprehend the image editing task; (2) **Bridging Module:** A bridging module that better interacts and extracts both the textual and image features; (3) **Generation Module:** A generation module that receives editing guidance via diffusion model to generate the target image.

7B [24] as our vision-language foundation model. The feature of source image $\mathbf{I}_{src}$ is extracted by a vision encoder $E_\phi(\cdot)$ and the text instruction is tokenized as text embeddings c. Aligned by a fully connect layer, the original image features are sent into the LLaVA decoder [30] along with text embeddings. The mentioned process is represented as:

$$
\begin{aligned}
v &= \mathrm{FC}(E_\phi(\mathbf{I}_{src})), \\
h &= \mathrm{LLaVA}_\omega(v, c).
\end{aligned}
\tag{1}
$$

Inspired by GILL [19], we extend the vocabulary of the LLM by introducing r special [MM] tokens, where "MM" indicates multi-modality comprehension of text and image information via MLLM. These tokens are appended to the end of the instruction c, and the model is trained by minimizing the negative log-likelihood of predicting [MM] tokens, conditioned on the previously generated tokens. The loss function can be represented below:

$$
L_{\mathrm{LLM}}(c) = -\sum_{i=1}^{r} \log p_{\{\omega \cup \theta\}}([\mathbf{MM}_i] \mid v, c, \\
[\mathbf{MM}_1], \dots, [\mathbf{MM}_{i-1}]).
\tag{2}
$$

The majority of the LLM parameters are kept frozen, while the LoRA [12] is employed to facilitate efficient training.

4.2. Bridging Module

The textual features provide high-level information while the image features provide more detailed condition. Therefore, in the bridging module, to comprehensively integrate both textual and image features as the conditions, we design a dual-stream condition alignment methodology.

Textual Branch via Q-Former & BIM. For the textual branch, similar to SmartEdit [14], we start from the *text-aligned Q-Former* [5, 22] which has primarily aligned with the original clip text embedding space. Then, given the learnable tokens q as inputs, we propose to further extract the reasoned textual information from the [MM] token hidden states h via cross-attention:

$$
q' = Q_\beta(q, h),
\tag{3}
$$

where q' represents the Q-former outputs that mainly encode the text instruction information.

Additionally, following SmartEdit [14], we employ the BIM module that enables bidirectional information exchange between the input image and the MLLM output:

$$
f_{txt}, v_{txt} = \mathrm{BIM}(E_\phi(\mathbf{I}_{src}), q'),
\tag{4}
$$

where f_{txt} and v_{txt} are the interacted features obtained from the BIM module. f_{txt} is used as the final textual condition to UNet and v_{txt} is the textual-aware vision features which is later added with U-Net inputs.

Image Branch via IAA. For the image branch, we propose an *Image Alignment Adapter (IAA)* module to activate the image information reasoned by MLLM, which consists of a mapper, a linear layer, and a layer normalization.

In detail, the mapper is a Multi-Layer Perceptron (MLP) that transforms the [MM] token hidden states $h \in \mathbb{R}^{r \times 4096}$ into a single embedding $\mathrm{Mapper}(h) \in \mathbb{R}^{1 \times 768}$, which is then aligned with the target image features extracted by a CLIP vision encoder. Intuitively, the target image contains direct information about the editing goal, *e.g.*, the expected background, therefore can provide explicit supervision. After that, the linear is further employed to expand

Table 2. **Quantitative comparison on AdvancedEdit-Eval.**

Methods	VIEScore↑	CLIPScore↑	PSNR↑	SSIM↑	LPIPS↓
InstructPix2Pix	0.342	19.528	20.192	0.694	0.182
MagicBrush	0.352	19.751	22.636	**0.743**	0.132
InstructDiffusion	0.318	19.390	18.025	0.624	0.223
MGIE	0.361	19.047	20.074	0.691	0.199
SmartEdit-7B	0.682	20.114	20.115	0.651	0.131
InsightEdit	0.738	20.395	21.267	0.675	0.112
InsightEdit with AdvancedEdit	**0.831**	**21.002**	**22.871**	0.716	**0.071**

the $\text{Mapper}(h) \in \mathbb{R}^{1 \times 768}$ back to token sequences $\mathbb{R}^{N \times 768}$ (N is set as 4 by default) followed by a layer normalization. Formally, the mentioned process is represented as:

$$\mathcal{L}_{\text{IAA}} = \|\text{CLIP}(\mathbf{I}_{\text{tar}}) - \text{Mapper}(h)\|_2^2, \\ f_{img} = \text{IAA}(h), \tag{5}$$

where $\mathcal{L}_{\text{IAA}}$ is the supervision from the target image, and f_{img} represents the output of the IAA, which will be sent to further cross-attention operation in U-Net.

