ZONE: Zero-Shot Instruction-Guided Local Editing

Shanglin Li[1]*, **Bohan Zeng**[1]*, **Yutang Feng**[2]*, **Sicheng Gao**[1], **Xuhui Liu**[1]
Jiaming Liu[3], **Li Lin**[3], **Xu Tang**[3], **Yao Hu**[3], **Jianzhuang Liu**[4], **Baochang Zhang**[1,5,6]†
[1] Institute of Artificial Intelligence, Beihang University, Beijing, China
[2] Sino-French Engineer School, Beihang University, Beijing, China
[3] Xiaohongshu Inc [4] Shenzhen Institute of Advanced Technology, Shenzhen, China
[5] Zhongguancun Laboratory, Beijing, China [6] Nanchang Institute of Technology, Nanchang, China

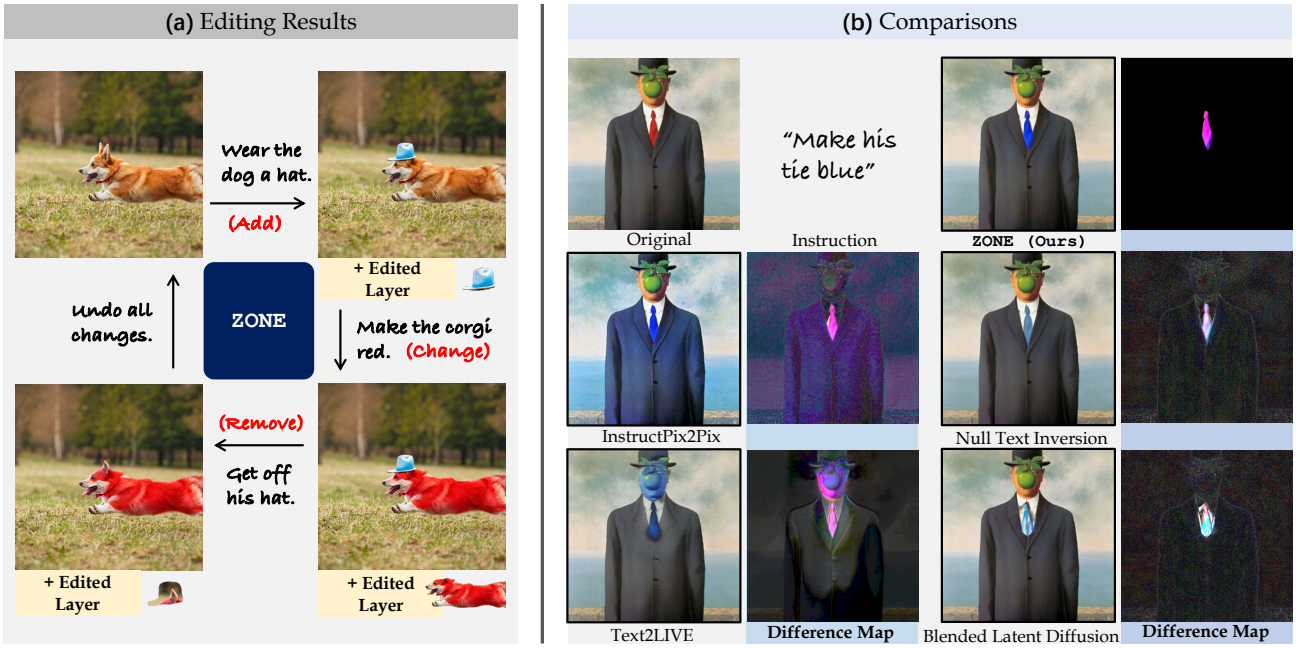

Figure 1. We propose ZONE, a zero-shot instruction-guided local editing approach. Our key idea is to edit and locate precise editing regions in an image with intuitive textual instructions. We demonstrate a multi-turn editing example in (a) and compare the difference maps between the edited image and the original image in (b) to highlight our method's ability for local editing.

Abstract

Recent advances in vision-language models like Stable Diffusion have shown remarkable power in creative image synthesis and editing. However, most existing text-to-image editing methods encounter two obstacles: First, the text prompt needs to be carefully crafted to achieve good results, which is not intuitive or user-friendly. Second, they are insensitive to local edits and can irreversibly affect non-edited regions, leaving obvious editing traces. To tackle these problems, we propose a Zero-shot instructiON-guided local image Editing approach, termed ZONE. We first convert the editing intent from the user-provided instruction (e.g., "make his tie blue") into specific image editing regions through InstructPix2Pix. We then propose a Region-IoU scheme for precise image layer extraction from an off-the-shelf segment model. We further develop an edge smoother based on FFT for seamless blending between the layer and the image.Our method allows for arbitrary manipulation of a specific region with a single instruction while preserving the rest. Extensive experiments demonstrate that our ZONE achieves remarkable local editing results and user-friendliness, outperforming state-of-the-art methods.

*These authors contributed equally.
†Corresponding Author: bczhang@buaa.edu.cn.

1. Introduction

Large-scale vision-language models, such as Stable Diffusion [44], DALL·E 2 [43], and Imagen [47], have revolutionized text-guided image editing by bridging the gap between natural language and image content. Trained on vast visual and textual data, these methods harness generative power to manipulate appearance and style in natural images, offering a wide array of possibilities for enhancing and manipulating images in domains such as photography, advertising, e-commerce, and social media. These advancements have opened up new possibilities for text-guided image editing, making it increasingly important in various applications.

State-of-the-art (SOTA) image generative techniques [38, 43, 44, 55] predominantly concentrate on stylization, where the desired appearance is determined by a reference image or textual description, often leading to global image alterations [25, 30, 49]. However, these methods often lack straightforward local editing capabilities, and the precise localization of these edits typically needs additional input guidance, such as segmentation masks [1, 14, 34], making text-driven editing cumbersome and potentially limiting its scope. Recent description-guided works[1] like Prompt-to-Prompt [15], DiffEdit [8], and Text2LIVE [3] make noteworthy contributions to mask-free local edits, but they either require complex textual descriptions (*e.g.,* Prompt-to-Prompt requires word-to-word alignment between the source image caption and the edited image caption, and DiffEdit uses query and reference prompts) or need to specify the edited object (*e.g.,* Text2LIVE asks for multiple prompts), which are not user friendly. Instruction-guided editing methods[2] [5, 11, 59, 62] present more elegant characteristics in this regard. They eliminate the need for image-anchored descriptions, requiring only descriptions of the desired edits (*e.g.*, "make it snowy"), which facilitates concise and intuitive expression. However, these methods suffer from the over-edit problem, potentially distorting high-frequency details in non-edited regions (see Fig. 1 (b)).

To tackle these problems, we propose ZONE, a **Z**ero-shot instructi**ON**-guided local image **E**diting approach. ZONE provides a more flexible and creative way to manipulate real images with layers.

Specifically, we leverage the pretrained instruction-guided model, InstructPix2Pix (IP2P) [5], for image editing. By exploring the attention mechanism of IP2P, we uncover the implicit associations between the editing locations and user-provided instructions in instruction-guided models. This allows us to identify the locations of the edited objects in instructions without the need for extra specification (*e.g.*, Stable Diffusion-based methods have to specify the tokens of the objects to edit). We further enhance this capability by proposing a Region-IoU scheme in conjunction with SAM [29], ensuring the mask refinement of the edited image layer. Our ZONE allows arbitrary image editing actions like "add", "remove", and "change", all accomplished with intuitive instructions. Additionally, ZONE supports multi-turn local editing without affecting non-edited regions, empowering high-fidelity local editing without any training or fine-tuning. Comprehensive experiments and user studies demonstrate that ZONE achieves remarkable results and user-friendliness in local image editing, outperforming existing SOTA methods.

To summarize, we make the following key contributions:

- We propose ZONE, a zero-shot image local editing method that enables users to edit localized regions of both real and synthetic images with simple instructions. ZONE preserves non-edited regions without loss and allows arbitrary manipulation of edited image layers.
- We reveal and exploit the different attention mechanisms between IP2P and Stable Diffusion when processing user instructions for image editing, with intuitive visual comparisons.
- We present a novel Region-IoU scheme and incorporate it with SAM for effective edited region refinement, and introduce a Fourier transform-based edge smoother to reduce the artifacts when compositing the image layers.
- Comprehensive experiments and user studies demonstrate that ZONE achieves high-fidelity local editing results without any auxiliary prompts, outperforming SOTA methods in photorealism and content preservation.

