Image Editing As Programs with Diffusion Models

Yujia Hu[1], **Songhua Liu**[2,1], **Zhenxiong Tan**[1], **Xingyi Yang**[3,1], **and Xinchao Wang**[1]*
[1]National University of Singapore
[2]School of Artificial Intelligence, Shanghai Jiao Tong University
[3]The Hong Kong Polytechnic University
yujia.hu@u.nus.edu, xinchao@nus.edu.sg

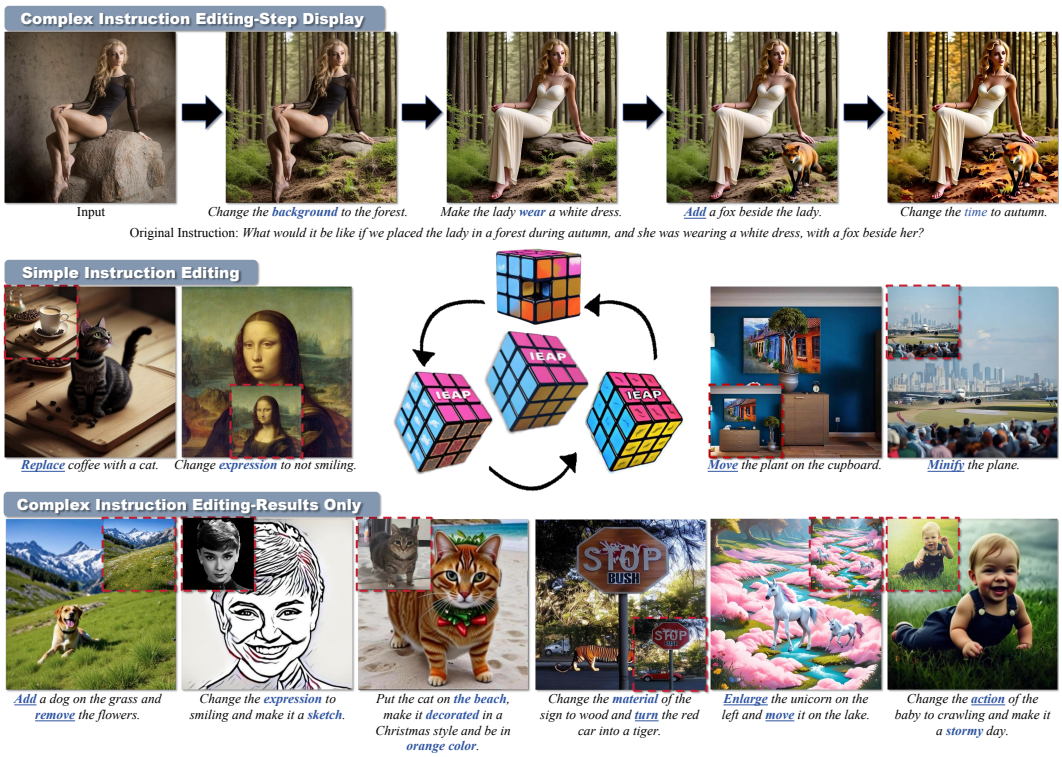

Figure 1: Visual results of our IEAP. Rows 1 and 3 showcase complex multi-step edits (Row 1 is further decomposed into individual instructions), while Row 2 shows single-instruction edits. Single instructions are underlined if needing to be reduced to atomic operations.

Abstract

While diffusion models have achieved remarkable success in text-to-image generation, they encounter significant challenges with instruction-driven image editing. Our research highlights a key challenge: these models particularly struggle with structurally-inconsistent edits that involve substantial layout changes. To address this gap, we introduce ***I**mage **E**diting **A**s **P**rograms* (IEAP), a unified image editing framework built upon the Diffusion Transformer (DiT) architecture. Specifically, IEAP deals with complex instructions by decomposing them into a sequence of programmable *atomic* operations. Each atomic operation manages a specific type of structurally consistent edit; when sequentially combined, IEAP enables the execution of arbitrary and structurally-inconsistent transformations. This re-

*Corresponding Author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

ductionist approach enables IEAP to robustly handle a wide spectrum of edits, encompassing both structurally-consistent and inconsistent changes. Extensive experiments demonstrate that IEAP significantly outperforms state-of-the-art methods on standard benchmarks across various editing scenarios. In these evaluations, our framework delivers superior accuracy and semantic fidelity, particularly for complex, multi-step instructions. Codes are available here.

1 Introduction

Image editing lies at the heart of a wide range of applications from photo retouching and content creation to visual storytelling and scientific visualization [42, 5, 63]. With the advent of diffusion models [23, 53, 47], the field has shifted towards highly precise and controllable manipulations [45, 12, 62]. The inherently progressive denoising process enables multi-stage pipelines [24, 4, 2] and localized editing methods [10, 77, 58], and its native support for multi-modal inputs has inspired unified frameworks that integrate heterogeneous signals within a single model [33, 15, 71, 18].

More recently, text-to-image pipelines based on Diffusion Transformers (DiTs) [46, 13, 31] have set new standards in generative fidelity. However, their capacity for instruction-driven editing [41, 27] remains under-explored. Notably, although there are a few existing methods [80, 37] that have extended DiTs to instruction-driven editing, they are always restricted to a narrow set of common editing operations and lack evaluation on comprehensive editing tasks.

To address this limitation, we initiate a taxonomy study of image editing instructions to systematically assess the editing capabilities of current DiT-based conditional generation methods. Our empirical analysis reveals an interesting performance dichotomy: While current methods demonstrate proficiency in structurally-consistent edits where the layouts of the input and output images remain aligned, they exhibit significant degradation when handling structurally-inconsistent operations that require layout modifications.

To overcome this issue, we introduce ***I**mage **E**diting **A**s **P**rograms* (IEAP), a unified framework atop the DiT architecture which is capable of handling diverse types of editing operations efficiently and robustly in this paper. Notably, we show that structurally-inconsistent instructions can in fact be reduced to a small set of simple operations, which are called as *atomic* operations in our paper. Thus, instead of treating each edit as a monolithic, end-to-end task, IEAP levarages the Chain-of-Thought (CoT) reasoning [65] to break the original editing command into a sequence of atomic operations, which are namely Region of Interest (RoI) localization, RoI inpainting, RoI editing, RoI compositing and global transformation, and then executes them in a sequential manner via a neural program interpreter [49].

The five atomic operations serve as the fundamental building blocks for complex editing tasks. As such, through the sequential combination of atomic operations, IEAP can robustly handle complex, multi-step instructions that are typically confound in conventional end-to-end approaches.

Extensive experiments show that our framework demonstrates state-of-the-art performance across standard benchmarks, excelling in both structural preservation and alteration tasks through atomic-level operation decomposition compared to other approaches. Simultaneously, the CoT reasoning and programming pipeline of IEAP enable significantly more accurate and semantically more coherent edits under complex, multi-step instructions even compared to the leading proprietary models.

Our main contributions can be summarized as follows:

- We present a comprehensive taxonomy and empirical analysis of instruction-driven editing in DiT-based conditional generation, revealing a performance dichotomy between structurally-consistent and -inconsistent edits.
- We introduce ***I**mage **E**diting **A**s **P**rograms* (IEAP), a unified framework on the DiT backbone that leverages CoT reasoning to parse free-form instructions into sequential atomic operations and then executes them sequentially by a neural program interpreter, thereby enabling robust handling of layout-altering and complex edits.
- Extensive experiments demonstrate that IEAP achieves state-of-the-art performance in both structure-preserving and -altering scenarios, delivering notably higher accuracy and semantic fidelity especially on complex, multi-step instructions compared to existing methods.

2 Related Work

Conditional image generation. Early conditional image generation approaches like ControlNet [77] typically adopt plug-in control adapters to incorporate single condition [3, 16, 35] like segmentation mask or diverse conditional inputs [82, 48, 26, 40, 70] to guide the generation of images. Recently, the field of conditional image generation has witnessed remarkable breakthroughs through the integration of DiTs [13, 46, 31], with continuous innovations improving output quality and edit precision [45]. Some methods [69, 32, 68, 9] aim to create a unified DiT foundation for versatile conditional image generation and editing by integrating diverse inputs within a single framework. while approaches like OminiControl [60] and so on [61, 79, 38, 80, 67] leverage LoRA-based fine-tuning [25] for lightweight and effective control.

Instructional image editing. Instruction-based image editing [41, 27] enables intuitive, language-driven modifications of existing images. Early works like InstructPix2Pix [6] establishes paired instruction–image datasets for supervised fine-tuning of diffusion models. For subsequent works, some of them focus on architectural refinement [38, 37, 81, 34, 20], which introduce specialized conditioning units and multi-stage training to improve control granularity and consistency, others concentrate on data-centric enhancements [76, 17, 56, 8], that expand instruction coverage and diversify edit examples. Moreover, some approaches [75, 28, 33, 15] has unified LLM-based [1] language reasoning with diffusion-based synthesis in a single framework, and some [72, 78] leverage CoT [65] and in-context learning [21] to enhance the reasoning ability of models for more complex editing tasks. Meanwhile, certain approaches [64, 66, 54] build upon multimodal understanding to decompose intricate instructions, ensuring editing precision and output reliability. More recently, some works [14, 80, 37] have advanced image editing with DiTs. For instance, ICEdit [80] leverages the in-context generation capabilities of large-scale DiTs to achieve flexible few-shot instruction editing, while Step1X-Edit [37] focuses on large-scale data construction and multi-modal integration to enable general-purpose image editing with performance approaching proprietary models.

3 Motivation

3.1 Preliminaries

Diffusion Transformer Fundamentals. The image generation process of text-guided DiTs [46, 13, 31] is accomplished by successively denoising input tokens in multiple steps. At step t, the model processes:

$$\mathbf{S}_t = [\mathbf{X}_t, \mathbf{C}_T] \tag{1}$$

where $\mathbf{X}_t \in \mathbb{R}^{N \times d}$ represents noisy image tokens and $\mathbf{C}_T \in \mathbb{R}^{M \times d}$ denotes text tokens, they share the embedding dimension d. Image tokens use Rotary Position Embedding (RoPE) [59] with spatial coordinates (i, j), while text tokens fix positions at $(0, 0)$, enabling Multi-Modal Attention (MMA) [44] mechanisms to model cross-modal interactions.

Unified Conditioning Framework. To integrate visual control signals, the prior work [60] extends the baseline formulation by incorporating encoded condition images:

$$\mathbf{S}_t = [\mathbf{X}_t, \mathbf{C}_T, \mathbf{C}_I] \tag{2}$$

where $\mathbf{C}_I \in \mathbb{R}^{N \times d}$ denotes latent tokens from condition images via the pretrained VAE encoder [30, 52]. This unified sequence enables tri-modal fusion within transformer architectures, eliminating spatial misalignment inherent in feature concatenation baselines.

Moreover, an auxiliary adaptive positional encoding mechanism further preserves spatial consistency across these modalities by assigning coordinates to each token type with minimal overhead.

Gap in Instruction-Driven DiT Editing. Despite the rapid advances in DiT-based conditional image generation [60, 79, 38, 67], research on instruction-driven editing [41, 27] remains scarce. The few existing methods [80, 37] that do support instructional edits are typically confined to a small set of routine operations, and lack a comprehensive evaluation across diverse editing scenarios, leaving DiT's true editing potential unclear. This gap motivates us to conduct a taxonomy study of DiT's ability in instructional image editing, which is detailed in Sec. 3.2.