Compared to textual features, the image features contain more detailed and comprehensive information, which can guide image editing more explicitly.

4.3. Generation Module

In the process of target image generation, inspired by IP-adapter [37], we apply a decoupled cross-attention mechanism to integrate both textual and image features. An additional cross-attention layer is employed on top of the cross-attention layer in the original UNet block to interact with the image features. As the text cross-attention and image cross-attention are detached, we can also adjust the weight of the image condition in the inference stage:

$$\mathbf{Z} = \text{Attention}(\mathbf{Q}, \mathbf{K}_{\mathbf{txt}}, \mathbf{V}_{\mathbf{txt}}) \\ + \lambda \cdot \text{Attention}(\mathbf{Q}, \mathbf{K}_{\mathbf{img}}, \mathbf{V}_{\mathbf{img}}), \tag{6}$$

where $\mathbf{K}_{\mathbf{img}}$ and $\mathbf{V}_{\mathbf{img}}$ are projected by f_{img}, $\mathbf{K}_{\mathbf{txt}}$ and $\mathbf{V}_{\mathbf{txt}}$ are projected by f_{txt}.

For the diffusion model, we concatenate the encoded image latent $\mathcal{E}(\mathbf{I}_{src})$ with the noisy latent z_t. The process can be formulated as:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{\mathcal{E}(y), \mathcal{E}(x), c_T, \epsilon \sim \mathcal{N}(0,1), t} \\ \left[\|\epsilon - \epsilon_\delta \left(t, \text{concat}\left[z_t, \mathcal{E}(x)\right] + v_t, f_{txt}, f_{img} \right) \|_2^2 \right]. \tag{7}$$

5. Results

5.1. Experiment Settings

Implementation Details. To train InsightEdit, we use CC12M [7], InstructPix2Pix [2], MagicBrush [39],

COCO [23], RefCOCO, GRefCOCO [17, 38], CO-COStuff [3], LISA [21, 36], ReasonEdit [14] and our proposed AdvancedEdit. Due to the limited resources, we only used $202,822$ editing pairs from AdvancedEdit for the experiments.

InsightEdit is trained in three stages, as detailed in the Appendix, and the training is conducted on a single machine with 8 H100 GPUs. We adopt the AdamW optimizer [18] as the optimizer during three stages. In the first training stage, the learning rate is $2e^{-4}$ and weight decay is 0. In the second stage and the third stage, the values of learning rate, weight decay, and warm-up ratio were set to $1e^{-5}$, 0, and 0.001, respectively. In the data construction pipeline, the MLLM we utilize is GPT-4o [2], while the LLM applied is Qwen2 [34].

Metrics. For evaluating the background consistency, we use PSNR, SSIM, and LPIPS [11, 41]. For the edited area, we calculate the CLIPScore [26] by comparing the edited object with its ground truth (GT) label. To more accurately evaluate editing effects and align with human preferences, we employ VIEScore [20], which utilizes MLLM to evaluate editing performance from two perspectives: instruction-following and background consistency. **We emphasize that VIEScore might better align with human preferences, making it a more convincing measure [20].**

5.2. Comparison with State-of-the-art

Results on AdvancedEdit-Eval. We compare our method with existing state-of-the-art instruction-based image methods, including instructPix2Pix, MagicBrush, InstructDiffusion, MGIE, and SmartEdit. First, we conduct comparative experiments on the AdvancedEdit-Eval as introduced in Section 3.2. The quantitative results presented in Table 2 show that InsightEdit shows good performance across all five metrics. The experimental results in PSNR and LPIPS scores prove that InsightEdit excels in preserving non-edited regions, while the enhanced CLIPScore validates our method's superior ability to follow instructions with greater accuracy. Specifically, in terms of VIEScore, both InsightEdit itself and InsightEdit trained on the AdvancedEdit dataset achieve state-of-the-art results, demonstrating that the model's generated outputs are preferred by

---
[2]https://openai.com/

Table 3. **Quantitative comparison on Reason-Edit.** All the methods we compared have been fine-tuned using the same training data as that used by SmartEdit.