2. Related Work

2.1. Generative Models for Image Manipulation

Image manipulation is a fundamental process within the realm of computer vision, involving altering images with the aid of additional conditions like textual prompts, labels, masks, or reference images. Two mainstream editing methods include Generative Adversarial Networks (GANs) and Diffusion Models (DMs). Typical image manipulation tasks comprise image-to-image translation [7, 10, 20, 26, 46, 49, 54, 63], super-resolution [13, 21, 31, 56], inpainting [19, 34, 41, 44], colorization [4, 33, 37, 53], and more. Although GAN-based methods excel when dealing with carefully curated data, they struggle with extensive and heterogeneous datasets [22, 23, 36]. To enhance generative expressiveness, [17, 18, 44, 50, 51, 57, 58] utilize DMs to achieve high-quality generation over diverse datasets. Recent research has yielded promising generation outcomes through the training or fine-tuning of large-scale text-to-image models [5, 24, 35, 38, 43, 47, 55], as well as by harnessing CLIP [42] embeddings to guide image manipulation using textual prompts [9, 25, 30]. Some prior works

[1] In this paper, we call them description-guided diffusion models.
[2] In this paper, we call them instruction-guided diffusion models.

[1, 2, 15, 40, 52] also demonstrate the zero-shot editing capability of pretrained DMs. Similarly, our method extensively exploits a pretrained DM's generative capability to facilitate diverse and stylized image editing. However, we uniquely explore the implicit relationship between the DM's editing regions during generation and the whole user instructions, enabling fine-grained layer-specific positioning.

2.2. Localized Image Editing

Several recent works have made attempts at localized image editing. Blend Diffusion [1] proposes a mask-guided method by blending edited regions with the other parts of the image at different noise levels along the diffusion process. Text2LIVE [3] introduces an RGBA layer generation approach with a CLIP-supervised generator for performing edits of objects in real images and videos. Prompt-to-Prompt [15] controls the spatial layouts of the image corresponding to the words in the prompt through cross-attention modification, enabling local edits by modifying textual prompts. Pix2Pix-Zero [40] preserves the structure of the original image with cross-attention guidance and applies an edit-direction embedding to make changes to localized objects. Instruction-based editing methods like IP2P [5] and MagicBrush [59] are trained or finetuned on triplet datasets to realize intuitive high-quality image editing based on user-provided instructions. PAIR-diffusion [14] allows editing the structure and appearance of each masked part in the original image independently. While these methods produce impressive results within their specific applications, they compromise on local image editing: instruction-guided methods [5, 59] and attention-based modifications [15, 40] introduce artifacts to non-edited regions, mask-based methods [1, 14] add complexity to user interactions, and CLIP-based methods [3, 40] sacrifice the flexibility of natural language editing. In contrast, our ZONE requires only a single instruction to achieve high-fidelity local image editing with an image layer.

2.3. Instruction-Guided Editing

Despite the significant progress of text-to-image models, most require detailed textual descriptions [38, 43–45, 47] to convey the desired image content, often falling short of user expectations for image editing. In contrast, direct instruction-guided modifications of target regions/attributes offer a more intuitive and convenient approach, such as "make the girl smile" and "give him a ball." Recent advancements in instruction-guided editing and generation [5, 11, 39, 59, 61, 62] have made notable progress. For instance, IP2P [5] employs GPT-3 [6] and Prompt-to-Prompt [15] to synthesize an instruction-editing dataset, utilizes a pretrained Stable Diffusion model [44] for weight initialization, and trains a diffusion model specialized in instruction-guided editing. MagicBrush [59] fine-tunes IP2P using a real image dataset, thereby demonstrating a superior performance in instruction-guided editing. In this paper, we aim to leverage the instruction-editing capability of these pretrained instruction-guided diffusion models to eliminate the need for additional masks in previous local editing approaches [1, 2, 38], enabling flexible and high-fidelity local editing based on a single user-provided instruction.

3. Preliminaries

Diffusion Models. Diffusion models [17, 48, 50] are probabilistic generative models founded on two complementary stochastic processes: *diffusion* and *denoising*. The *diffusion* process progressively adds different amounts of Gaussian noise to a clean image x_0 towards Gaussian distribution $x_T \sim \mathcal{N}(0, I)$ in T timesteps: $x_t = \sqrt{\alpha_t} x_0 + \sqrt{1-\alpha_t}\epsilon$, where α_t defines the level of noise, and $\epsilon \sim \mathcal{N}(0, I)$.

In the *denoising* process, a neural network ϵ_θ is designed to predict the noise ϵ for x_t to get a "cleaner" image gradually. This process is achieved by minimizing the denoising objective: $\mathcal{L} = \mathbb{E}_{x_0,t,\epsilon} \|\epsilon - \epsilon_\theta(x_t, t)\|_2^2$. Rombach et al. [44] introduce a latent diffusion model (LDM), which speeds up both processes by reducing images into a lower-dimensional latent space utilizing a variational auto-encoder [28]. This advancement has underpinned the achievements of Stable Diffusion, serving as the fundamental model for many diffusion-based works.

InstructPix2Pix. InstructPix2Pix [5] (IP2P) is a pioneering conditional diffusion model that edits images from user-provided instructions. Specifically, IP2P constructs an instruction dataset to fine-tune the pretrained Stable Diffusion. Given a target image x, an image condition c_I, and a textual instruction condition c_T, IP2P projects x to the latent $z = \mathcal{E}(x)$ with a pretrained encoder $\mathcal{E}$, and then fine-tunes Stable Diffusion by minimizing the following objective:

$$\mathcal{L} = \mathbb{E}_{\mathcal{E}(x),\mathcal{E}(c_I),c_T,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta\left(z_t, t, \mathcal{E}\left(c_I\right), c_T\right)\|_2^2\right], \tag{1}$$

where the denoising network ϵ_θ accepts two input conditions and predicts the noise ϵ. IP2P also finds it beneficial to perform classifier-free guidance [16] concerning both conditions, thus controlling the strength of edit by image guidance scale s_I and instruction guidance scale s_T:

$$\begin{aligned}\tilde{\epsilon_\theta}\left(z_t, c_I, c_T\right) =& \epsilon_\theta\left(z_t, \varnothing, \varnothing\right) \\ &+ s_I \cdot \left(\epsilon_\theta\left(z_t, c_I, \varnothing\right) - \epsilon_\theta\left(z_t, \varnothing, \varnothing\right)\right) \\ &+ s_T \cdot \left(\epsilon_\theta\left(z_t, c_I, c_T\right) - \epsilon_\theta\left(z_t, c_I, \varnothing\right)\right).\end{aligned} \tag{2}$$

At inference time, IP2P can modify an image with a user-provided instruction and trade-off the generated sample ac-

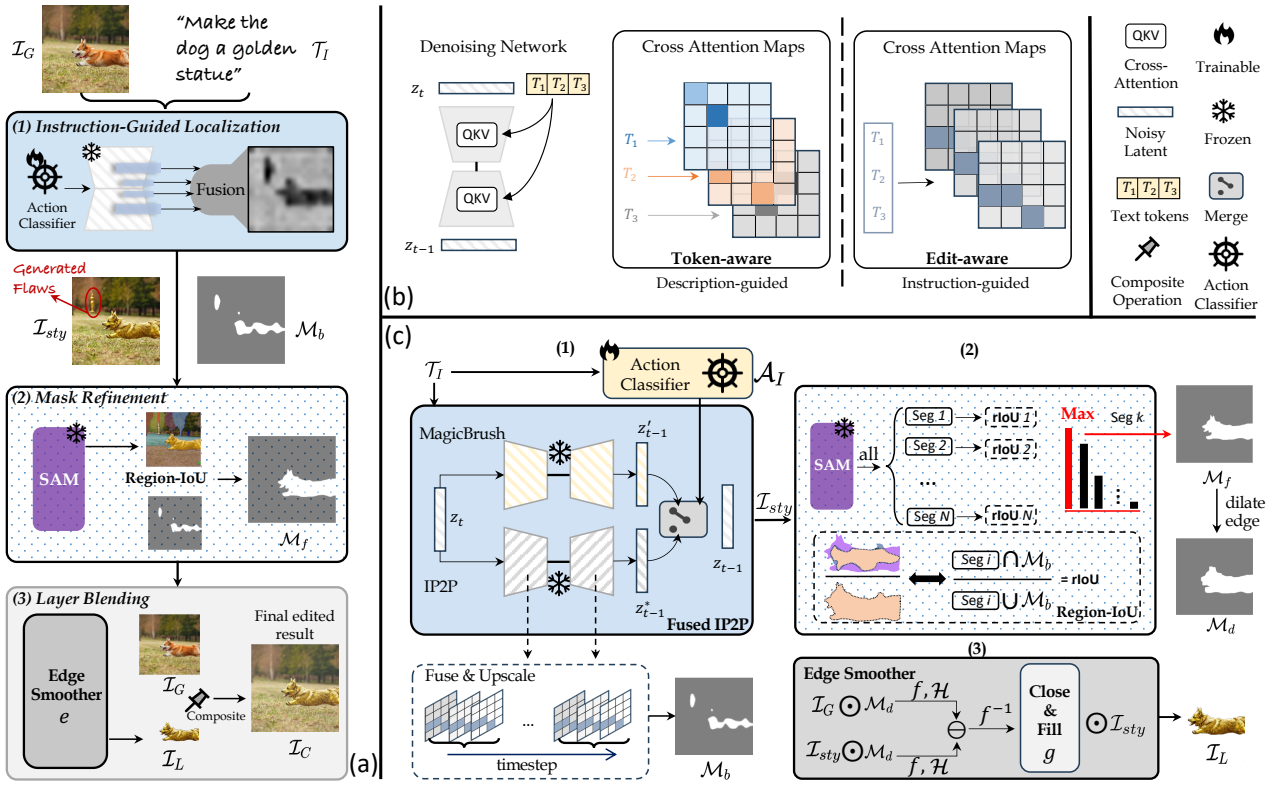

Figure 2. **Overview of ZONE.** (a) Three modules in ZONE. (b) Distinct difference between description-guided and instruction-guided diffusion models on cross-attention. The former usually follows a *token-aware* format, while the latter is *edit-aware*. (c) Implementation details of the modules shown in (a).

cording to the strengths of the guidance image and the edit instruction through s_I and s_T.