Figure 2: Results of our preliminary experiments. Figure (a) shows the GPT-4o scores for three editing types across instruction faithfulness and semantic consistency, ranging from 1 to 5. Figure (b) shows the representative failure cases from local semantic editing.

3.2 Preliminary Experiments and Observations

To this end, we conduct a comprehensive evaluation of diffusion models for instruction-driven editing, uncovering an interesting performance dichotomy: *While these methods excel at structurally-consistent edits, they falter dramatically on structurally-inconsistent operations that demand explicit layout modifications.*

Taxonomy and Experimental Setup. To enable systematic analysis [27, 73, 72], we first categorize instruction-based image editing into three main types: local semantic editing, which modifies the identity, position or size, e.g., add, remove, replace, action change, move and resize; local attribute editing, which adjusts certain properties of objects, e.g., color change, texture change, appearance change, expression change, and background change; and overall content editing, which alters the whole image consistently, e.g., tone transfer and style change.

Then we use AnyEdit dataset [73] and OminiControl [60] to train models on the above editing types, accompanied by GPT-4o [29] to rate each edit on instruction faithfulness and semantic consistency.

Results and Analysis. As shown in Fig. 2(a), both local attribute editing and overall content editing attain relatively high GPT-4o scores, whereas local semantic editing exhibits a notable performance drop. As illustrated in Fig. 2(b), the cases of "add" and "action change" alter unrelated areas like the background, and the remaining four cases demonstrate a complete failure.

We attribute this discrepancy to the fact that, unlike local attribute and overall content edits, local semantic edits require explicit spatial-layout modifications. For instance, "add" and "delete" operations necessitate instance-level scene recomposition, while "move" and "resize" further demand precise coordinate system recalibration.

Key Insight. Based on the above analysis, spatial-layout modification remains a critical challenge for diffusion-based editing models; conversely, edits that preserve the original layout demonstrate substantially better performance. We speculate that, with limited training data, it is difficult for the model to learn the complex patterns underlying layout-changing tasks. Although DiT architectures [46, 13, 31] employ powerful full-attention mechanisms to capture long-range dependencies, they still struggle with editing operations that require nontrivial scene reconfiguration.

Due to the combinatorial complexity of spatial-layout modifications and the empirical limitations of DiT architectures, we propose to simplify the layout-editing paradigm through decomposition, which is detailed in Sec. 4.

4 Methods

4.1 Program with Atomic Operations

The insight in Sec. 3.2 motivates us to decouple semantic and spatial reasoning. Building on this foundation, we propose a programmatic reduction framework that systematically decomposes complex editing instructions into modular atomic operations. Specifically, we first formulate instruction-driven image editing as an executable program via Chain-of-Thought (CoT) reasoning [65], and then use a neural program interpreter [49] to transcode the reasoning graph into a dynamic execution plan, sequentially invoking relevant atomic modules.

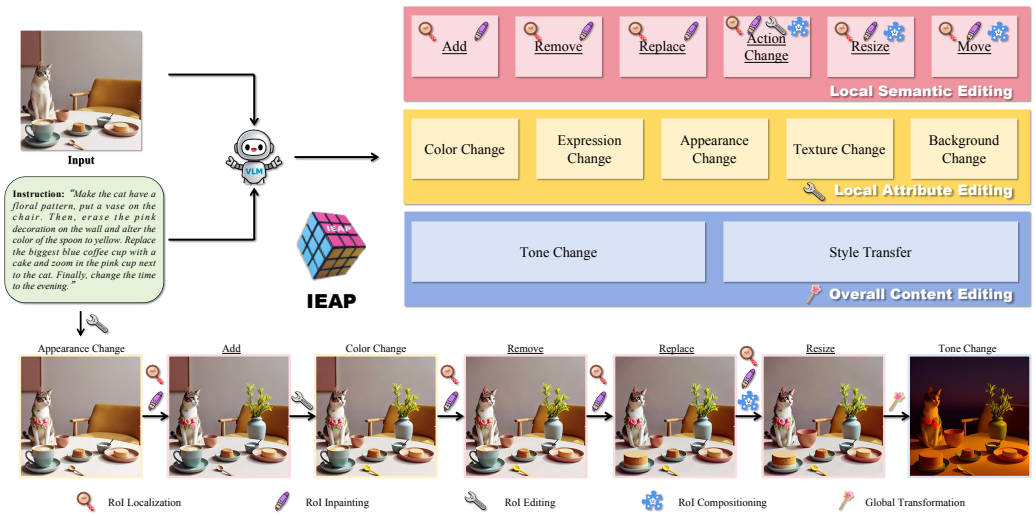

Figure 3: Our pipeline. The original instruction is first parsed by a VLM into atomic operations, which are then sequentially executed via a neural program interpreter.

4.2 General Pipeline

We abstract all editing instructions into five atomic primitives: (1) RoI Localization: Identify and isolate the relevant region in the image that the instruction refers to, serving as the spatial grounding step for subsequent localized edits; (2) RoI Inpainting: Introduce new visual content or remove existing elements within the localized region, enabling semantic-level additions, substitutions, or deletions; (3) RoI Editing: Modify visual attributes within the region, such as color, texture, or appearance, to reflect fine-grained property changes specified by the instruction; (4) RoI Compositing: Reintegrate the edited region into the full image while preserving spatial coherence and visual continuity; (5) Global Transformation: Adjust the overall content for coherent full-image modifications, such as changing the illumination, weather, or style of the whole image.

The overall pipeline is shown as Fig. 3. We reduce any editing instruction into an arbitrary combination of the five atomic operations described above, which can be formulated as:

$$T \equiv \bigoplus_{k=1}^{K} \mathcal{A}_k, \qquad \mathcal{A}_k \in \{\mathcal{A}_{\text{loc}}, \mathcal{A}_{\text{inp}}, \mathcal{A}_{\text{edit}}, \mathcal{A}_{\text{comp}}, \mathcal{A}_{\text{global}}\} \tag{3}$$

where T denotes the free-form editing instruction, $\bigoplus$ represents the sequential program combination, K is the number of atomic operations, $\mathcal{A}_{\text{loc}}$, $\mathcal{A}_{\text{inp}}$, $\mathcal{A}_{\text{edit}}$, $\mathcal{A}_{\text{comp}}$, and $\mathcal{A}_{\text{global}}$ represent the five atomic primitives respectively.

RoI Localization. All problematic local semantic edits share a common first step: localizing a Region of Interest (RoI) in the image for editing. Given an image I and an editing instruction T, we first employ a Large Language Model (LLM) [1] to locate the text RoI:

$$\rho = M_{\text{LLM}}(T), \tag{4}$$

where ρ represents the text RoI extracted by the LLM M_{LLM}. Subsequently, we achieve accurate localization of image RoI by:

$$R = M_{\text{seg}}(I, \rho), \tag{5}$$

where R denotes the image RoI segmented by the segmentation model M_{seg} [74].

For add operation, the instruction may not specify a text RoI, or the specification may be ambiguous. In such cases, we first derive the overall layout of all candidate objects using the capability of segmentation models [50, 74], and then prompt the LLM to determine the appropriate image RoI based on T.

Regarding move and resize, once the image RoI is obtained, we update the spatial layout of the image using an LLM [1]. Specifically, we provide the LLM with a set of in-context examples that define our layout representation and demonstrate representative editing patterns [36]. Given the current layout L and the instruction T, the LLM is prompted to produce a modified layout L_{edit}, as formulated below:

$$\text{Tags} = M_{\text{LLM}}(I), \quad L = M_{\text{seg}}(\text{Tags}), \quad L_{\text{edit}} = M_{\text{LLM}}(L, T). \tag{6}$$

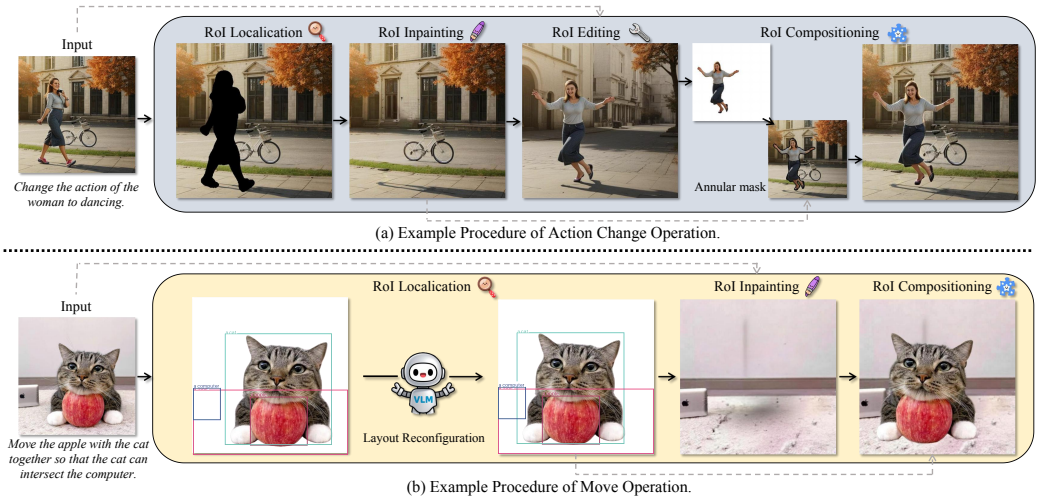

Figure 4: Example procedure. Figure (a) and Figure (b) illustrate the procedures of action change and movement respectively.

We then derive the geometric differences between L and L_{edit} and convert them into the corresponding affine transformations, consisting of translation, scaling, and reshaping, and apply it to R to update the spatial configuration, yielding the transformed mask R'.

RoI Inpainting. Once the image RoI has been localized, we apply inpainting to seamlessly fill and complete the region. For additive and substitutive operations, which aim to introduce new objects, we employ a prompt-conditioned inpainting process to guide the generation of new content. Specifically, we first extract the semantic entity E from the instruction T via an LLM [1]:

$$E = M_{\text{LLM}}(T), \tag{7}$$

and then construct a composite prompt P in the form: *"add E on the black region"*. For removal operations, which aim to eliminate existing content without introducing new semantics, we adopt a background-oriented infilling strategy, setting P as *"fill in the hole of the image"*. The edited image I_{edit} is then generated by:

$$I_{\text{edit}} = M_{\text{inpaint}}\left(I \odot (1 - R), P\right), \tag{8}$$

where M_{inpaint} denotes the inpainting model trained by us.

RoI Editing. When operations pertain to property change are performed, we use the trained attribute editing model M_{attr} to perform edits in this stage to obtain I_{edit}:

$$I_{\text{edit}} = M_{\text{attr}}\left(I, T\right). \tag{9}$$

RoI Compositing. To ensure seamless integration of the edited RoI with its surrounding context, we first construct an annular mask M_{ann} by applying morphological dilation and erosion [51, 55] to the transformed RoI mask R':

$$M_{\text{ann}} = \text{Dilate}(R', k_1) \setminus \text{Erode}(R', k_2). \tag{10}$$

Then, we employ a fusion network M_{fusion}, trained on ring-masked object boundaries, to refine the pre-composited image I_{prep} using the generated annular mask. The final edited image is obtained as:

$$I_{\text{edit}} = M_{\text{fusion}}\left(I_{\text{prep}} \odot (1 - M_{\text{ann}}), P\right), \tag{11}$$

where P is set as "inpaint the black-bordered region so that the object's edges blend smoothly with the background" to guide seamless boundary blending.