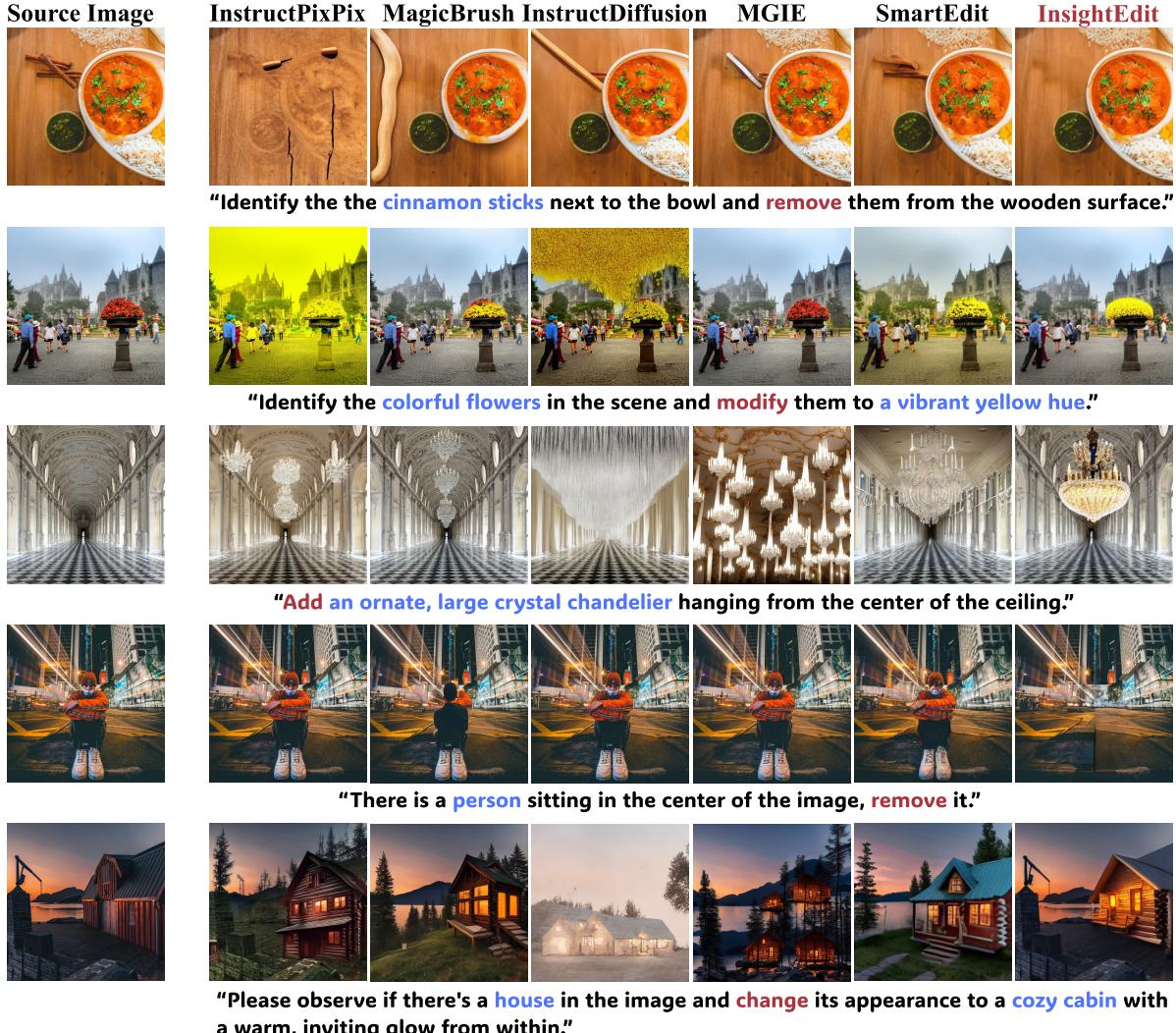| Methods	Understanding Scenarios					Reasoning Scenarios				
	VIEScore↑	CLIPScore↑	PSNR↑	SSIM↑	LPIPS↓	VIEScore↑	CLIPScore↑	PSNR↑	SSIM↑	LPIPS↓
InstructPix2Pix	0.493	22.762	21.576	0.721	0.089	0.548	19.413	24.234	0.707	0.083
MagicBrush	0.422	22.620	18.120	0.680	0.143	0.530	19.755	22.101	0.694	0.113
InstructDiffusion	0.457	23.080	23.258	0.743	0.067	0.670	19.523	21.453	0.666	0.117
MGIE	0.373	21.947	16.094	0.695	0.200	0.540	18.037	22.977	0.743	0.144
SmartEdit-7B	0.866	23.611	22.049	0.731	0.087	0.835	20.950	25.258	0.742	0.055
InsightEdit	0.901	23.734	23.588	0.749	0.067	0.893	20.867	25.712	0.747	0.049
InsightEdit with AdvancedEdit	**0.934**	**24.421**	**24.503**	**0.760**	**0.054**	**0.947**	**21.141**	**26.090**	**0.750**	**0.047**