4. Method

Overview of ZONE. We aim to make localized edits on an image with simple instructions. As depicted in Fig. 1 (a), such edits include performing three primary actions: (i) "add": add an object to the image without specifying location with user-provided masks; (ii) "remove": remove the object in the scene; (iii) "change": change the style (*i.e.,* texture) of an existing object or replace the object with another object. Additionally, our method allows high-fidelity multi-turn edits with a series of instructions.

As outlined in Fig. 2 (a), our approach consists of the following steps: First, we train an action classifier for steering different editing requirements and concurrently generate and position the editing region using a fused IP2P, as detailed in Section 4.2 and Fig. 2 (c). Second, we devise a mask refinement module for an edited image layer in Section 4.3. Finally, in Section 4.4, we propose an FFT-based edge smoother for seamless blending of the edited image layer with the original image.

4.1. Problem Statement

Given an RGB image $\mathcal{I}_G \in \mathbb{R}^{3\times H\times W}$ and a textual instruction $\mathcal{T}_I$, we aim to locate and edit image regions following $\mathcal{T}_I$ and maintain the original non-edited regions. Inspired by Text2LIVE [3], we extract an edited layer $\mathcal{I}_L$ with color and opacity that are composited over $\mathcal{I}_G$. As opposed to previous works [1, 3, 15, 40], we neither rely on any user-defined mask nor need non-intuitive prompt engineering, realizing precise local editing and seamless layer blending.

4.2. Instruction-Guided Localization

Many local editing methods require users to explicitly specify the object they want to edit with a prompt or a mask [1, 3, 8, 40]. This is not intuitive and often requires a certain learning cost. Our approach locates and edits the implicitly designated object from the user's instruction. For example, a user-provided instruction like "make her old" can implicitly convey the user's editing intent to modify the woman in the scene (*locate*) by making her appear older (*edit*).

As shown in Fig. 2 (b), our key finding is that the operational mechanisms of instruction-guided and description-guided diffusion models on cross-attention exhibit a distinct difference. Specifically, we empirically demonstrate that: (i) a description-guided model displays a *token-aware* char-

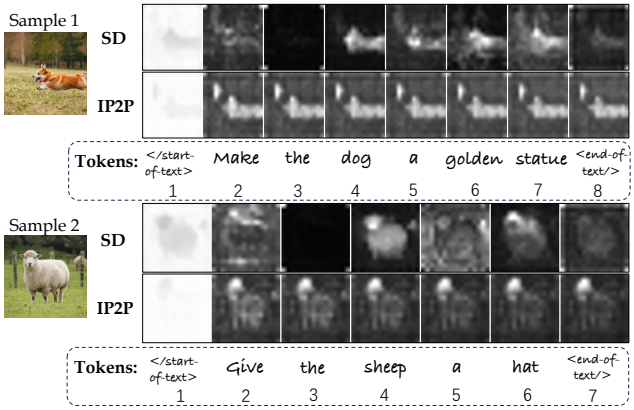

Figure 3. **Token-wise cross-attention map difference.** We average the cross-attention maps among all timesteps for each sample. IP2P shows consistency in all tokens, while Stable Diffusion (SD) demonstrates a one-to-one correspondence with tokens.

acteristic on its cross-attention maps, associating each input text token with a corresponding spatial structure; (ii) an instruction-guided model's cross-attention maps share similar spatial features, demonstrating an *edit-aware* characteristic, being insensitive to single tokens but responsive to the overall editing intent.

Given a noisy latent z_t and a textual embedding c_T, the denoising UNet ϵ_θ predicts the noise ϵ at each timestep t. The generation is conditioned on the textual prompt $\mathcal{T}_I$ by computing cross-attention between the textual embedding c_T and the spatial features $\phi(z_t)$, and updates $\phi(z_t)$ as $\hat{\phi}(z_t)$:

$$M = \text{Softmax}(\frac{QK^T}{\sqrt{d}}),\ \hat{\phi}(z_t) = M \cdot V, \tag{3}$$

where the query $Q = W_Q\phi(z_t)$, key $K = W_K c_T$, and value $V = W_V c_T$ are obtained with linear projections W_Q, W_K, and W_V. $M \in \mathbb{R}^{H^{'} \times W^{'} \times L}$ contains L cross-attention maps that are correlated to the similarity between Q and K. Typically $H^{'}$ and $W^{'}$ are $1/32$ of the original image size H and W in Stable Diffusion. For the description-guided Stable Diffusion model, let M^l be the attention map of the l-th token, $l \in \{1, 2, \ldots, L\}$. For the instruction-guided IP2P, M^l shares a uniform characteristic across all tokens, concentrated directly at the edited location without token specification, as visualized in Fig. 3.

Based on this finding, we devise a simple yet effective localization module that semantically locates the edited region with instruction $\mathcal{T}_I$. Specifically, we first collect the attention maps of the denoising model of IP2P from all timesteps of the denoising process. Then, we average and resize the maps to obtain averaged attention maps $\mathcal{M}_A \in \mathbb{R}^{H \times W \times L}$. Note that the L tokens include the "start-of-text" and "end-of-text" tokens, whose corresponding attention maps are M^1 and M^L, respectively. As depicted in Fig. 3, the attention weights (*i.e.,* the brightness of the pixels) of these maps tend to decrease along the tokens, so we subtract the last token's cross-attention map from the first token's and binarize the result with a fixed threshold T:

$$\mathcal{M}_b(m,n) = \begin{cases} 1, \text{ if } \mathcal{M}_A^1(m,n) - \mathcal{M}_A^L(m,n) < T, \\ 0, \text{ others,} \end{cases} \tag{4}$$

where T is empirically set to 128. This yields a rough, noise-filtered edited region mask $\mathcal{M}_b$ most related to $\mathcal{T}_I$ (see Fig. 2 (a)).

Moreover, we find that IP2P performs not as well as MagicBrush in the "remove" editing but preserves better object identity in terms of "add" and "change". Therefore, we design a fused IP2P module with a trainable action classifier $\mathcal{A}_I$. As illustrated in Fig. 2 (c), we lock the weights of both IP2P and MagicBrush and use a pretrained action classifier $\mathcal{A}_I$ to steer the denoising process based on $\mathcal{T}_I$:

$$z_{t-1} = (z^*_{t-1} + \beta \cdot z'_{t-1})/(1+\beta), \tag{5}$$

where z^*_{t-1} and z'_{t-1} are the denoised latents by IP2P and MagicBrush, respectively. β is a hyperparameter to control the guidance strength of MagicBrush on IP2P, empirically set to 0.2 if $\mathcal{A}_I(\mathcal{T}_I)$ is classified to "remove" and 0.01 for other actions. This module generates a globally edited image $\mathcal{I}_{sty}$ according to $\mathcal{T}_I$. $\mathcal{I}_{sty}$ serves as the canvas, from which the edited region is cropped out to form a separate image layer in the following steps.

4.3. Mask Refinement

The location mask $\mathcal{M}_b$ and $\mathcal{I}_{sty}$ obtained in Section 4.2 are insufficient for precise local editing, since $\mathcal{M}_b$ only indicates the general location of the edited region, as illustrated in Fig. 2 (a). An intuitive and effective mask refinement method is to use an off-the-shelf segmentation model. We leverage the Segment Anything Model (SAM) [29] to generate precise masks of the canvas $\mathcal{I}_{sty}$ at various levels. However, we do not use SAM's preset point or box prompts for segmentation selection, because these prompts could potentially lead to misselection or omission of SAM's segmentation results due to IP2P's over-edit problem (which is also reflected in $\mathcal{M}_b$, see $\mathcal{I}_{sty}$ and $\mathcal{M}_b$ in Fig. 2 (a)), resulting in a final mask that does not accurately reflect $\mathcal{T}_I$'s editing intention. Therefore, we propose a Region-IoU (rIoU) scheme to obtain the accurate segmentation mask.