Global Transformation. Like RoI editing, in the scenarios involving global transformation, we use the trained global transformation model M_{global} to perform edits in this final stage to obtain I_{edit}.

Figure 5: Comparison results of ours with baseline methods on representative editing cases. Others exhibit poor performance even on some common editing operations, while our approach demonstrates superior effectiveness across all operations.

5 Experiments

5.1 Experimental Settings

Training Settings. We train four specialized models for RoI inpainting, RoI editing, RoI compositing, and global transformation respectively. All models are fine-tuned on FLUX.1-dev [31] using LoRA [25], with default settings for rank 128 and alpha 128. Training is conducted with a batch size of 1 and runs for 50,000 iterations each. We use the Prodigy optimizer [39], enabling safeguard warmup and bias correction, with a weight decay of 0.01. The experiments are conducted on single NVIDIA H100 GPU (80GB).

Method	MagicBrush test				AnyEdit test			
	CLIP$_{im}$ ↑	CLIP$_{out}$ ↑	L1 ↓	DINO ↑	CLIP$_{im}$ ↑	L1 ↓	DINO ↑	GPT ↑
InstructPix2Pix	0.838	0.229	0.112	0.758	0.801	0.110	0.765	3.83
MagicBrush	0.886	0.241	0.074	0.859	0.824	0.128	0.742	3.90
UltraEdit	0.911	0.227	0.061	0.883	0.833	0.114	0.772	3.93
GenArtist	0.908	0.232	0.063	0.889	0.829	0.121	0.776	3.98
OmniGen2	0.881	0.242	0.100	0.830	0.857	0.132	0.772	4.13
ICEdit	0.913	0.236	**0.058**	0.885	0.847	0.110	0.765	4.13
Ours	**0.922**	**0.247**	0.060	**0.897**	**0.882**	**0.096**	**0.825**	**4.41**

Table 1: Quantitative results on MagicBrush and AnyEdit test set.

Method	Local Semantic Editing				Local Attribute Editing				Overall Content Editing			
	CLIP$_{im}$ ↑	L1 ↓	DINO ↑	GPT ↑	CLIP$_{im}$ ↑	L1 ↓	DINO ↑	GPT ↑	CLIP$_{im}$ ↑	L1 ↓	DINO ↑	GPT ↑
InstructP2P	0.826	0.132	0.738	3.74	0.790	0.135	0.737	3.92	0.766	0.156	0.642	3.91
MagicBrush	0.860	0.106	0.796	3.90	0.809	0.117	0.762	4.21	0.763	0.187	0.616	3.99
UltraEdit	0.867	0.095	0.812	3.86	0.801	0.092	0.793	3.94	0.754	0.201	0.611	4.41
GenArtist	0.864	0.097	0.821	3.88	0.814	0.108	0.801	3.96	0.752	0.207	0.595	4.38
OmniGen2	0.893	0.097	0.834	4.12	0.832	0.114	0.786	4.06	0.783	0.224	0.634	4.42
ICEdit	0.881	0.088	0.810	4.08	0.825	0.095	0.795	4.06	0.759	0.188	0.603	4.45
Ours	**0.907**	**0.081**	**0.854**	**4.42**	**0.861**	**0.083**	**0.821**	**4.54**	**0.895**	**0.107**	**0.879**	**4.51**

Table 2: Quantitative results on different types of editing operations.

Dataset Setup. For both the RoI editing and global transformation models, we sample from the relevant subsets of the AnyEdit [73] dataset and apply GPT-4o [29] to filter the data of some types that have numerous noisy examples. To cover facial expression edits absent in AnyEdit, we integrate the CelebHQ-FM dataset [11], which offers consistent identities and annotated expressions suitable for our instruction schema. The RoI inpainting and RoI compositing models are trained on samples from the "add", "remove" and "replace" splits of AnyEdit. For each sample, we first obtain the image RoI according to the editing instruction. In the RoI Inpainting training setup, we set the pixels within image RoI to black as input to train. For RoI Compositing, we set k_1 and k_2 as 3 in default to blackout the annular mask region of image RoI as input for training.

Evaluation Settings. We evaluate our method on two benchmarks: MagicBrush test set [76], a widely used dataset spanning diverse editing types, and AnyEdit test set [73], from which we select 16 instruction-based editing categories. For MagicBrush, we follow previous works [76, 81, 15, 56] and report CLIPimg, CLIPout [22], L_1, and DINO [7, 43] scores to measure the similarity between the generated results and ground-truth images. While for AnyEdit, where some categories lack reference captions required for calculating CLIPout, we instead leverage GPT-4o [29] to rate each edited image on a scale from 1 to 5 across three dimensions: instruction faithfulness, semantic consistency, and aesthetic quality, with the final GPT score obtained by averaging the three aspect scores.

We first compare our method with existing state-of-the-art open-source baselines, including InstructPix2Pix [6], MagicBrush [76], UltraEdit [81], GenArtist [64], OmniGen2 [66] and ICEdit [80]. In addition, to demonstrate the competitiveness of our approach against powerful proprietary multimodal foundation models in complex image editing scenarios, we further make comparisons with SeedEdit (Doubao) [57], Gemini 2.0 Flash [19], and GPT-4o [29].

5.2 Comparisons with State of the Art.

Qualitative Comparisons. Fig. 5 shows the results of our approach against other six methods [6, 76, 81, 64, 66, 80] on some representative editing cases. Unlike previous methods, which sometimes misinterpret or fail to execute the given instructions, modify unintended regions, introduce undesired artifacts, or produce visually implausible results, our method consistently exhibits clear and consistent advantages in accurately following the instruction, maintaining structural coherence, preserving instance-level fidelity and retaining fine-grained visual details.

Quantitative Comparisons. Table 1 exhibits the quantitative comparison results of our method and other approaches [6, 76, 81, 64, 66, 80] on MagicBrush test set [76] and AnyEdit test set [73]. The results show that our method demonstrates state-of-the-art performance on both datasets. On

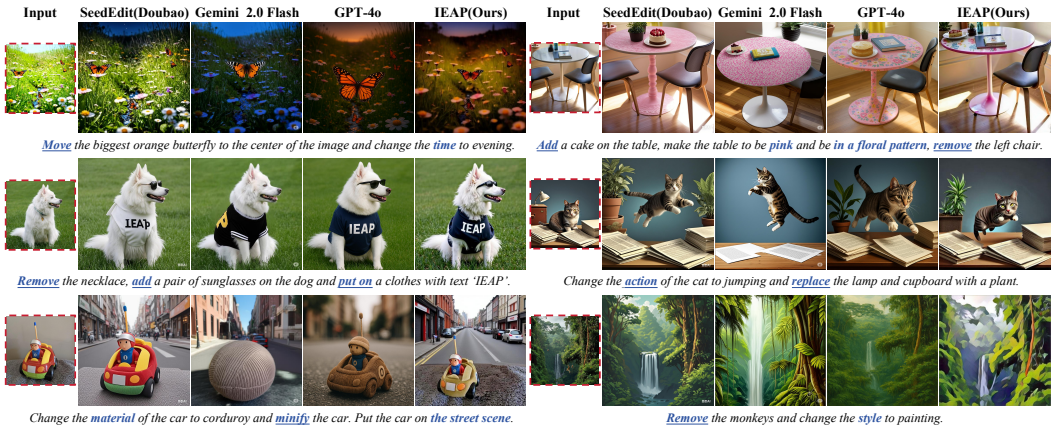

Figure 6: Comparisons on Complex Instructions with Leading Multimodal Models. Our method achieves comparable or even better edit completeness and pre-post consistency.

MagicBrush, our method achieves the best performance in terms of caption alignment, semantic consistency, and preservation of fine-grained structural details. Although it incurs a marginal increase in pixel-level deviation compared to the best [80], this is far outweighed by the substantial gains in perceptual quality and semantic fidelity. Furthermore, on AnyEdit, our approach yields significant and comprehensive improvements across all evaluation metrics, further highlighting its superiority over existing techniques.

To provide a more fine-grained analysis of editing performance, we group a subset of the instruction-based categories from the AnyEdit test set [73] into three macro-tasks: local semantic editing, local attribute editing and overall semantic editing. For local attribute editing, we augment with some CelebHQ-FM [11] test images to evaluate facial expression changes. The quantitave comparison results are shown in Tab. 2, where our method consistently outperforms other candidates across all three task categories and evaluation metrics.

Comparisons with Cutting-Edge Multimodal Models. To demonstrate the superiority of our reduction strategy on complex editing tasks, we also conduct comparative experiments against prominent closed-source multimodal models [57, 19, 29]. As illustrated in Fig. 6, our method rivals, and in most cases surpasses the performance of these leading models on intricate scenarios requiring multiple sequential edits. Unlike competing approaches, which frequently omit specified instructions or introduce extraneous alterations unrelated to the editing directives, our framework faithfully executes each instruction while ensuring superior image consistency and instance preservation.

5.3 Ablation Studies

Settings	CLIP_{im} ↑	CLIP_{out} ↑	L1 ↓	DINO ↑	GPT ↑
w/o CoT & Reduction	0.873	0.241	0.117	0.795	4.10
w/o RoI Inpainting	0.861	0.218	0.124	0.775	3.65
w/o RoI Editing	0.900	0.244	0.088	0.843	4.23
w/o Layout Reconfiguration	0.900	0.245	0.088	0.848	4.31
w/o Annular Mask Integration	0.906	**0.252**	0.083	**0.854**	4.39
Full	**0.907**	**0.252**	**0.081**	**0.854**	**4.42**

Table 3: Module-wise ablation results on AnyEdit local semantic editing test set.

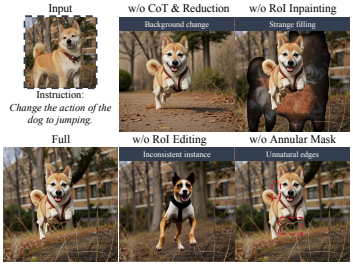

Figure 7: Qualitative ablation of action change operation.

Module-wise Ablation Studies. To quantify the impact of each key component in our framework, we perform a series of ablation studies on the AnyEdit [73] local semantic editing test set as we split in Sec. 5.2. As shown in Tab. 3, we first substitute our CoT reasoning and reduction pipeline with end-to-end editing pipeline, resulting in a marked performance deterioration across all metrics. Next, we replace our specialized RoI inpainting and RoI editing models respectively with the generic inpainting model from [60], which induces performance declines of varying degrees. We then remove the LLM-guided layout reconfiguration and instead employing random layout modifications for relevant operations, which incurs a noticeable performance decline. Finally, omitting the annular

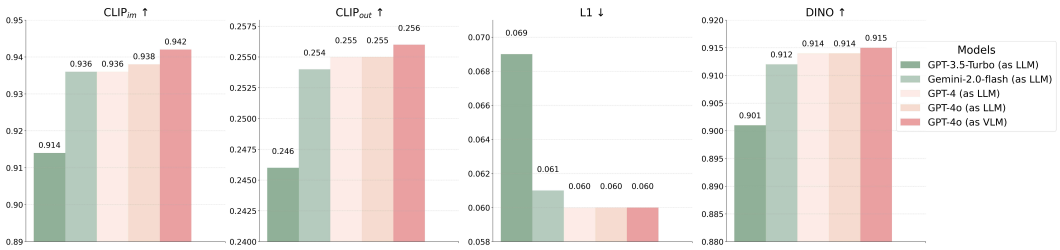

Figure 8: Ablation results of LLMs/VLMs on layout modification tasks.

mask integration produces a modest drop, underscoring its role in precise boundary delineation. Fig. 7 exhibits the ablation results on an example of "action change", visually showcasing each module's necessity. Collectively, these ablation results confirm that each component in our pipeline contributes significantly in handling robust local semantic editing tasks requiring layout changes.