Figure 4. **Qualitative comparison on AdvancedEdit.** InsightEdit shows superior instruction following and background consistency capability.

humans over other methods, whether regarding background preservation or the ability to follow complex instructions.

Figure 4 illustrates the qualititive results. As shown in the last column, our method trained on the AdvancedEdit dataset demonstrates a clear advantage over competing approaches. As can be seen, other methods struggle to maintain background consistency with the original image, often introducing artifacts or unintended alterations, while InsightEdit provides specific editing on the required regions. In the meanwhile, InsightEdit performs strong instruction following capabilities, having a great understanding of spatial, color, number, and detailed objects.

Results on Reason-Edit. Additionally, we conducted comparative experiments on the Reason-Edit[14], which is specifically designed to assess editing abilities with an emphasis on understanding and reasoning across various sce-

IAA	PSNR↑	SSIM↑	LPIPS↓	CLIPScore↑	VIEScore↑
	22.348	0.692	0.095	20.652	7.307
✓	**22.871**	**0.716**	**0.071**	**21.002**	**7.545**

Table 4. **Ablation of IAA.** Our IAA module performs best.

"Could you swap the white flowers with a diverse collection of colorful wildflowers, creating a vivid and colorful foreground?"

"Transform the worn wooden table into a sleek, modern marble countertop while maintaining the arrangement of the objects."

Figure 5. Demonstration of the effectiveness of the IAA module.

SimpleEdit	AdvancedEdit	PSNR↑	SSIM↑	LPIPS↓	CLIPScore↑	VIEScore↑
		21.267	0.675	0.112	20.395	6.974
✓		22.260	0.708	0.080	20.557	7.213
	✓	**22.871**	**0.716**	**0.071**	**21.002**	**7.545**

Table 5. **Ablation of AdvancedEdit.** Our method with AdvancedEdit performs best.

"Find the small bird in the image and replace it a monarch butterfly with orange and black wings."

"Can you identify the specific bright light source from the small window on the staircase wall, remove it smoothly and blend the area to match the surrounding tones? "
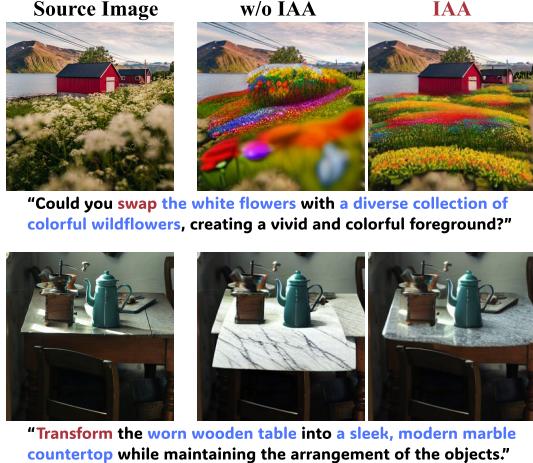
Figure 6. Demonstration of the effectiveness of AdvancedEdit.