As depicted in Fig. 2 (c), by sending $\mathcal{I}_{sty}$ to SAM, we extract all the possible instance segments $\mathcal{S} = \{\mathcal{S}^j\}_{j=1}^N$. Note that $\mathcal{S}$ contains the segments from all levels of SAM's segmentation. We define rIoU $\mathcal{R}(j)$ as:

$$\mathcal{R}(j) = \frac{\text{area}(\mathcal{S}^j \cap \mathcal{M}_b)}{\text{area}(\mathcal{S}^j \cup \mathcal{M}_b)},\ j = 1, 2, \ldots, N. \tag{6}$$

Figure 4. **Visualization and ablation.** The first 4 columns show the intermediate results related to the edge smoother. The last column compares the final edited results with and without the edge smoother.

If $k = \underset{j=1,2,...,N}{\arg\max}\{\mathcal{R}(j)\}$, then we obtain the refined mask $\mathcal{M}_f = \mathcal{S}^k$. One example is shown in Fig. 2 (a) or (c).

4.4. Layer Blending

After the mask refinement, we obtain an edited image layer $\mathcal{I}_L' = \mathcal{I}_{sty} \odot \mathcal{M}_f$, which retains the color information of $\mathcal{I}_{sty}$ within the region where $\mathcal{M}_f = 1$, with the rest being transparent. A naïve way to get the final edited result $\mathcal{I}_C$ is to stitch $\mathcal{I}_L'$ and the original image $\mathcal{I}_G$ at pixel-level. This fundamentally tackles the over-edit problem encountered in instruction-guided methods for local editing. Nevertheless, directly pasting $\mathcal{I}_L'$ back to $\mathcal{I}_G$ may result in noticeable artifacts, such as jagged edges and incomplete coverage of the edited region in the original image, as indicated by the yellow arrows in Fig. 4 (b).

We tackle this problem by designing a novel edge smoother with Fast Fourier Transform (FFT). Given the original image $\mathcal{I}_G$, the canvas $\mathcal{I}_{sty}$, and the refined location mask $\mathcal{M}_f$, we first dilate $\mathcal{M}_f$ to $\mathcal{M}_d$ to incorporate more edge information in $\mathcal{I}_{sty}$ that may not be included in $\mathcal{I}_L'$. Then we get the dilated edited image layer $\mathcal{I}_{L,d} = \mathcal{I}_{sty} \odot \mathcal{M}_d$ and the dilated original image layer $\mathcal{I}_{G,d} = \mathcal{I}_G \odot \mathcal{M}_d$, as shown in the second column of Fig. 4. The edge smoother e is defined by:

$$e(\mathcal{I}_{L,d}, \mathcal{I}_{G,d}) = g(f^{-1}(\mathcal{H}(f(\mathcal{I}_{L,d})) - \mathcal{H}(f(\mathcal{I}_{G,d})))), \quad (7)$$

where g is a composition of binarization and morphological closing and filling functions, f and f^{-1} represent FFT and inverse FFT, respectively, and $\mathcal{H}$ is an ideal low-pass filter:

$$\mathcal{H}(f_s) = \begin{cases} f_s(c), & \text{if } \|c - c_0\|_2 \leq D_0, \\ 0, & \text{if } \|c - c_0\|_2 > D_0, \end{cases} \quad (8)$$

where $f_s \in \mathbb{R}^{H \times W}$ is the frequency spectrum of the image transformed by f, c is the coordinate in f_s, c_0 is the center coordinate of f_s, and D_0 is set empirically to 200 for a 512×512 image. We use the edge smoother e to get the final mask $\mathcal{M}_f^*$.

As shown in the second column of Fig. 4, we observe that both $\mathcal{I}_{G,d}$ and $\mathcal{I}_{L,d}$ share similar low-frequency characteristics on non-edited regions (*e.g.,* background), but they hold different low-frequency characteristics on the edited regions (*e.g.,* hat and the shadow below it). Therefore, we can exclude the non-edited regions and retain the edited regions by subtracting the low-frequency components between $\mathcal{I}_{L,d}$ and $\mathcal{I}_{G,d}$ in the frequency domain: $d_s = \mathcal{H}(f(\mathcal{I}_{L,d})) - \mathcal{H}(f(\mathcal{I}_{G,d}))$ and invert it back to the image domain to get the difference mask $\mathcal{M}_{dm} = f^{-1}(d_s)$. The final mask $\mathcal{M}_f^*$ is then obtained by $\mathcal{M}_f^* = g(\mathcal{M}_{dm}) = e(\mathcal{I}_{L,d}, \mathcal{I}_{G,d})$. Finally, we get the final edited image layer $\mathcal{I}_L$ by $\mathcal{I}_L = \mathcal{I}_{sty} \odot \mathcal{M}_f^*$, and the final edited result $\mathcal{I}_C$ is acquired by compositing $\mathcal{I}_G$ and $\mathcal{I}_L$. The intuitive visualization of these intermediate results are shown in Fig. 4.

The implementation details and more discussions can be found in the supplementary material.

5. Experiments

5.1. Experimental Setup

Baselines. We conduct comprehensive experiments for the local editing task by comparing ZONE with five state-of-the-art image editing methods that are capable of local editing: *Text2LIVE* [3], *DiffEdit* [8], *IP2P* [5], *Pix2Pix-Zero* [40], and *MagicBrush* [59]. The implementation of these methods can be found in the supplementary material.

Datasets. We randomly select and annotate 100 samples for evaluation, including 60 real images from the Internet and 40 synthetic images. To ensure the representativeness of the evaluation, we consider the diversity of scenes and objects in the sample selection. In particular, we divide the test set into three categories: 32 images for "add", 54 for "change", and 14 for "remove" actions. All these 100 images are listed in the supplementary material.

Evaluation Metrics. Following [5, 59], we perform qualitative and quantitative comparisons using a variety of evaluation metrics. Learned Perceptual Image Patch Similarity (LPIPS) [60] is used to quantify the perceptual similarity between the original and edited image. CLIP text-image similarity (CLIP-T) [12] is employed to assess the alignment between the edited image and its corresponding caption, and CLIP image similarity (CLIP-I) is used to evaluate the layout similarity and semantic correlation between the edited image and the original image, serving as a reliable indicator of the edited image's quality. We also use L1 and L2 distances for pixel-level difference comparison.

Type	Methods	L1 ↓	L2 ↓	LPIPS ↓	CLIP-I ↑	CLIP-T ↑
Description-guided	DiffEdit [8]	0.0426	0.0099	0.1695	0.8947	0.2815
	Text2LIVE [3]	0.0511	0.0075	0.2176	0.9075	**0.3062**
	Pix2Pix-Zero [40]	0.1198	0.0342	0.4375	0.7679	0.2701
Instruction-guided	InstructPix2Pix [5]	0.0945	0.0274	0.2816	0.9089	0.2907
	MagicBrush [59]	0.0919	0.0378	0.2903	0.8959	0.2939
	ZONE (Ours)	**0.0146**	**0.0061**	**0.0441**	**0.9688**	0.2969

Table 1. **Quantitative evaluation.** We use L1 and L2 to gauge pixel-level structural similarity, LPIPS and CLIP-I to evaluate image quality, and CLIP-T to assess text-image semantic similarity. The best and the second best results are marked in **bold** and underline, respectively.