Ablation Studies on Different LLMs/VLMs. We present comparative experimental results involving various LLMs and VLMs to analyze their influence on editing quality. For complex instruction decomposition, we evaluate challenging instructions from the MagicBrush dataset [76]. As shown in Table 4, the step remains robust across different VLMs, but performance drops notably when the original image is excluded.

Model Type	Model	Accuracy (%)
VLM	GPT-4o	100.0
VLM	Gemini-2.0-flash	96.7
LLM	GPT-4	90.0
LLM	GPT-3.5-turbo	76.7

Table 4: Ablation results on complex instruction decomposition.

As in IEAP, LLMs also assist in layout modifications for "add", "move" and "resize" operations, we also compare the quantitative editing performance using different models. As shown in Fig. 8, the layout modification capacity is robust across various LLMs/VLMs generally.

5.4 Applications

(a) Precise editing. (b) Multi-object manipulation. (c) Generalizability to open-world edits. (d) Image inpainting.

Figure 9: Applications of IEAP: Handling precise, multi-object, open-world, and inpainting tasks.

Our method also demonstrates strong versatility across diverse image editing scenarios. As shown in Fig. 9(a), our method can accurately localize and edit complex target entities described in the instruction. In Fig. 9(b), when multiple synonymous objects are mentioned, all relevant instances are consistently modified. Moreover, as illustrated in Fig. 9(c), our model generalizes well to some unseen and challenging tasks such as swapping object locations, which remain difficult even for advanced large models. Finally, as shown in Fig. 9(d), our approach can naturally extend to image inpainting with satisfactory performance.

6 Conclusions, Limitations and Future Work

In this paper, we propose Image Editing As Programs (IEAP), a unified DiT-based framework for instruction-driven image editing. By defining five atomic operations and using CoT reasoning to convert instructions into sequential programs, IEAP processes the ability to handle both simple and complex edits. Experiments demonstrate that IEAP outperforms state-of-the-art methods in both structure-preserving and structure-altering tasks, especially for complex edits.

Despite its strong overall performance, there are also some limitations. First, for complex shadow changes, our method sometimes leaves shadows inconsistent after compositing operations. Second, multiple editing iterations may induce progressive image quality decay. Future work could focus on addressing these issues via physics-aware shadow modeling and diffusion-based quality restoration.

Acknowledgment

This project is supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 2 (Award Number: MOE-T2EP20122-0006).

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, et al. Gpt-4 technical report, 2024.

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

[3] Omri Avrahami, Thomas Hayes, Oran Gafni, Sonal Gupta, Yaniv Taigman, Devi Parikh, Dani Lischinski, Ohad Fried, and Xi Yin. Spatext: Spatio-textual representation for controllable image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18370–18380, 2023.

[4] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18208–18218, 2022.

[5] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009.

[6] 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.

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

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

[9] Xi Chen, Zhifei Zhang, He Zhang, Yuqian Zhou, Soo Ye Kim, Qing Liu, Yijun Li, Jianming Zhang, Nanxuan Zhao, Yilin Wang, et al. Unireal: Universal image generation and editing via learning real-world dynamics. *arXiv preprint arXiv:2412.07774*, 2024.

[10] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022.

[11] Brian DeCann and Kirill Trapeznikov. Comprehensive dataset of face manipulations for development and evaluation of forensic tools, 2022.

[12] Dave Epstein, Allan Jabri, Ben Poole, Alexei Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. *Advances in Neural Information Processing Systems*, 36:16222–16239, 2023.

[13] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[14] Kunyu Feng, Yue Ma, Bingyuan Wang, Chenyang Qi, Haozhe Chen, Qifeng Chen, and Zeyu Wang. Dit4edit: Diffusion transformer for image editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 2969–2977, 2025.

[15] 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.

[16] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *European Conference on Computer Vision*, pages 89–106. Springer, 2022.

[17] 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.

[18] Vidit Goel, Elia Peruzzo, Yifan Jiang, Dejia Xu, Xingqian Xu, Nicu Sebe, Trevor Darrell, Zhangyang Wang, and Humphrey Shi. Pair diffusion: A comprehensive multimodal object-level image editor. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8609–8618, 2024.

[19] Google. Experiment with gemini 2.0 flash native image generation. Technical report, Google AI Studio, 2025.

[20] Qin Guo and Tianwei Lin. Focus on your instruction: Fine-grained and multi-instruction image editing by attention modulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6986–6996, 2024.

[21] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training, 2022.

[22] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021.

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[24] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.

[25] 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, 2021.

[26] Lianghua Huang, Di Chen, Yu Liu, Yujun Shen, Deli Zhao, and Jingren Zhou. Composer: Creative and controllable image synthesis with composable conditions. *arXiv preprint arXiv:2302.09778*, 2023.

[27] Yi Huang, Jiancheng Huang, Yifan Liu, Mingfu Yan, Jiaxi Lv, Jianzhuang Liu, Wei Xiong, He Zhang, Liangliang Cao, and Shifeng Chen. Diffusion model-based image editing: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–27, 2025.

[28] 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.

[29] Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, et al. Gpt-4o system card, 2024.

[30] Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.

[31] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

[32] Duong H Le, Tuan Pham, Sangho Lee, Christopher Clark, Aniruddha Kembhavi, Stephan Mandt, Ranjay Krishna, and Jiasen Lu. One diffusion to generate them all. *arXiv preprint arXiv:2411.16318*, 2024.

[33] Shufan Li, Harkanwar Singh, and Aditya Grover. Instructany2pix: Flexible visual editing via multimodal instruction following. *arXiv preprint arXiv:2312.06738*, 2023.

[34] Sijia Li, Chen Chen, and Haonan Lu. Moecontroller: Instruction-based arbitrary image manipulation with mixture-of-expert controllers. *arXiv preprint arXiv:2309.04372*, 2023.

[35] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22511–22521, 2023.

[36] Long Lian, Boyi Li, Adam Yala, and Trevor Darrell. Llm-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models, 2024.

[37] Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, et al. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025.

[38] Chaojie Mao, Jingfeng Zhang, Yulin Pan, Zeyinzi Jiang, Zhen Han, Yu Liu, and Jingren Zhou. Ace++: Instruction-based image creation and editing via context-aware content filling, 2025.

[39] Konstantin Mishchenko and Aaron Defazio. Prodigy: An expeditiously adaptive parameter-free learner. *arXiv preprint arXiv:2306.06101*, 2023.

[40] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 4296–4304, 2024.

[41] Thanh Tam Nguyen, Zhao Ren, Trinh Pham, Thanh Trung Huynh, Phi Le Nguyen, Hongzhi Yin, and Quoc Viet Hung Nguyen. Instruction-guided editing controls for images and multimedia: A survey in llm era. *arXiv preprint arXiv:2411.09955*, 2024.

[42] Byong Mok Oh, Max Chen, Julie Dorsey, and Frédo Durand. Image-based modeling and photo editing. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 433–442, 2001.

[43] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[44] Zexu Pan, Zhaojie Luo, Jichen Yang, and Haizhou Li. Multi-modal attention for speech emotion recognition. *arXiv preprint arXiv:2009.04107*, 2020.

[45] Rishubh Parihar, VS Sachidanand, Sabariswaran Mani, Tejan Karmali, and R Venkatesh Babu. Precisecontrol: Enhancing text-to-image diffusion models with fine-grained attribute control. In *European Conference on Computer Vision*, pages 469–487. Springer, 2024.

[46] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[47] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[48] Can Qin, Shu Zhang, Ning Yu, Yihao Feng, Xinyi Yang, Yingbo Zhou, Huan Wang, Juan Carlos Niebles, Caiming Xiong, Silvio Savarese, et al. Unicontrol: A unified diffusion model for controllable visual generation in the wild. *arXiv preprint arXiv:2305.11147*, 2023.

[49] Scott Reed and Nando De Freitas. Neural programmer-interpreters. *arXiv preprint arXiv:1511.06279*, 2015.

[50] 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.

[51] Jean-Francois Rivest, Pierre Soille, and Serge Beucher. Morphological gradients. *Journal of Electronic Imaging*, 2(4):326–336, 1993.

[52] 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.

[53] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[54] Noam Rotstein, Gal Yona, Daniel Silver, Roy Velich, David Bensaïd, and Ron Kimmel. Pathways on the image manifold: Image editing via video generation. *arXiv preprint arXiv:2411.16819*, 2024.

[55] Khairul Anuar Mat Said and Asral Bahari Jambek. Analysis of image processing using morphological erosion and dilation. In *Journal of Physics: Conference Series*, volume 2071, page 012033. IOP Publishing, 2021.

[56] Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8871–8879, 2024.

[57] Yichun Shi, Peng Wang, and Weilin Huang. Seededit: Align image re-generation to image editing. *arXiv preprint arXiv:2411.06686*, 2024.

[58] Yujun Shi, Chuhui Xue, Jun Hao Liew, Jiachun Pan, Hanshu Yan, Wenqing Zhang, Vincent YF Tan, and Song Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8839–8849, 2024.

[59] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[60] Zhenxiong Tan, Songhua Liu, Xingyi Yang, Qiaochu Xue, and Xinchao Wang. Ominicontrol: Minimal and universal control for diffusion transformer. *arXiv preprint arXiv:2411.15098*, 2024.

[61] Zhenxiong Tan, Qiaochu Xue, Xingyi Yang, Songhua Liu, and Xinchao Wang. Ominicontrol2: Efficient conditioning for diffusion transformers. *arXiv preprint arXiv:2503.08280*, 2025.

[62] Nikolaos Tsagkas, Jack Rome, Subramanian Ramamoorthy, Oisin Mac Aodha, and Chris Xiaoxuan Lu. Click to grasp: Zero-shot precise manipulation via visual diffusion descriptors. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11610–11617. IEEE, 2024.

[63] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[64] Zhenyu Wang, Aoxue Li, Zhenguo Li, and Xihui Liu. Genartist: Multimodal llm as an agent for unified image generation and editing. *Advances in Neural Information Processing Systems*, 37:128374–128395, 2024.

[65] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[66] Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, Ze Liu, Ziyi Xia, Chaofan Li, Haoge Deng, Jiahao Wang, Kun Luo, Bo Zhang, Defu Lian, Xinlong Wang, Zhongyuan Wang, Tiejun Huang, and Zheng Liu. Omnigen2: Exploration to advanced multimodal generation, 2025.

[67] Shaojin Wu, Mengqi Huang, Wenxu Wu, Yufeng Cheng, Fei Ding, and Qian He. Less-to-more generalization: Unlocking more controllability by in-context generation. *arXiv preprint arXiv:2504.02160*, 2025.