narios (*i.e.* color, position, mirror, multiple-objects, reason, and size). The results in Table 3 show that InsightEdit outperforms other metrics, except for a slight CLIP score difference in reasoning tasks, where it slightly trails SmartEdit. Furthermore, when trained on our AdvancedEdit dataset, InsightEdit demonstrates improvements across all metrics, highlighting the model's ability to leverage the dataset to enhance both comprehension and reasoning capabilities, while also emphasizing the intrinsic value of the data used.

5.3. Ablation Studies

Ablation Studies on IAA. To validate the effectiveness of IAA, we conduct comparative experiments on InsightEdit with the AdvancedEdit dataset. Table 4 shows the quantitive results, it can be seen that, with the IAA module, InsightEdit leads to notable improvements across various evaluation metrics. The qualitative results can be seen in Figure 5. In the first example, without the IAA module, the house is erroneously transformed into wildflowers, resulting in poor background consistency due to the absence of the red house. In contrast, with the IAA module, the red house is preserved accurately, maintaining consistency with the source image. In the second example, the marble table is edited poorly without the IAA module, while the addition of the IAA module leads to a more natural edit that aligns well with the instruction.

Ablation Studies on Instruction Complexity. We investigate the impact of instruction complexity and granularity on model performance by comparing two datasets introduced in Section 3.1: SimpleEdit Dataset and AdvancedEdit Dataset. Table 5 demonstrates that incorporating the SimpleEdit Dataset alone enhances the model's per-

formance, primarily due to the high-quality outputs generated by our mask-based image editing approach. However, the inclusion of the AdvancedEdit Dataset leads to substantial improvements, particularly in CLIPScore and VIEScore, highlighting the importance of increased instruction detail and the value of integrating complex instructions. The qualitative comparison results, as illustrated in Figure 6, reveal that without any data augmentation, the method struggles to understand the instructions and produce effective edits in both two examples. When the SimpleEdit dataset is introduced, there is some improvement, though the edit quality remains suboptimal. However, with the use of the AdvancedEdit dataset, the method demonstrates the ability to accurately interpret editing instructions, resulting in vivid and precise edits. For instance, in the first example, the method produces finely detailed orange and black butterfly wings, while in the second example, it accurately identifies and removes a light source, maintaining harmony in the surrounding content. These results further show that advanced instruction can handle more complex image editing scenarios and generate more vivid editing results.

6. Discussion and Conclusion

In conclusion, we present InsightEdit, an end-to-end instruction-based image editing method, utilizing a two-stream bridging mechanism to integrate both the textual and visual features reasoned by the powerful MLLM into the diffusion model to guide the image editing process more precisely. We design an automatic data construction pipeline to generate a large-scale, high-quality, and high-fidelity image editing dataset, considering both image and instruction perspectives. We hope that our efforts will mo-

tivate more researchers in the future.

Limitation and Future Work. Several challenges remain to be addressed, including the potential for incorporating more advanced MLLM for an improved understanding of instructions. Additionally, upgrading to more powerful diffusion models could potentially enhance the editing quality.

References

[1] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM transactions on graphics (TOG)*, 42 (4):1–11, 2023. 2

[2] 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. 1, 2, 3, 6

[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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22560–22570, 2023. 1

[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. 5

[6] Tuhin Chakrabarty, Kanishk Singh, Arkadiy Saakyan, and Smaranda Muresan. Learning to follow object-centric image editing instructions faithfully. *arXiv preprint arXiv:2310.19145*, 2023. 2, 3

[7] 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

[8] 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] Zigang Geng, Binxin Yang, Tiankai Hang, Chen Li, Shuyang Gu, Ting Zhang, Jianmin Bao, Zheng Zhang, Houqiang Li, Han Hu, et al. Instructdiffusion: A generalist modeling interface for vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12709–12720, 2024. 1, 3