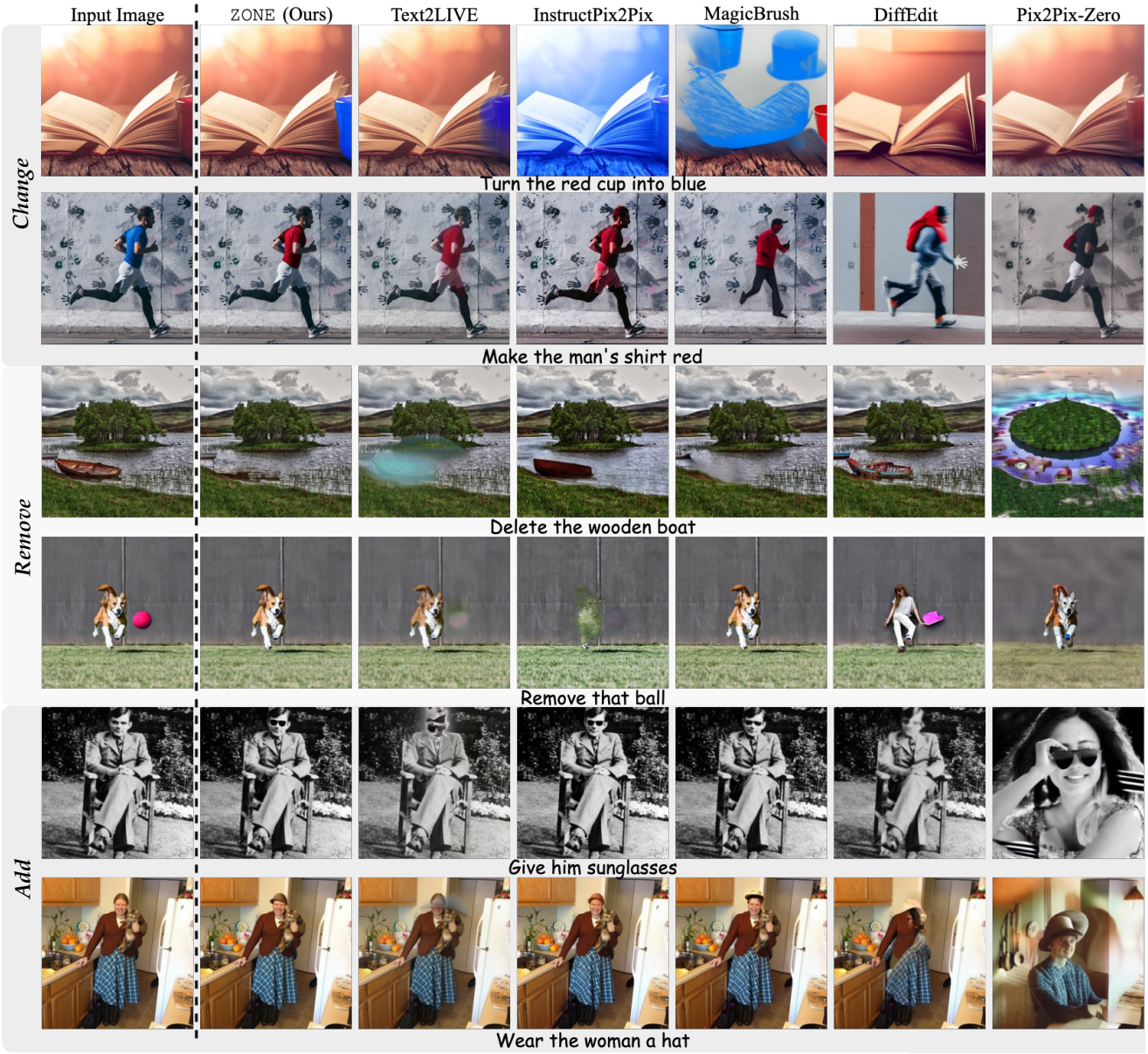

Figure 5. **Qualitative comparison.** We compare the editing efficacy of our ZONE with existing SOTA methods. The instructions (or instructions that are equivalent to the descriptions required by some baselines) used for editing are written below each row of the images.

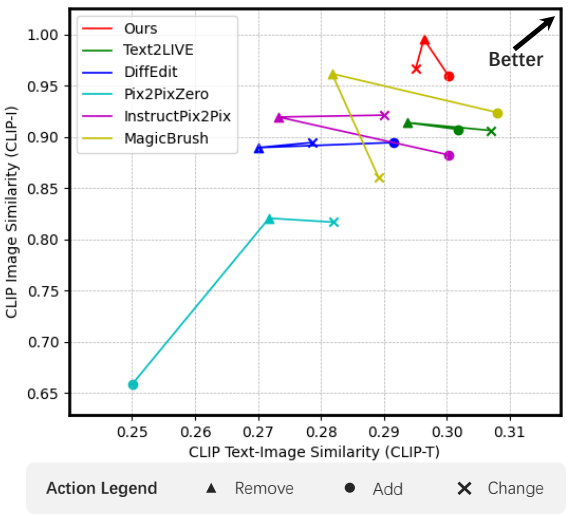

Figure 6. **Stability analysis.** We categorize the test set into three actions ("Remove", "Add", and "Change") and calculate their respective CLIP-I and CLIP-T values. Our method achieves the best quality-stability trade-off for all actions.

Figure 7. **Detailed comparison.** We show a zoomed-in sample where ZONE effectively resolves the over-edit problem.

5.2. Comparisons

Quantitative Evaluation. As shown in Table 1, we measure the models with the five metrics. The quantitative results indicate the following: (i) Our method significantly outperforms our counterparts on metrics related to image structure and quality, implying the efficacy of ZONE's preservation of the non-edited regions. (ii) Text2LIVE performs best on CLIP-T, but the qualitative comparison in Fig. 5 does not support this result. We surmise that Text2LIVE performs better on this metric potentially due to its direct supervision by CLIP.

To quantify the stability of the edits, we divide the test set into three action groups: "change", "add", and "remove". We then test the CLIP-I and CLIP-T metrics for each model and plot the CLIP curves in Fig. 6, where the performances of the same method on these actions are connected with lines of the same color. Our interpretation is as follows: first, the shorter the projection of the line on the axis, the higher the semantic *stability* (*i.e.,* maintaining similar performances under different editing instructions) of the image editing; second, if the curve is closer to the upper right corner, it indicates that the method's editing *quality* is more superior. Our method achieves the best trade-off between *quality* and *stability*, demonstrating strong editing stability and representativeness.

Methods	SR (%)	UPR (%)
DiffEdit [8]	27.0 ± 5.6	8.8
Text2LIVE [3]	32.3 ± 4.3	17.3
Pix2Pix-Zero [40]	18.3 ± 4.3	10.4
InstructPix2Pix [5]	60.6 ± 4.9	18.9
MagicBrush [59]	50.0 ± 5.1	18.0
ZONE (Ours)	**69.0** ± **3.7**	**26.6**

Table 2. **Human evaluation.** Our ZONE obtains the highest success rate (SR) and user preference rate (UPR).

Qualitative Comparsion. In Fig. 5, we illustrate the editing results for the baselines and our method. We select six sets of images (including synthetic and real images) and group them based on actions. Our ZONE shows precise local editing capability while preserving the remaining pixels, this is especially important when there are perceptually important high-frequency details, such as faces, textures, or texts. A zoomed-in comparison is shown in Fig. 7. Both InstructPix2Pix and Text2LIVE introduce distortions to the non-edited areas during the editing process. For instance, InstructPix2Pix distorts the nearby clock and paints the orange outside of the basket red. In comparison, Text2LIVE maintains a better structure but generates a "barrel" of apples and introduces an obvious foggy effect to the image. Our method, however, can clearly distinguish between the edited region and the non-edited regions, demonstrating the best local editing efficacy.

5.3. Human Evaluation

Due to the lack of an effective metric to measure editing effects (mainly due to the absence of ground truth images after editing), the metrics mentioned in Section 5.1 alone are not sufficient to demonstrate the superiority of our method over existing ones. To further validate the editing effects of ZONE, in addition to the visual comparison in Fig. 5, we also conduct a human evaluation to calculate the success rate (SR) and user preference rate (UPR) of the edited images with the editing instructions. Table 2 shows a consistent preference for our method by users, as well as a dominant success rate over other methods.

Please refer to our supplementary material for more visualizations and details of this user study.

6. Conclusion

We present ZONE, a zero-shot instruction-guided local image editing approach, which leverages the localization capability within the pre-trained instruction-guided diffusion

models. Our approach innovatively utilizes the editing intent regions inherent in the instructions, rather than focusing on individual tokens, eliminating the need for specific guidance. By integrating the Region-IoU scheme and FFT-based edge smoother with a pretrained segmentation model, ZONE effectively realizes precise local editing. Comprehensive experiments and user studies further demonstrate the superiority of ZONE over SOTA methods.

References

[1] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *CVPR*, 2022. 2, 3, 4

[2] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *TOG*, 2023. 3

[3] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *ECCV*, 2022. 2, 3, 4, 6, 7, 8, 12, 13, 14

[4] Marc Górriz Blanch, Marta Mrak, Alan F Smeaton, and Noel E O'Connor. End-to-end conditional gan-based architectures for image colourisation. In *MMSPW*, 2019. 2

[5] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023. 2, 3, 6, 7, 8, 12, 13, 14

[6] 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. In *NeurIPS*, 2020. 3

[7] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, 2018. 2

[8] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022. 2, 4, 6, 7, 8, 12, 17

[9] Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. Vqgan-clip: Open domain image generation and editing with natural language guidance. In *ECCV*, 2022. 2

[10] Xiaoyue Duan, Shuhao Cui, Guoliang Kang, Baochang Zhang, Zhengcong Fei, Mingyuan Fan, and Junshi Huang. Tuning-free inversion-enhanced control for consistent image editing. *arXiv preprint arXiv:2312.14611*, 2023. 2

[11] Alaaeldin El-Nouby, Shikhar Sharma, Hannes Schulz, Devon Hjelm, Layla El Asri, Samira Ebrahimi Kahou, Yoshua Bengio, and Graham W Taylor. Tell, draw, and repeat: Generating and modifying images based on continual linguistic instruction. In *CVPR*, 2019. 2, 3

[12] Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *TOG*, 2022. 6

[13] Sicheng Gao, Xuhui Liu, Bohan Zeng, Sheng Xu, Yanjing Li, Xiaoyan Luo, Jianzhuang Liu, Xiantong Zhen, and Baochang Zhang. Implicit diffusion models for continuous super-resolution. In *CVPR*, 2023. 2