[68] Bin Xia, Yuechen Zhang, Jingyao Li, Chengyao Wang, Yitong Wang, Xinglong Wu, Bei Yu, and Jiaya Jia. Dreamomni: Unified image generation and editing. *arXiv preprint arXiv:2412.17098*, 2024.

[69] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. *arXiv preprint arXiv:2409.11340*, 2024.

[70] Xingqian Xu, Jiayi Guo, Zhangyang Wang, Gao Huang, Irfan Essa, and Humphrey Shi. Prompt-free diffusion: Taking" text" out of text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8682–8692, 2024.

[71] Shiyuan Yang, Xiaodong Chen, and Jing Liao. Uni-paint: A unified framework for multimodal image inpainting with pretrained diffusion model. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 3190–3199, 2023.

[72] Siwei Yang, Mude Hui, Bingchen Zhao, Yuyin Zhou, Nataniel Ruiz, and Cihang Xie. `Complex-Edit`: Cot-like instruction generation for complexity-controllable image editing benchmark, 2025.

[73] Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Anyedit: Mastering unified high-quality image editing for any idea. *arXiv preprint arXiv:2411.15738*, 2024.

[74] Haobo Yuan, Xiangtai Li, Tao Zhang, Zilong Huang, Shilin Xu, Shunping Ji, Yunhai Tong, Lu Qi, Jiashi Feng, and Ming-Hsuan Yang. Sa2va: Marrying sam2 with llava for dense grounded understanding of images and videos, 2025.

[75] Hong Zhang, Zhongjie Duan, Xingjun Wang, Yingda Chen, Yuze Zhao, and Yu Zhang. Nexus-gen: A unified model for image understanding, generation, and editing. *arXiv preprint arXiv:2504.21356*, 2025.

[76] 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*, 36:31428–31449, 2023.

[77] 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.

[78] Xinyu Zhang, Mengxue Kang, Fei Wei, Shuang Xu, Yuhe Liu, and Lin Ma. Tie: Revolutionizing text-based image editing for complex-prompt following and high-fidelity editing, 2024.

[79] Yuxuan Zhang, Yirui Yuan, Yiren Song, Haofan Wang, and Jiaming Liu. Easycontrol: Adding efficient and flexible control for diffusion transformer, 2025.

[80] Zechuan Zhang, Ji Xie, Yu Lu, Zongxin Yang, and Yi Yang. In-context edit: Enabling instructional image editing with in-context generation in large scale diffusion transformer, 2025.

[81] Haozhe Zhao, Xiaojian Shawn 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. *Advances in Neural Information Processing Systems*, 37:3058–3093, 2024.

[82] Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K Wong. Uni-controlnet: All-in-one control to text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36:11127–11150, 2023.

Technical Appendices and Supplementary Material

In this part, we provide additional algorithm illustration, implementation details, more comparison results, more visualization results, and more analysis and discussions of the proposed approach.

A Algorithm Illustration

To better elaborate the details of the proposed IEAP, we provide an algorithmic illustration for the whole pipeline in Alg. 1.

Algorithm 1 IEAP: Image Editing As Programs

Input:

- I: input image path
- T: original instruction
- $\{\texttt{RoI\_Localization}, \texttt{RoI\_Inpainting}, \ldots, \texttt{Global\_Transformation}\}$: editing primitives
- $\texttt{cot\_with\_gpt}(\cdot)$: CoT prompt to GPT–4o
- $\texttt{extract\_instructions}(\cdot)$: parse CoT output
- $\texttt{infer\_with\_DiT}(\texttt{op}, \cdot)$: invoke DiT for primitive op
- $\texttt{roi\_localization}(I, instr)$: returns mask for region of interest
- $\texttt{fusion}(I_1, I_2)$: blends two intermediate outputs
- $\texttt{layout\_change}(I, instr)$: compute geometric transform

Output: final edited image I^*

$uri \leftarrow \texttt{encode\_image\_to\_datauri}(I)$
$(\mathcal{C}, \mathcal{T}) \leftarrow \texttt{cot\_with\_gpt}(uri, T)$ ▷ Categories and instructions
$I^{(0)} \leftarrow I$
for $i = 1$ to $|\mathcal{C}|$ **do**
  $cat \leftarrow \mathcal{C}[i], \quad instr \leftarrow \mathcal{T}[i]$
  if $cat \in \{\texttt{Add}, \texttt{Remove}, \texttt{Replace}\}$ **then**
    $M \leftarrow \texttt{roi\_localization}(I^{(i-1)}, instr)$
    $I' \leftarrow \texttt{infer\_with\_DiT}(\texttt{RoI Inpainting}, M, instr)$
    $I^{(i)} \leftarrow I'$
  else if $cat = \texttt{Action Change}$ **then**
    $M \leftarrow \texttt{roi\_localization}(I^{(i-1)}, instr)$
    $I_{bg} \leftarrow \texttt{infer\_with\_DiT}(\texttt{RoI Inpainting}, M, instr)$
    $I_{act} \leftarrow \texttt{infer\_with\_DiT}(\texttt{RoI Editing}, I^{(i-1)}, instr)$
    $I^{(i)} \leftarrow \texttt{infer\_with\_DiT}(\texttt{RoI Compositing}, \texttt{fusion}(I_{bg}, I_{act}), instr)$
  else if $cat \in \{\texttt{Move}, \texttt{Resize}\}$ **then**
    $M \leftarrow \texttt{roi\_localization}(I^{(i-1)}, instr)$
    $I_{bg} \leftarrow \texttt{infer\_with\_DiT}(\texttt{RoI Inpainting}, M, instr)$
    $I_{lc} \leftarrow \texttt{layout\_change}(I^{(i-1)}, instr)$
    $I^{(i)} \leftarrow \texttt{infer\_with\_DiT}(\texttt{RoI Compositing}, \texttt{fusion}(I_{bg}, I_{lc}), instr)$
  else if $cat \in \{\texttt{Appearance Change}, \texttt{Background Change},$
  $\texttt{Color Change}, \texttt{Material Change}, \texttt{Expression Change}\}$ **then**
    $I^{(i)} \leftarrow \texttt{infer\_with\_DiT}(\texttt{RoI Editing}, I^{(i-1)}, instr)$
  else if $cat \in \{\texttt{Tone Transfer}, \texttt{Style Change}\}$ **then**
    $I^{(i)} \leftarrow \texttt{infer\_with\_DiT}(\texttt{Global Transformation}, I^{(i-1)}, instr)$
  else
    raise ValueError("Invalid category: "cat")
  end if
end for
return $I^{(|\mathcal{C}|)}$

B Implementation Details

In this section, we present the prompts employed to leverage a VLM for CoT reasoning over complex instructions, providing further details on the layout-adjustment prompts.

Below are the detailed prompts used to invoke the VLM for the CoT process on complex instructions:

Now you are an expert in image editing. Based on the given single image, what atomic image editing instructions should be if the user wants to {instruction}? Let's think step by step.
Atomic instructions include 13 categories as follows:
- Add: Introduce a new object, person, or element into the image, e.g.: add a car on the road
- Remove: Eliminate an existing object or element from the image, e.g.: remove the sofa in the image
- Color Change: Modify the color of a specific object, e.g.: change the color of the shoes to blue
- Material Change: Alter the surface material or texture of an object, e.g.: change the material of the sign like stone
- Action Change: Modify the pose or action of an instance, e.g.: change the action of the boy to raising hands
- Expression Change: Adjust the facial expression, e.g.: change the expression to smiling
- Replace: Substitute one object in the image with a different object, e.g.: replace the coffee with an apple
- Background Change: Change the background scene to another, e.g.: change the background into forest
- Appearance Change: Modify visual attributes such as patterns or accessories, e.g.: make the cup have a floral pattern
- Move: Change the spatial position of an object within the image, e.g.: move the plane to the left
- Resize: Adjust the scale or size of an object, e.g.: enlarge the clock
- Tone Transfer: Change the global atmosphere or lighting conditions, e.g.: change the weather to foggy
- Style Change: Modify the entire image to adopt a different visual style, e.g.: make the style of the image to cartoon
Respond *only* with a numbered list. Each line must begin with the category in square brackets, then the instruction. Please strictly follow the atomic categories. The operation (what) and the target (to what) are crystal clear. Do not split replace to add and remove. Always place [Tone Transfer] and [Style Change] instructions at the end of the list.
For example:
1. [Add] add a car on the road
2. [Color Change] change the color of the shoes to blue
3. [Move] move the lamp to the left
Do not include any extra text, explanations, JSON or markdown, just the list.

Below are the detailed prompts used to adjust the layout of move and resize operations:

You are an intelligent bounding box editor. I will provide you with the current bounding boxes and the editing instruction. Your task is to generate the new bounding boxes after editing. Let's think step by step.
The images are of size 512x512. The top-left corner has coordinate [0, 0]. The bottom-right corner has coordinnate [512, 512]. The bounding boxes should not overlap or go beyond the image boundaries. Each bounding box should be in the format of (object name, [top-left x coordinate, top-left y coordinate, bottom-right x coordinate, bottom-right y coordinate]).
Do not add new objects or delete any object provided in the bounding boxes. Do not change the size or the shape of any object unless the instruction requires so.
Please consider the semantic information of the layout. When resizing, keep the bottom-left corner fixed by default. When swaping locations, change according to the center point.
If needed, you can make reasonable guesses. Please refer to the examples below:
Input bounding boxes: [("bed", [50, 300, 450, 450]), ("pillow", [200, 200, 300, 230])]
Editing instruction: Move the pillow to the left side of the bed.
Output bounding boxes: [("bed", [50, 300, 450, 450]), ("pillow", [70, 270, 170, 300])]

Input bounding boxes: [('a green car', [21, 281, 232, 440]), ('a blue truck', [269, 283, 478, 443]), ('a red air balloon', [66, 8, 211, 143]), ('a bird', [296, 42, 439, 142])]
Editing instruction: Move the car to the right.
Output bounding boxes: [('a green car', [81, 281, 292, 440]), ('a blue truck', [269, 283, 478, 443]), ('a red air balloon', [66, 8, 211, 143]), ('a bird', [296, 42, 439, 142])]
Input bounding boxes: [("sofa", [100, 300, 400, 400]), ("dog", [150, 250, 250, 300])]
Editing instruction: Enlarge the dog.
Output bounding boxes: [("sofa", [100, 300, 400, 400]), ("dog", [150, 225, 300, 300])]
Input bounding boxes: [("chair", [100, 350, 200, 450]), ("lamp", [300, 200, 360, 300])]
Editing instruction: Swap the location of the chair and the lamp.
Output bounding boxes: [("chair", [280, 200, 380, 300]), ("lamp", [120, 350, 180, 450])]
Now, the current bounding boxes is {bbox}, the instruction is {instruction}.