[10] 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. 1, 3

[11] 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

[12] 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. 5

[13] Yi Huang, Jiancheng Huang, Yifan Liu, Mingfu Yan, Jiaxi Lv, Jianzhuang Liu, Wei Xiong, He Zhang, Shifeng Chen, and Liangliang Cao. Diffusion model-based image editing: A survey. *arXiv preprint arXiv:2402.17525*, 2024. 1

[14] Yuzhou Huang, Liangbin Xie, Xintao Wang, Ziyang Yuan, Xiaodong Cun, Yixiao Ge, Jiantao Zhou, Chao Dong, Rui Huang, Ruimao Zhang, et al. Smartedit: Exploring complex instruction-based image editing with multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8362–8371, 2024. 1, 2, 3, 5, 6, 7

[15] Mude Hui, Siwei Yang, Bingchen Zhao, Yichun Shi, Heng Wang, Peng Wang, Yuyin Zhou, and Cihang Xie. Hq-edit: A high-quality dataset for instruction-based image editing. *arXiv preprint arXiv:2404.09990*, 2024. 2

[16] Xuan Ju, Xian Liu, Xintao Wang, Yuxuan Bian, Ying Shan, and Qiang Xu. Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion. *arXiv preprint arXiv:2403.06976*, 2024. 1, 3

[17] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014. 6

[18] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[19] Jing Yu Koh, Daniel Fried, and Russ R Salakhutdinov. Generating images with multimodal language models. *Advances in Neural Information Processing Systems*, 36, 2024. 3, 5

[20] Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhu Chen. Viescore: Towards explainable metrics for conditional image synthesis evaluation. *arXiv preprint arXiv:2312.14867*, 2023. 4, 6

[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. 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. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 5

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 6

[24] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 5

[25] Ben Mann, N Ryder, M Subbiah, J Kaplan, P Dhariwal, A Neelakantan, P Shyam, G Sastry, A Askell, S Agarwal, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 1, 2020. 3

[26] 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. 6

[27] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024. 3

[28] 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

[29] 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. 3

[30] 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. 5

[31] Su Wang, Chitwan Saharia, Ceslee Montgomery, Jordi Pont-Tuset, Shai Noy, Stefano Pellegrini, Yasumasa Onoe, Sarah Laszlo, David J Fleet, Radu Soricut, et al. Imagen editor and editbench: Advancing and evaluating text-guided image inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18359–18369, 2023. 1, 2

[32] Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22428–22437, 2023.

[33] Shaoan Xie, Yang Zhao, Zhisheng Xiao, Kelvin CK Chan, Yandong Li, Yanwu Xu, Kun Zhang, and Tingbo Hou. Dreaminpainter: Text-guided subject-driven image inpainting with diffusion models. *arXiv preprint arXiv:2312.03771*, 2023. 2

[34] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. 6

[35] Ling Yang, Bohan Zeng, Jiaming Liu, Hong Li, Minghao Xu, Wentao Zhang, and Shuicheng Yan. Editworld: Simulating world dynamics for instruction-following image editing. *arXiv preprint arXiv:2405.14785*, 2024. 2, 3

[36] Senqiao Yang, Tianyuan Qu, Xin Lai, Zhuotao Tian, Bohao Peng, Shu Liu, and Jiaya Jia. An improved baseline for reasoning segmentation with large language model. *arXiv preprint arXiv:2312.17240*, 2023. 6

[37] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. 6

[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] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2, 3, 6

[40] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 3

[41] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6

[42] Shu Zhang, Xinyi Yang, Yihao Feng, Can Qin, Chia-Chih Chen, Ning Yu, Zeyuan Chen, Huan Wang, Silvio Savarese, Stefano Ermon, et al. Hive: Harnessing human feedback for instructional visual editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9026–9036, 2024. 2, 3

[43] Haozhe Zhao, Xiaojian Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. Ultraedit: Instruction-based fine-grained image editing at scale. *arXiv preprint arXiv:2407.05282*, 2024. 2, 3

[44] Junhao Zhuang, Yanhong Zeng, Wenran Liu, Chun Yuan, and Kai Chen. A task is worth one word: Learning with task prompts for high-quality versatile image inpainting. *arXiv preprint arXiv:2312.03594*, 2023. 3