[14] Vidit Goel, Elia Peruzzo, Yifan Jiang, Dejia Xu, Nicu Sebe, Trevor Darrell, Zhangyang Wang, and Humphrey Shi. Pair-diffusion: Object-level image editing with structure-and-appearance paired diffusion models. *arXiv preprint arXiv:2303.17546*, 2023. 2, 3

[15] 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. 2, 3, 4

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

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 2, 3

[18] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *JMLR*, 2022. 2

[19] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *TOG*, 2017. 2

[20] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 2

[21] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *CVPR*, 2023. 2

[22] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 2

[23] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020. 2

[24] 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 *CVPR*, 2023. 2

[25] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *CVPR*, 2022. 2

[26] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *ICML*, 2017. 2

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 12

[28] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3

[29] 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. 2, 5

[30] Gihyun Kwon and Jong Chul Ye. Clipstyler: Image style transfer with a single text condition. In *CVPR*, 2022. 2

[31] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017. 2

[32] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022. 15

[33] Jianxin Lin, Peng Xiao, Yijun Wang, Rongju Zhang, and Xiangxiang Zeng. Diffcolor: Toward high fidelity text-guided image colorization with diffusion models. *arXiv preprint arXiv:2308.01655*, 2023. 2

[34] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022. 2

[35] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. 2

[36] Ron Mokady, Omer Tov, Michal Yarom, Oran Lang, Inbar Mosseri, Tali Dekel, Daniel Cohen-Or, and Michal Irani. Self-distilled stylegan: Towards generation from internet photos. In *SIGGRAPH*, 2022. 2

[37] Kamyar Nazeri, Eric Ng, and Mehran Ebrahimi. Image colorization using generative adversarial networks. In *AMDO*, 2018. 2

[38] Alexander Quinn 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. In *ICML*, 2022. 2, 3

[39] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022. 3

[40] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *SIGGRAPH*, 2023. 3, 4, 6, 7, 8, 12

[41] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 2

[42] 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 *ICML*, 2021. 2, 12, 17

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

[44] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2, 3, 15

[45] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, 2023. 3

[46] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *SIGGRAPH*, 2022. 2

[47] 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. In *NeurIPS*, 2022. 2, 3

[48] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 3

[49] Kihyuk Sohn, Nataniel Ruiz, Kimin Lee, Daniel Castro Chin, Irina Blok, Huiwen Chang, Jarred Barber, Lu Jiang, Glenn Entis, Yuanzhen Li, et al. Styledrop: Text-to-image generation in any style. *arXiv preprint arXiv:2306.00983*, 2023. 2

[50] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2, 3

[51] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, 2019. 2

[52] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *CVPR*, 2023. 3

[53] Hanzhang Wang, Deming Zhai, Xianming Liu, Junjun Jiang, and Wen Gao. Unsupervised deep exemplar colorization via pyramid dual non-local attention. *TIP*, 2023. 2

[54] Tengfei Wang, Ting Zhang, Bo Zhang, Hao Ouyang, Dong Chen, Qifeng Chen, and Fang Wen. Pretraining is all you need for image-to-image translation. *arXiv preprint arXiv:2205.12952*, 2022. 2

[55] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022. 2

[56] Zongsheng Yue, Jianyi Wang, and Chen Change Loy. Resshift: Efficient diffusion model for image super-resolution by residual shifting. *arXiv preprint arXiv:2307.12348*, 2023. 2

[57] Bohan Zeng, Shanglin Li, Yutang Feng, Hong Li, Sicheng Gao, Jiaming Liu, Huaxia Li, Xu Tang, Jianzhuang Liu, and Baochang Zhang. Ipdreamer: Appearance-controllable 3d object generation with image prompts. *arXiv preprint arXiv:2310.05375*, 2023. 2

[58] Bohan Zeng, Shanglin Li, Xuhui Liu, Sicheng Gao, Xiaolong Jiang, Xu Tang, Yao Hu, Jianzhuang Liu, and Baochang Zhang. Controllable mind visual diffusion model. *arXiv preprint arXiv:2305.10135*, 2023. 2

[59] 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, 6, 7, 8, 12, 13, 14

[60] 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, 16

[61] 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. *arXiv preprint arXiv:2303.09618*, 2023. 3

[62] Tianhao Zhang, Hung-Yu Tseng, Lu Jiang, Weilong Yang, Honglak Lee, and Irfan Essa. Text as neural operator: Image manipulation by text instruction. In *ACMMM*, 2021. 2, 3

[63] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 2

Appendix

In this supplementary, we first give more visualization results, then detail the datasets and the implementation, and finally state the social impact and the limitations.

A. More Visualizations

In this section, we first present more visualizations of the samples from the test set under two comparison settings: (i) single-turn editing, and (ii) multi-turn editing. To make the comparison more representative, we compare our ZONE with three state-of-the-art (SOTA) text-to-image approaches, Text2LIVE (T2L) [3], InstructPix2Pix (IP2P) [5], and MagicBrush (MB) [59]. Then we conduct an ablation study to show the efficacy of our fused IP2P module, through cross-attention map visualization.

A.1. Single-Turn Editing Examples

We show more single-turn editing examples to further validate ZONE's remarkable ability of local image editing. In particular, we compare it with the other methods for local editing using 9 images and their corresponding instructions (or prompts equivalent to the instructions). As evident in Fig. 8, the results generated by ZONE surpass those of the other methods, demonstrating its impressive prowess in local editing.

A.2. Multi-Turn Editing Examples

We use our ZONE to edit 2 images in a multi-turn style and compare the editing results with those obtained from the other methods. Specifically, each method is employed to edit each image three times, with different instructions. As illustrated in Fig. 9, our ZONE can achieve high-quality local edits under multiple instructions and preserve the original image's non-edited regions. In contrast, the results generated by the other methods exhibit noticeable distortions from the original images after multiple rounds of editing, which is not preferred in practical applications.

A.3. Cross-Attention Map Visualization

As shown in Fig. 10, the first row demonstrates the editing results, and the second row illustrates the averaged cross-attention maps. From the cross-attention maps, we can see that by fusing the denoised latents of the two methods as described in Equation (5) of the main paper, our approach achieves a better localization capability under the "Remove" editing intent compared to the two methods.

B. Implementation Details

We conduct all our experiments based on open-source projects and models. We adopt an NVIDIA V100-SXM2-32GB GPU for the action classifier training and for ZONE testing. The action classifier $\mathcal{A}_I$ leverages the instruction embeddings extracted by the text encoder of InstructPix2Pix (IP2P) [5] as its input, and outputs the probability logits for each action. To train the action classifier, we first use GPT-3.5 to generate samples for training and testing, and then we lock the weights of the text encoder of IP2P and optimize $\mathcal{A}_I$ using Adam [27] with a learning rate of 0.1 for 30 epochs. The action classifier achieves 100% top-1 classification accuracy on the test set.

We set 20 sampling steps for the fused IP2P and average the cross-attention layers of the first three UNet upsampling blocks and the second to the fourth downsampling blocks to get the fused cross-attention maps among all the denoising steps.

The action classifier $\mathcal{A}_I$ is a simple Multi-Layer Perceptron (MLP), comprising two linear layers with an intermediate ReLU activation function. The input dimension of the first linear layer and the length of the embedding outputted from the CLIP text encoder [42] are the same (equal to 768), and the output dimension at this layer is 128. The intermediate ReLU function introduces non-linearity to the output, and the second linear layer takes the 128-dimensional output from the ReLU function and produces a 3-dimensional output to classify the given instruction.

C. Experimental Details

C.1. Baselines

To ensure consistency and convenience in method comparison, we uniformly adopt the implementation from the diffusers project [3] for IP2P [5], MagicBrush [59], DiffEdit [8], and Pix2Pix-Zero [40], and use their default parameters to generate

[3] https://github.com/huggingface/diffusers

Figure 8. Single-turn editing examples. IP2P: InstructPix2Pix [5]; T2L: Text2LIVE [3]; MB: MagicBrush [59].