Below are the detailed prompts used to adjust the layout of add operations:

You are an intelligent bounding box editor. I will provide you with the current bounding boxes and an add editing instruction. Your task is to determine the new bounding box of the added object. Let's think step by step.
The images are of size 512x512. The top-left corner has coordinate [0, 0]. The bottom-right corner has coordinnate [512, 512].
The bounding boxes should not go beyond the image boundaries. The new box must be at least as large as needed to encompass the object. Each bounding box should be in the format of (object name, [top-left x coordinate, top-left y coordinate, bottom-right x coordinate, bottom-right y coordinate]).
Do not delete any object provided in the bounding boxes. Please consider the semantic information of the layout, preserve semantic relations.
If needed, you can make reasonable guesses. Please refer to the examples below:
Input bounding boxes: [('a green car', [21, 281, 232, 440])]
Editing instruction: Add a bird on the green car.
Output bounding boxes: [('a bird', [80, 150, 180, 281]), ('a green car', [21, 281, 232, 440])]
Input bounding boxes: [('stool', [300, 350, 380, 450])]
Editing instruction: Add a cat to the left of the stool.
Output bounding boxes: [('a cat', [180, 300, 300, 450])]
Input bounding boxes: [('the white cat', [200, 300, 320, 420])]
Editing instruction: Add a hat on the white cat.
Output bounding boxes: [('the white hat', [200, 260, 320, 310]), ('cat', [200, 300, 320, 420])]
Now, the current bounding boxes is {bbox}, the instruction is {instruction}.

C More Quantitative Results

Method	CLIP$_{im}$ ↑	CLIP$_{out}$ ↑	L1 ↓	DINO ↑	GPT$_{IF}$ ↑	GPT$_{FC}$ ↑	GPT$_{AQ}$ ↑	GPT$_{avg}$ ↑
InstructPix2Pix	0.847	0.264	0.092	0.829	4.50	4.40	4.26	4.39
MagicBrush	0.889	<u>0.277</u>	0.068	0.892	<u>4.66</u>	4.76	<u>4.62</u>	4.68
UltraEdit	0.897	0.274	**0.056**	0.909	3.36	4.24	4.22	3.94
ICEdit	<u>0.925</u>	<u>0.277</u>	0.057	<u>0.915</u>	4.60	<u>4.80</u>	**4.76**	**4.72**
IEAP(Ours)	**0.928**	**0.278**	**0.056**	**0.917**	**4.68**	**4.84**	4.60	<u>4.71</u>

Table 5: Quantitative comparison results on AnyEdit Add test set.

Method	CLIP$_{im}$ ↑	CLIP$_{out}$ ↑	L1 ↓	DINO ↑	GPT$_{IF}$ ↑	GPT$_{FC}$ ↑	GPT$_{AQ}$ ↑	GPT$_{avg}$ ↑
InstructPix2Pix	0.800	0.202	0.108	0.721	2.74	3.42	3.20	3.12
MagicBrush	0.853	<u>0.211</u>	0.083	0.800	3.08	3.60	3.18	3.29
UltraEdit	0.846	<u>0.211</u>	0.066	0.802	2.50	3.54	3.44	3.16
ICEdit	<u>0.895</u>	0.212	**0.054**	<u>0.875</u>	<u>4.06</u>	**4.48**	**4.32**	**4.29**
IEAP(Ours)	**0.916**	**0.230**	<u>0.057</u>	**0.886**	**4.18**	<u>3.88</u>	<u>3.66</u>	<u>3.91</u>

Table 6: Quantitative comparison results on AnyEdit Remove test set.

Method	CLIP$_{im}$ ↑	CLIP$_{out}$ ↑	L1 ↓	DINO ↑	GPT$_{IF}$ ↑	GPT$_{FC}$ ↑	GPT$_{AQ}$ ↑	GPT$_{avg}$ ↑
InstructPix2Pix	0.766	0.234	0.179	0.588	3.72	3.68	3.80	3.73
MagicBrush	0.806	0.248	0.148	0.671	4.52	4.48	4.38	4.46
UltraEdit	0.779	0.242	0.142	0.621	3.80	4.40	4.40	4.20
ICEdit	0.797	0.228	0.128	0.614	3.68	4.02	4.04	3.91
IEAP(Ours)	**0.866**	**0.252**	**0.099**	**0.701**	**4.68**	**4.68**	**4.48**	**4.61**

Table 7: Quantitative comparison results on AnyEdit Replace test set.

Method	CLIP$_{im}$ ↑	CLIP$_{out}$ ↑	L1 ↓	DINO ↑	GPT$_{IF}$ ↑	GPT$_{FC}$ ↑	GPT$_{AQ}$ ↑	GPT$_{avg}$ ↑
InstructPix2Pix	0.829	0.254	0.164	0.774	3.46	3.84	3.58	3.63
MagicBrush	0.831	0.266	0.156	0.784	2.96	4.28	4.28	3.84
UltraEdit	0.847	0.259	0.157	0.781	2.92	4.22	4.24	3.79
ICEdit	0.827	0.255	**0.152**	0.745	2.68	4.04	4.04	3.59
IEAP(Ours)	**0.848**	**0.267**	0.154	**0.798**	**4.66**	**4.86**	**4.68**	**4.73**

Table 8: Quantitative comparison results on AnyEdit Action Change test set.

Method	CLIP$_{im}$ ↑	CLIP$_{out}$ ↑	L1 ↓	DINO ↑	GPT$_{IF}$ ↑	GPT$_{FC}$ ↑	GPT$_{AQ}$ ↑	GPT$_{avg}$ ↑
InstructPix2Pix	0.881	0.219	0.127	0.771	3.82	4.44	4.36	4.21
MagicBrush	0.902	0.219	0.088	0.828	2.94	3.94	3.90	3.59
UltraEdit	0.923	0.211	0.074	0.867	3.48	4.40	**4.40**	4.09
ICEdit	0.944	0.213	0.063	0.868	3.28	**4.64**	4.30	4.07
IEAP(Ours)	**0.963**	**0.223**	**0.058**	**0.903**	**3.88**	4.44	4.38	**4.23**

Table 9: Quantitative comparison results on AnyEdit Relation test set.

Method	CLIP$_{im}$ ↑	CLIP$_{out}$ ↑	L1 ↓	DINO ↑	GPT$_{IF}$ ↑	GPT$_{FC}$ ↑	GPT$_{AQ}$ ↑	GPT$_{avg}$ ↑
InstructPix2Pix	0.831	0.241	0.124	0.746	2.94	3.56	3.62	3.37
MagicBrush	0.875	0.258	0.094	0.802	2.80	3.88	4.00	3.56
UltraEdit	0.908	0.262	0.073	0.889	3.22	**4.38**	**4.38**	4.00
ICEdit	0.895	0.253	0.074	0.841	3.14	4.28	4.26	3.89
IEAP(Ours)	**0.923**	**0.263**	**0.066**	**0.921**	**4.38**	4.32	4.28	**4.32**

Table 10: Quantitative comparison results on AnyEdit Resize test set.

Method	CLIP$_{im}$ ↑	CLIP$_{out}$ ↑	L1 ↓	DINO ↑	GPT$_{IF}$ ↑	GPT$_{FC}$ ↑	GPT$_{AQ}$ ↑	GPT$_{avg}$ ↑
InstructPix2Pix	0.815	0.280	0.139	0.744	3.60	4.08	3.92	3.87
MagicBrush	0.852	**0.294**	0.094	0.815	3.96	4.32	3.98	4.09
UltraEdit	0.857	0.277	**0.068**	**0.845**	4.04	4.62	4.42	4.36
ICEdit	0.847	0.273	0.085	0.808	4.04	4.42	4.16	4.21
IEAP(Ours)	**0.886**	0.285	0.082	0.833	**4.06**	**4.72**	**4.80**	**4.53**

Table 11: Quantitative comparison results on AnyEdit Appearance test set.

Method	CLIP$_{im}$ ↑	CLIP$_{out}$ ↑	L1 ↓	DINO ↑	GPT$_{IF}$ ↑	GPT$_{FC}$ ↑	GPT$_{AQ}$ ↑	GPT$_{avg}$ ↑
InstructPix2Pix	0.725	0.224	0.216	0.582	3.40	3.60	3.44	3.48
MagicBrush	0.746	0.230	0.228	0.567	4.58	4.38	4.46	4.47
UltraEdit	0.796	**0.257**	0.169	0.747	3.48	4.36	3.14	3.66
ICEdit	0.799	0.241	0.166	0.757	3.04	4.16	3.88	3.69
IEAP(Ours)	**0.801**	0.243	**0.165**	**0.759**	**4.74**	**4.68**	**4.70**	**4.71**

Table 12: Quantitative comparison results on AnyEdit Background Change test set.

Method	CLIP$_{im}$ ↑	CLIP$_{out}$ ↑	L1 ↓	DINO ↑	GPT$_{IF}$ ↑	GPT$_{FC}$ ↑	GPT$_{AQ}$ ↑	GPT$_{avg}$ ↑
InstructPix2Pix	0.886	0.279	0.120	**0.876**	3.60	4.40	4.00	4.00
MagicBrush	0.898	**0.282**	0.087	0.869	4.20	**4.82**	4.62	4.55
UltraEdit	0.890	0.280	0.065	0.87	3.80	4.40	4.20	4.13
ICEdit	0.896	0.278	0.073	0.849	**4.72**	4.80	4.64	**4.72**
IEAP(Ours)	**0.911**	0.276	**0.059**	**0.876**	4.62	4.72	**4.78**	4.71

Table 13: Quantitative comparison results on AnyEdit Color Change test set.

Method	CLIP$_{im}$ ↑	L1 ↓	DINO ↑	GPT$_{IF}$ ↑	GPT$_{FC}$ ↑	GPT$_{AQ}$ ↑	GPT$_{avg}$ ↑
InstructPix2Pix	0.776	0.068	0.936	3.74	4.60	4.30	4.21
MagicBrush	0.770	0.064	0.940	3.86	4.48	4.18	4.17
UltraEdit	0.699	0.073	0.907	3.14	4.10	3.80	3.68
ICEdit	0.796	0.065	0.943	3.16	4.60	4.30	4.02
IEAP(Ours)	**0.882**	**0.052**	**0.945**	**4.34**	**4.72**	**4.50**	**4.52**

Table 14: Quantitative comparison results on Expression test set.

Method	CLIP$_{im}$ ↑	L1 ↓	DINO ↑	GPT$_{IF}$ ↑	GPT$_{FC}$ ↑	GPT$_{AQ}$ ↑	GPT$_{avg}$ ↑
InstructPix2Pix	0.746	0.130	0.549	4.00	4.18	4.04	4.07
MagicBrush	0.778	0.110	0.621	3.36	4.06	3.84	3.75
UltraEdit	0.765	0.086	0.598	3.34	4.28	4.04	3.89
ICEdit	0.787	0.086	0.616	3.48	3.92	3.58	3.66
IEAP(Ours)	**0.826**	**0.055**	**0.696**	**4.08**	**4.48**	**4.18**	**4.25**

Table 15: Quantitative comparison results on Material Change test set.

Method	CLIP$_{im}$ ↑	L1 ↓	DINO ↑	GPT$_{IF}$ ↑	GPT$_{FC}$ ↑	GPT$_{AQ}$ ↑	GPT$_{avg}$ ↑
InstructPix2Pix	0.710	0.212	0.463	3.56	4.32	3.94	3.94
MagicBrush	0.692	0.214	0.440	3.12	4.64	4.00	3.92
UltraEdit	0.703	0.201	0.467	4.02	4.8	**4.62**	4.48
ICEdit	0.706	0.219	0.458	4.04	**4.82**	4.36	4.41
IEAP(Ours)	**0.922**	**0.097**	**0.915**	**4.44**	4.64	4.44	**4.51**

Table 16: Quantitative comparison results on AnyEdit Style Change test set.