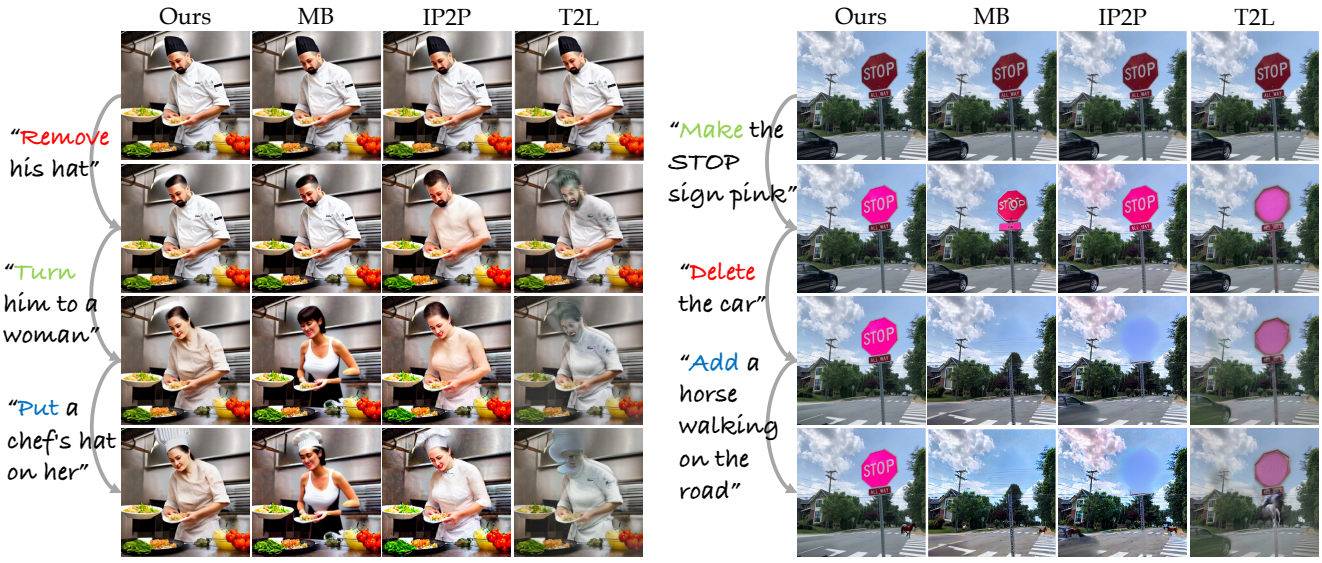

Figure 9. Multi-turn editing examples. IP2P: InstructPix2Pix [5]; T2L: Text2LIVE [3]; MB: MagicBrush [59]. Best viewed zoomed in.

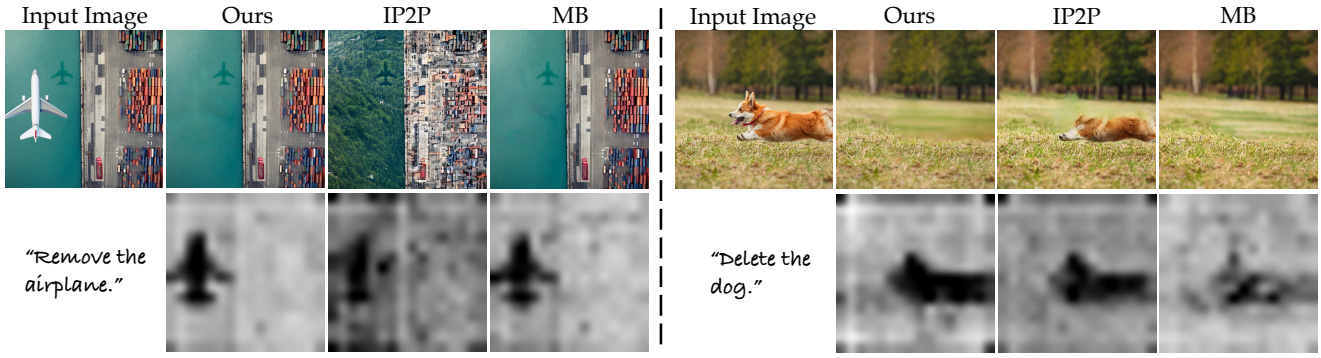

Figure 10. Cross-attention map comparisons. The **darker parts** in each cross-attention map (the second row) denote the edit regions.

results and calculate the metrics. For Text2LIVE [3], we conduct experiments using its official code repository. To eliminate the potential discrepancies in generative capabilities arising from different versions of Stable Diffusion used across these methods, we employ Stable Diffusion 1.5 [4] as the base model. Notably, since half of these methods do not support instructions as textual inputs, we design text prompts or additional assistance equivalent to instructions during our comparative experiments to achieve a relatively fair comparison.

C.2. Datasets

In this section, we provide the generation details of the dataset for action classification and the test set that we collect to evaluate the metrics for our ZONE and other editing methods.

Dataset for action classification. We employ GPT-3.5 [5] to generate the dataset used for training the action classifier. Our primary objective is to generate sentences that closely resemble user instructions, with the editing focus on common items found in real images. To achieve this, we choose categories from the COCO dataset to serve as the vocabulary for sentence generation. The following prompt is designed for generating training and testing data:

[4] https://huggingface.co/runwayml/stable-diffusion-v1-5
[5] https://chat.openai.com

"Now you are a dataset bot, who will generate a training dataset for a three-fold (change, add, and remove) sentence classification task. Specifically, you should generate a sentence along with its label. In this task, we aim to generate a dataset for "change", "add", and "remove" (labeled 0, 1, 2): For example: "turn the cat into a dog, 0", "give the dog a hat, 1", "get rid of the person on the left, 2". You should generate 450 sentence-label pairs if I give the instruction "train", and 150 pairs when I give the instruction "test". I expect your response to be straight-forward, each sentence should be within 30 words, and you should freely select the words in the following list: ['person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus', 'train', 'truck', 'boat', 'traffic light', 'fire hydrant', 'stop sign', 'parking meter', 'bench', 'bird', 'cat', 'dog', 'horse', 'sheep', 'cow', 'elephant', 'bear', 'zebra', 'giraffe', 'backpack', 'umbrella', 'handbag', 'tie', 'suitcase', 'frisbee', 'skis', 'snowboard', 'sports ball', 'kite', 'baseball bat', 'baseball glove', 'skateboard', 'surfboard', 'tennis racket', 'bottle', 'wine glass', 'cup', 'fork', 'knife', 'spoon', 'bowl', 'banana', 'apple', 'sandwich', 'orange', 'broccoli', 'carrot', 'hot dog', 'pizza', 'donut', 'cake', 'chair', 'couch', 'potted plant', 'bed', 'dining table', 'toilet', 'tv', 'laptop', 'mouse', 'remote', 'keyboard', 'cell phone', 'microwave', 'oven', 'toaster', 'sink', 'refrigerator', 'book', 'clock', 'vase', 'scissors', 'teddy bear', 'hair drier', 'toothbrush'] and make sure the sentence is short and clear, and the label is correct, and the dataset is balanced. Please just reply with the sentence-label pairs, and wait for my instructions. Note that the generated sentence-label pairs should not repeat."

The data generated by GPT-3.5 undergoes manual verification. The final training dataset includes 150 samples each for the "add", "remove", and "change" actions, while the test dataset comprises 50 samples for each action. We show some samples from the training dataset in Table 3.

Turn the bicycle into a motorcycle, 0
Make the apple a banana, 0
Swap the baseball glove for a tennis racket, 0
Replace the chair with a couch, 0
Put a frisbee next to the cat, 1
Attach a remote to the TV, 1
Include a toothbrush on the dining table, 1
Give the horse a suitcase, 1
Remove the horse, 2
Take away the umbrella, 2
Delete the traffic light, 2
Erase the microwave, 2

Table 3. Examples of the training dataset of the action classifier.

Test set for evaluation. We present the test set utilized in our evaluation in Fig. 11. Initially, we gather 60 images from the Internet and create 40 synthetic images using Stable Diffusion 1.5 [44]. Subsequently, each image is cropped to a resolution of 512×512. Then we use BLIP [32] to caption each image and manually annotate the instructions, output captions, source objects, and target objects. Three annotation examples are shown in Table 4.

Keys	**Example 1**	**Example 2**	**Example 3**
action	Change	Remove	Add
input caption	A blue car in front of a forest	A man in black with a tie	A photo of Elon Musk
output caption	A red car in front of a forest	A man in black	A photo of Elon Musk with glasses
instruction	paint the car red	get off his tie	give him glasses
source object	blue car	a tie	N/A
target object	red car	N/A	glasses

Table 4. Three annotation examples. "N/A" indicates the absence of words.

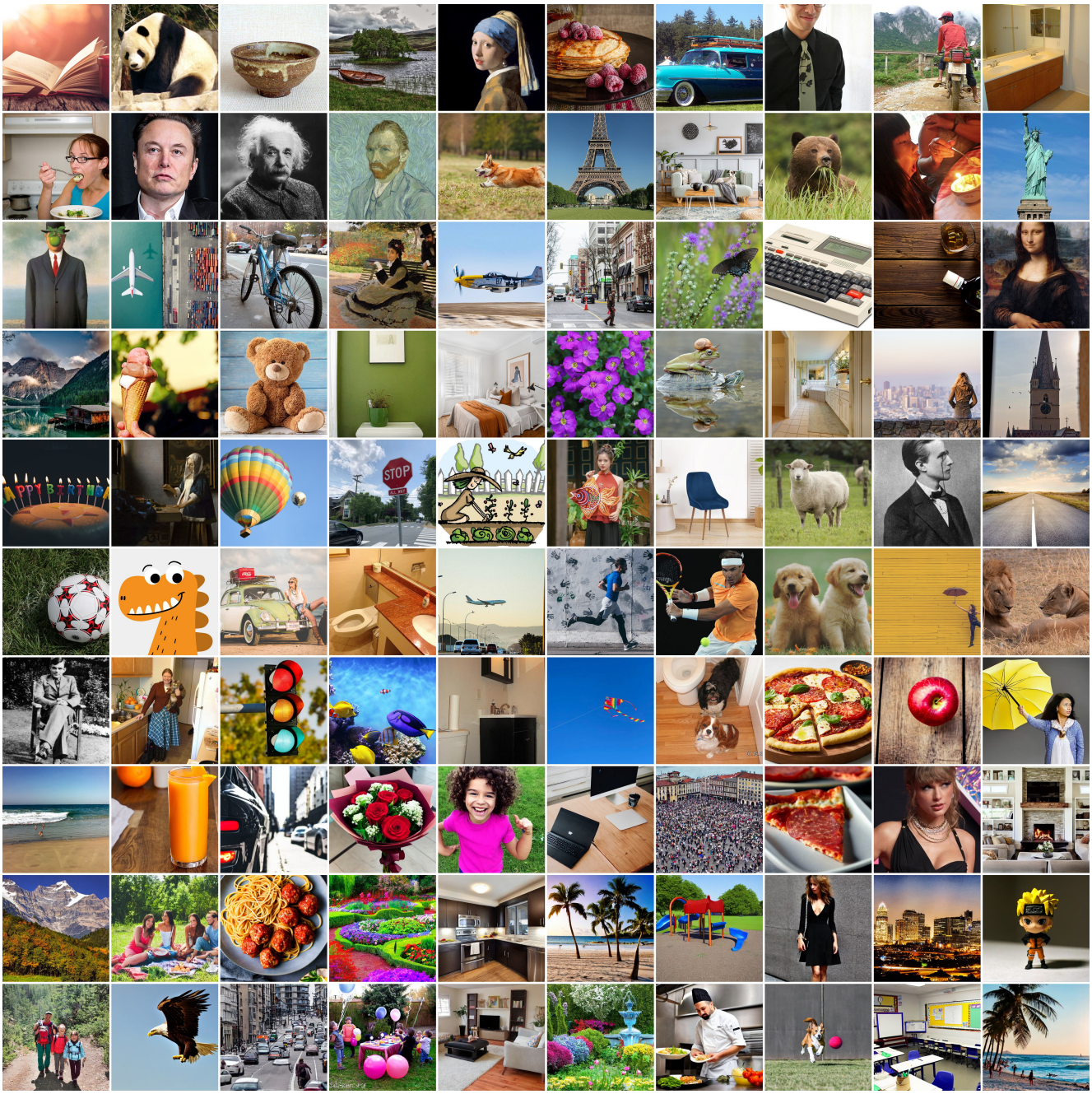

Figure 11. **Images in the test set**. We calculate the evaluation metrics and provide visualizations in the main paper with images in this set.

C.3. Evaluation Metrics

L1/L2 distance. The L1 and L2 distances serve as the metrics for evaluating structural and pixel-wise similarities between two images. The L1 distance measures absolute differences in pixel values, while the L2 distance calculates squared differences. Both metrics play a critical role in assessing dissimilarity, with smaller distances indicating greater image similarity in both pixel intensity and spatial structure.

LPIPS score. LPIPS (Learned Perceptual Image Patch Similarity) [60] is a metric designed for evaluating the perceptual similarity between two images. It takes into account both pixel-level differences and high-level visual features, providing a

comprehensive measure of how similar images appear to humans.

CLIP-based metrics. CLIP, or Contrastive Language-Image Pre-training, is a transformative model that excels in understanding the intricate relationships between text descriptions and images [42]. Through a pre-training process that employs contrastive learning, CLIP learns a shared embedding space where images and text descriptions are represented as vectors. This shared space is designed to bring semantically related content in close proximity. The model tokenizes images into regions and text into tokens, leveraging a transformer architecture with cross-modal attention to establish connections between corresponding regions and tokens. Both the CLIP-I and CLIP-T metrics evaluate the input image/text in the shared embedding space:

- CLIP image similarity (CLIP-I) is designed to evaluate the image quality in both semantics and structure. This metric is computed by calculating the cosine similarity of the embedding vectors of the source image and the target image.
- CLIP text-image similarity (CLIP-T) is used to evaluate the alignment between the edited image and its corresponding caption. More specifically, CLIP-T calculates the cosine similarity between the embedding vectors of the edited image and its corresponding caption.

More evaluation details. We employ 512×512 images as inputs for each method during evaluations. However, DiffEdit [8] requires image inputs with a resolution of 768×768 to function properly. So we first resize the test images to 768×768 for DiffEdit to ensure its proper performance and resize the outputs back to 512×512 to calculate the metrics.

C.4. Human Evaluation

Success rate. We invite five volunteers to annotate the success rates of the six methods on the test set. To simplify the annotation process and avoid bias, we design a tool that can display the editing results of each method in a randomly shuffled order and anonymous style (see Fig. 12). The volunteers are then asked to decide whether to accept or reject the edited image based on the editing quality (*i.e.,* preservation of the non-edited regions and the realism of the edited image) and text-image alignment between the output caption and the edited image. Ultimately, the success rate of each method is obtained by dividing the number of accepted results by the total number. To minimize annotation bias, we calculate the mean and standard deviation of the success rates from the five volunteers and demonstrate the results in Table 2 of the main paper.

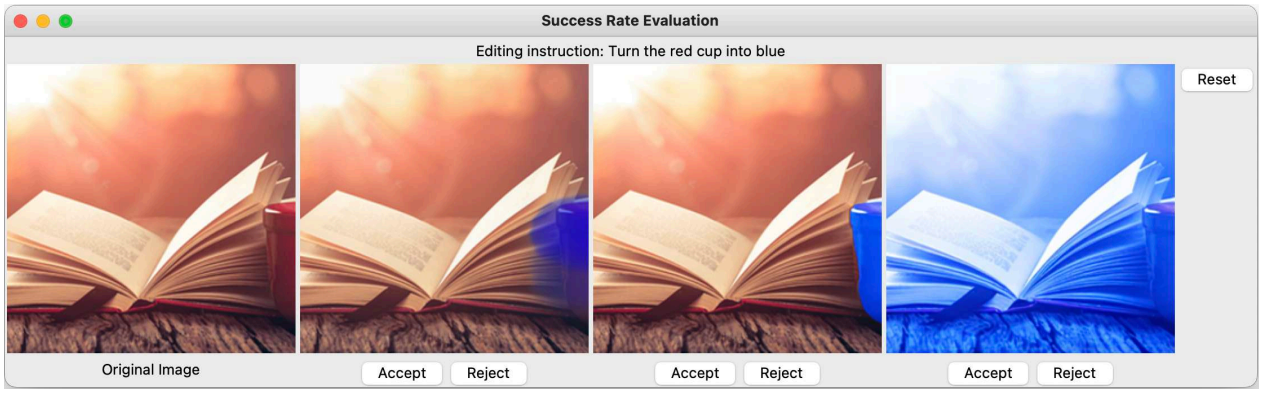

Figure 12. A screenshot of the annotation tool. The original image, the instruction, and three results randomly selected from the six methods are displayed each time.

User preference rate. We conduct a user study, which includes 16 sets of randomly selected editing results. Each set contains six results obtained by the six methods that we compare in the experiment, presented in a randomly shuffled order. The users are asked to give a preference score according to the degree of agreement between each editing result and the corresponding instruction, as well as the similarity to the original image, with the score from 1 to 10 and a higher score indicating a higher preference. A total of 30 users participate in this test. The final results are calculated by dividing the total score obtained for each method S_i by the total score obtained for all methods:

$$UPR(i) = 100 \times S_i / \sum_{i=1}^{6} S_i. \tag{9}$$

D. Limitations

While our method can produce impressive local manipulations of images and address the over-edit issue of InstructPix2Pix, it still has limitations. First, its editing capabilities are constrained by the instruction-guided diffusion models we employ, which may lead to occasional ineffectiveness in editing. This issue can be addressed in the future with more powerful instruction-guided diffusion models. Secondly, our method falls short of localization in complex scenes (*e.g.,* multiple similar objects or tiny objects), which is a challenging task that still needs to be explored. Lastly, the current set of editing actions is relatively limited, more actions like "move", "resize", or "copy" will be considered in future work.

E. Social Impact

Our work introduces a novel method for image local editing, which edits a specific region in the original image with an intuitive instruction. This method allows for precise local editing without affecting other areas of the image, resulting in a realistic final composite image. Malicious groups may exploit this advantage to spread false information or cause misunderstanding. However, we believe that the harm caused by such improper usage can be mitigated with AI-generated content watermarking algorithms or supervising regulations.