Method	CLIP$_{im}$ ↑	CLIP$_{out}$ ↑	L1 ↓	DINO ↑	GPT$_{IF}$ ↑	GPT$_{FC}$ ↑	GPT$_{AQ}$ ↑	GPT$_{avg}$ ↑
InstructPix2Pix	0.822	0.260	**0.100**	0.821	3.72	4.48	3.92	4.04
MagicBrush	0.834	0.266	0.159	0.791	3.56	4.64	3.98	4.06
UltraEdit	0.804	**0.268**	0.201	0.767	4.12	4.62	4.26	4.33
ICEdit	0.812	0.260	0.157	0.748	4.06	**4.88**	**4.56**	4.50
IEAP(Ours)	**0.868**	**0.268**	0.116	**0.843**	**4.44**	4.64	4.44	**4.51**

Table 17: Quantitative comparison results on AnyEdit Tone Transfer test set.

Method	CLIP$_{im}$ ↑	L1 ↓	DINO ↑	GPT$_{IF}$ ↑	GPT$_{FC}$ ↑	GPT$_{AQ}$ ↑	GPT$_{avg}$ ↑
InstructPix2Pix	0.815	0.134	0.647	3.40	4.04	**4.80**	4.08
MagicBrush	0.835	0.081	0.697	1.82	3.56	3.50	2.96
UltraEdit	0.833	0.066	0.756	2.58	4.02	4.02	3.54
ICEdit	0.906	**0.042**	**0.842**	2.98	4.40	3.40	3.59
IEAP(Ours)	**0.908**	0.056	0.794	**3.42**	**4.48**	4.46	**4.12**

Table 18: Quantitative comparison results on AnyEdit Counting test set.

Method	$\text{CLIP}_{im}\uparrow$	L1 ↓	DINO ↑	$\text{GPT}_{IF}\uparrow$	$\text{GPT}_{FC}\uparrow$	$\text{GPT}_{AQ}\uparrow$	$\text{GPT}_{avg}\uparrow$
InstructPix2Pix	0.773	0.208	0.581	3.46	4.18	4.08	3.91
MagicBrush	0.806	0.174	0.631	2.98	3.88	4.04	3.63
UltraEdit	0.825	**0.167**	**0.669**	2.82	4.38	4.38	3.86
ICEdit	0.806	0.171	0.629	3.56	4.16	4.06	3.93
IEAP(Ours)	**0.833**	0.169	0.662	**3.88**	**4.44**	**4.52**	**4.28**

Table 19: Quantitative comparison results on AnyEdit Implicit Change test set.

Method	$\text{CLIP}_{im}\uparrow$	L1 ↓	DINO ↑	$\text{GPT}_{IF}\uparrow$	$\text{GPT}_{FC}\uparrow$	$\text{GPT}_{AQ}\uparrow$	$\text{GPT}_{avg}\uparrow$
InstructPix2Pix	0.887	0.111	0.858	**4.30**	4.50	4.30	**4.37**
MagicBrush	0.900	0.100	0.874	4.12	4.36	**4.54**	4.34
UltraEdit	0.922	**0.077**	0.911	3.24	4.4	4.36	4.00
ICEdit	0.898	0.079	0.864	4.16	4.46	4.20	4.27
IEAP(Ours)	**0.938**	0.084	**0.925**	4.18	**4.56**	4.38	**4.37**

Table 20: Quantitative comparison results on AnyEdit Move test set.

Method	$\text{CLIP}_{im}\uparrow$	$\text{CLIP}_{out}\uparrow$	L1 ↓	DINO ↑	$\text{GPT}_{IF}\uparrow$	$\text{GPT}_{FC}\uparrow$	$\text{GPT}_{AQ}\uparrow$	$\text{GPT}_{avg}\uparrow$
InstructPix2Pix	0.688	0.243	0.189	0.742	1.04	4.38	3.92	3.11
MagicBrush	0.680	0.255	0.156	0.786	1.02	4.48	4.10	3.20
UltraEdit	0.732	0.279	**0.147**	**0.843**	1.96	4.46	3.98	3.47
ICEdit	**0.810**	**0.289**	0.155	0.811	**4.18**	4.42	**4.68**	**4.43**
IEAP(Ours)	0.788	0.285	0.162	0.786	3.96	**4.58**	4.06	4.20

Table 21: Quantitative comparison results on AnyEdit Textual Change test set.

D More Visualization Results

In this section, we provide more visualization results, as shown below:

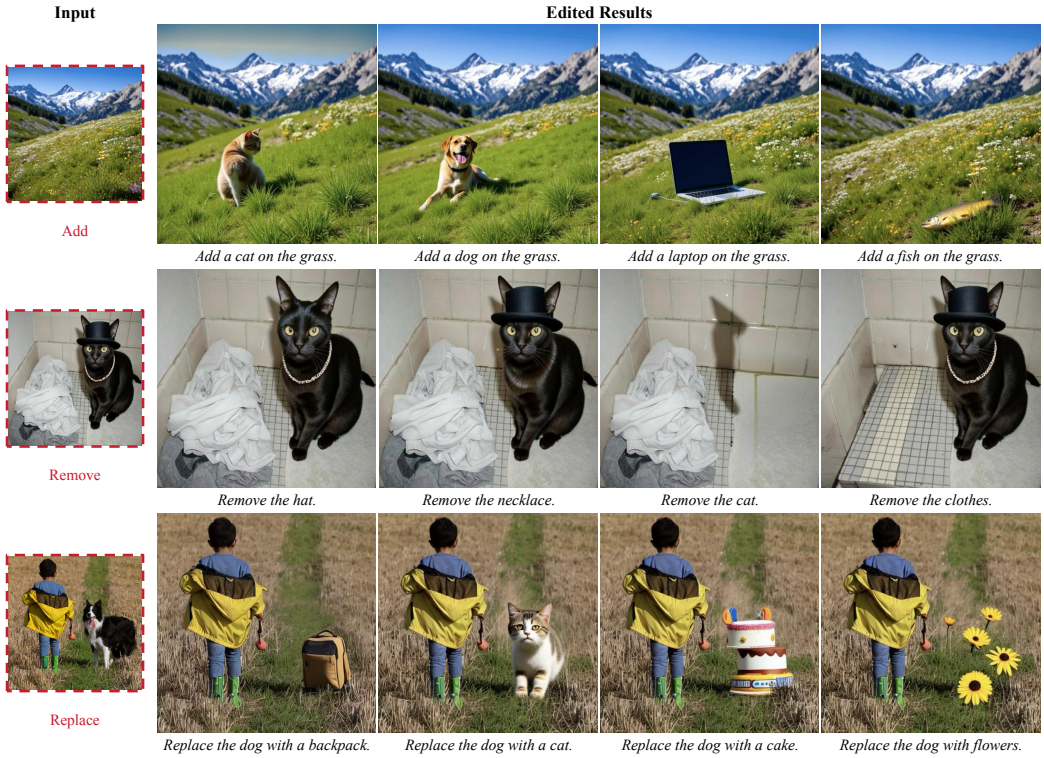

Figure 10: More Visualization Results.

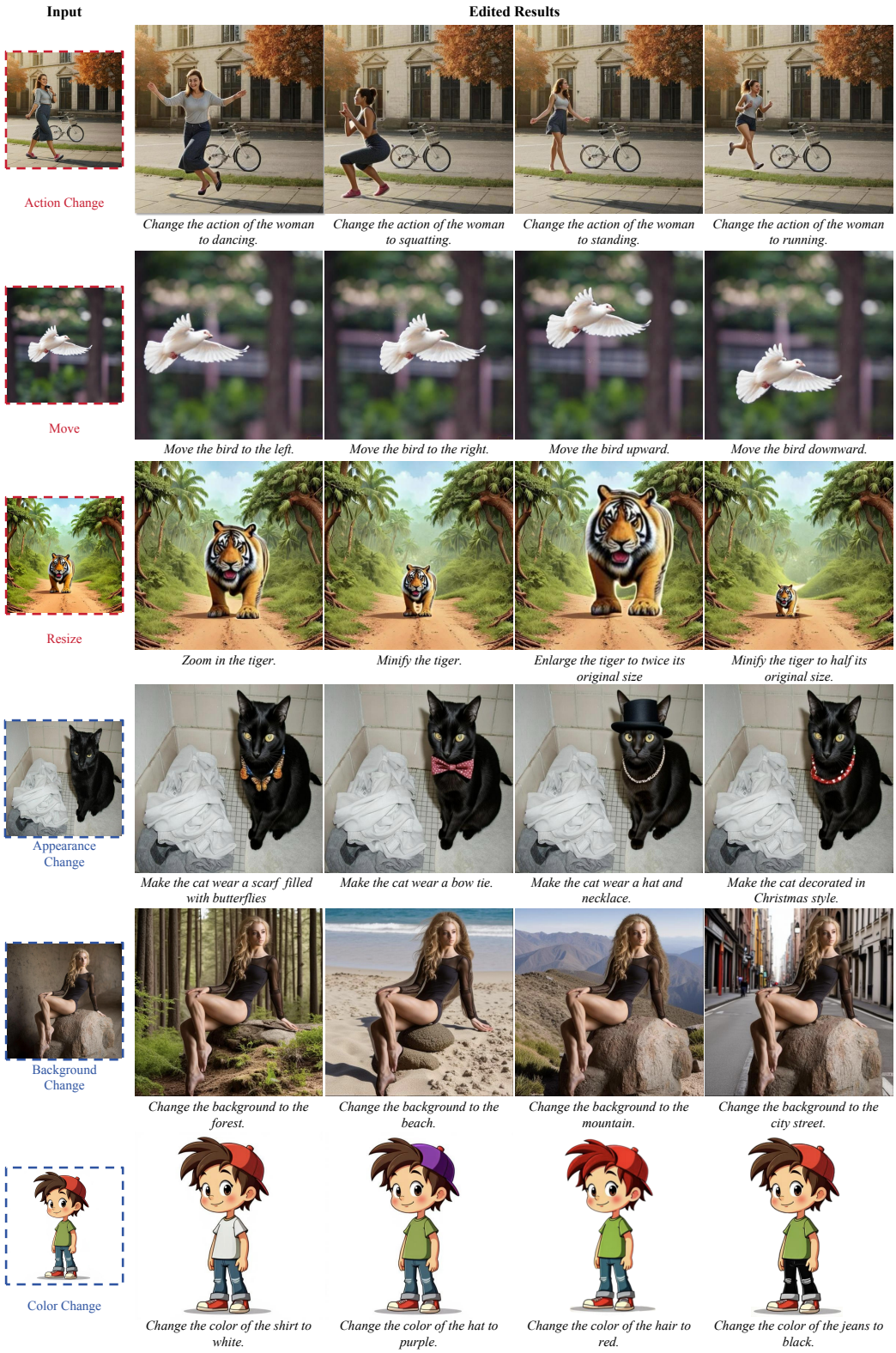

Figure 11: More Visualization Results.

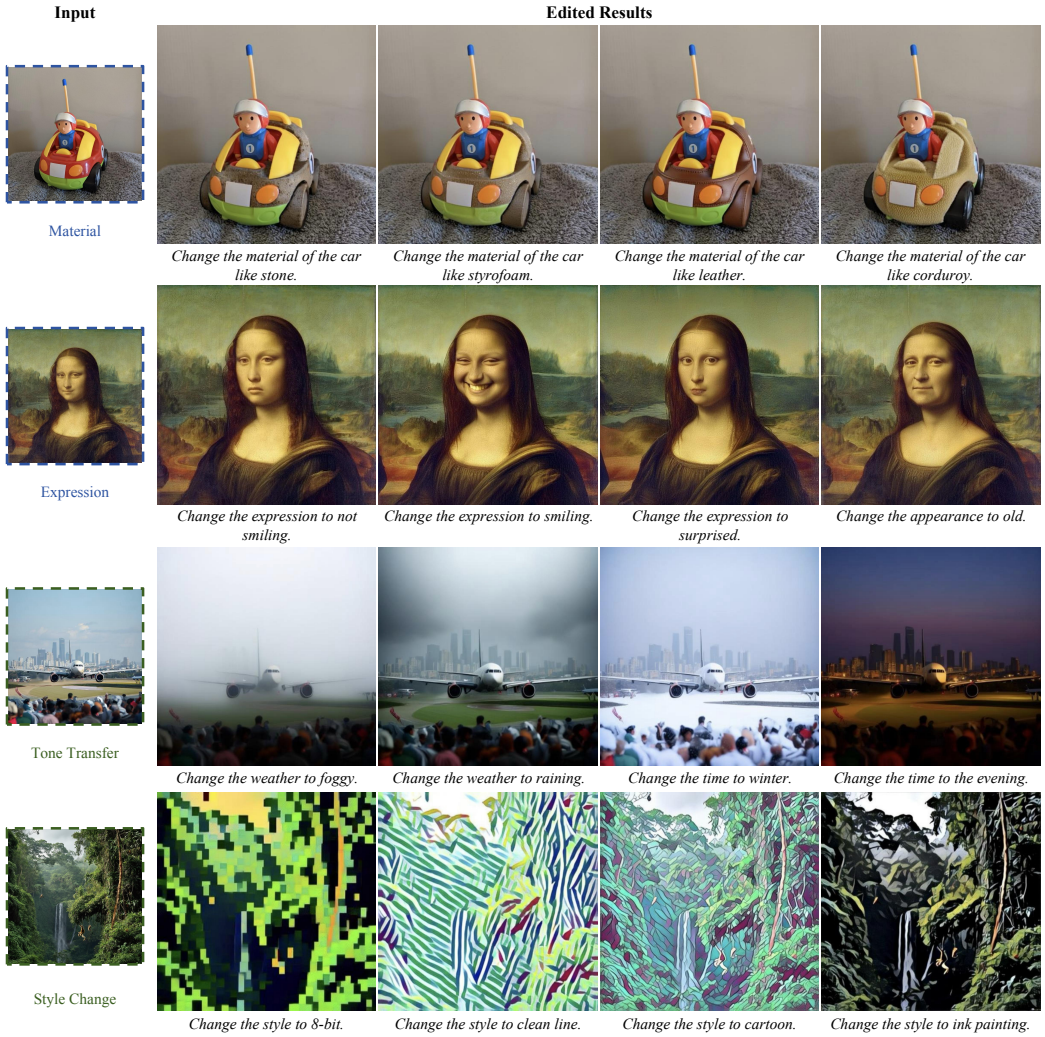

Figure 12: More Visualization Results.

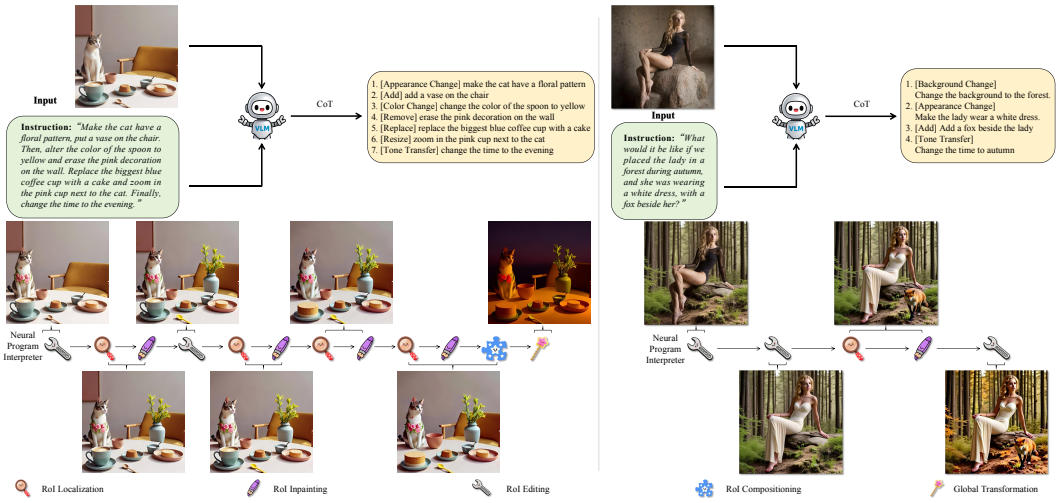

Figure 13: More Detailed Visualization Processes of the pipeline.

E Analysis and Discussions

E.1 Runtime Performance Analysis

We quantified the inference latency of IEAP across different editing types on a single NVIDIA H200 GPU. For local attribute editing and overall content editing, only one atomic diffusion-based step is required. In contrast, local semantic editing typically involves multiple atomic operations, depending on the specific editing intent:

$$\begin{aligned}
\text{Add/Replace:} &\quad 1 \times \text{RoI Localization} + 1 \times \text{Diffusion} + 2 \times \text{LLM Response},\\
\text{Remove:} &\quad 1 \times \text{RoI Localization} + 1 \times \text{Diffusion} + 1 \times \text{LLM Response},\\
\text{Action Change:} &\quad 2 \times \text{RoI Localization} + 3 \times \text{Diffusion} + 2 \times \text{LLM Response},\\
\text{Move/Resize:} &\quad 1 \times \text{RoI Localization} + 2 \times \text{Diffusion} + 2 \times \text{LLM Response}.
\end{aligned}$$

Empirical measurements show that each diffusion step takes approximately $5\,\mathrm{s}$ when using FLUX.1-dev and around $4\,\mathrm{s}$ with FLUX.1-schnell. The LLM response incurs about $1\,\mathrm{s}$ latency, while RoI localization requires roughly $2\,\mathrm{s}$. Based on these estimates, the end-to-end latencies for representative operations are approximately: Add/Replace: 8–9 s; Remove: 7–8 s; Action Change: 18–20 s; Move/Resize: 12–14 s; Other operations: typically 4–5 s.

On the AnyEdit test set, which includes 16 subtypes of editing, our method achieves a weighted average latency of approximately $7\,\mathrm{s}$ per edit. For comparison, the end-to-end editing systems InstructPix2Pix, MagicBrush, UltraEdit, and ICEdit exhibit average latencies of $4\,\mathrm{s}$, $4\,\mathrm{s}$, $1\,\mathrm{s}$, and $3\,\mathrm{s}$, respectively. Although IEAP incurs slightly higher inference latency, it provides faithful, modular, and compositional control over both simple and complex multi-step edits, extending beyond the capabilities of purely end-to-end approaches.

E.2 Limitations and Future Work

Limitations. Despite its strengths, IEAP exhibits several limitations in handling dynamic scenes and complex physical interactions. First, the RoI compositing may introduce geometric distortions or texture discontinuities when editing highly dynamic or non-rigid content, such as motion-blurred instances, and fluid or smoke effects. For example, in the task of "changing the cat's action to jumping," in Fig. 6, the rapid motion of fur can produce blurred regions that fail to blend naturally with the background. Second, RoI compositing struggles to simulate physically consistent lighting effects in scenes with reflective or refractive surfaces, sometimes resulting in mismatched shadow directions and illumination conflicts between edited objects and their environments. For example, in the task of "change the action of the woman to dancing," in Fig. 4, the shadows before and after editing remain the same, but the action of the woman has changed, so it is unnatural. Third, the DiT-based architecture and multi-stage atomic operations incur substantial inference latency for $5\,\mathrm{s}$ to $9\,\mathrm{s}$ per operation on a single H100 GPU, precluding real-time interactivity in applications such as AR/VR. Finally, the requirement for high-memory GPUs like NVIDIA H100 (80 GB) limits reproducibility for resource-constrained researchers, and multi-iteration editing can exacerbate image quality degradation over successive operations.

Future Work. As for future work, several avenues may be pursued to overcome the identified limitations. To begin with, physics-aware compositing techniques and motion-compensated inpainting could be explored to better accommodate dynamic blur and fluid effects, thereby ensuring seamless integration of non-rigid edits. Meanwhile, differentiable lighting models or neural rendering modules may be incorporated to enforce global illumination consistency, particularly in reflective and refractive contexts. On the performance front, model distillation, operation fusion, and sparse attention strategies could be investigated to reduce per-operation latency and facilitate interactive editing. To enhance accessibility, memory optimization and support for smaller-footprint architectures amenable to commodity GPUs may be implemented. Moreover, iterative refinement and error-correction mechanisms may be developed to mitigate quality degradation over successive editing steps. Furthermore, beyond still-image editing, an extension to video-based complex instruction editing could be considered, where temporal coherence and motion consistency present additional challenges and opportunities for dynamic, multi-step visual manipulation.

E.3 Societal Impacts and Ethical Safeguards

Positive Societal Impacts. The proposed IEAP framework introduces a modular and interpretable approach to complex image editing, which holds significant potential to benefit a range of creative and technical domains. By decomposing high-level visual instructions into atomic operations, IEAP enables users to perform multi-step edits with enhanced precision and control. This capability is particularly valuable in digital content creation, advertising, and education, where fine-grained manipulation of visual content is often required. For example, IEAP's ability to support structurally inconsistent modifications can streamline visual storytelling workflows or facilitate the generation of accurate scientific visualizations for publications and teaching materials. Furthermore, its potential extensions to fields such as medical imaging by enabling localized enhancement of diagnostic visuals, and accessibility technology by generating descriptive visual representations for users with visual impairments, demonstrate the framework's broader societal utility and interdisciplinary relevance.

Negative Societal Impacts and Ethical Safeguards. Despite its benefits, IEAP's high-fidelity editing capabilities also introduce ethical risks, particularly in the domains of misinformation and privacy. The framework's precision in altering visual content could be misused for the creation of deepfakes or manipulated images intended for disinformation, identity falsification, or reputational harm. Operations such as "Remove" or "Replace" could be exploited to tamper with sensitive or private imagery, potentially infringing on individual rights.

To address these concerns, the development and deployment of IEAP adhere to strict ethical standards. Specifically, safeguards include the implementation of data filtering pipelines, such as the use of GPT-4o-filtered subsets of AnyEdit and the compliance-oriented CelebHQ-FM dataset, to reduce harmful biases and content. Additionally, the modular nature of IEAP facilitates transparency and traceability in the editing process, supporting future content provenance systems designed to detect and flag manipulated media. All these safeguards jointly contribute to ongoing efforts in AI safety and accountability.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discussed the limitations in Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper fully disclose all the information needed to reproduce the main experimental results of the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Please refer to the supplemental material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We explained the experimental setting and details in Section 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Justification: We follow the other relevant works which do not include this part.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We explained the compute resources in Section 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Please refer to the supplemental material.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Please refer to the supplemental material.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited the original paper that produced the code package or dataset and the name of the license (e.g., CC-BY 4.0) is included for each asset.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subject.

    Guidelines: The paper does not involve crowdsourcing nor research with human subjects.

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.