

## 478 Technical Appendices and Supplementary Material

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

### 481 A Algorithm Illustration

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

---

#### Algorithm 1 IEAP: Image Editing As Programs

---

##### Input:

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

##### Output: final edited image $I^*$

```

1:  $uri \leftarrow \text{encode\_image\_to\_datauri}(I)$ 
2:  $(\mathcal{C}, \mathcal{T}) \leftarrow \text{cot\_with\_gpt}(uri, T)$  ▷ Categories and instructions
3:  $I^{(0)} \leftarrow I$ 
4: for  $i = 1$  to  $|\mathcal{C}|$  do
5:    $cat \leftarrow \mathcal{C}[i], \text{instr} \leftarrow \mathcal{T}[i]$ 
6:   if  $cat \in \{\text{Add, Remove, Replace}\}$  then
7:      $M \leftarrow \text{roi\_localization}(I^{(i-1)}, \text{instr})$ 
8:      $I' \leftarrow \text{infer\_with\_DiT}(\text{RoI Inpainting}, M, \text{instr})$ 
9:      $I^{(i)} \leftarrow I'$ 
10:  else if  $cat = \text{Action Change}$  then
11:     $M \leftarrow \text{roi\_localization}(I^{(i-1)}, \text{instr})$ 
12:     $I_{bg} \leftarrow \text{infer\_with\_DiT}(\text{RoI Inpainting}, M, \text{instr})$ 
13:     $I_{act} \leftarrow \text{infer\_with\_DiT}(\text{RoI Editing}, I^{(i-1)}, \text{instr})$ 
14:     $I^{(i)} \leftarrow \text{infer\_with\_DiT}(\text{RoI Compositing}, \text{fusion}(I_{bg}, I_{act}), \text{instr})$ 
15:  else if  $cat \in \{\text{Move, Resize}\}$  then
16:     $M \leftarrow \text{roi\_localization}(I^{(i-1)}, \text{instr})$ 
17:     $I_{bg} \leftarrow \text{infer\_with\_DiT}(\text{RoI Inpainting}, M, \text{instr})$ 
18:     $I_{lc} \leftarrow \text{layout\_change}(I^{(i-1)}, \text{instr})$ 
19:     $I^{(i)} \leftarrow \text{infer\_with\_DiT}(\text{RoI Compositing}, \text{fusion}(I_{bg}, I_{lc}), \text{instr})$ 
20:  else if  $cat \in \{\text{Appearance Change, Background Change,}$ 
21:  $\text{Color Change, Material Change, Expression Change}\}$  then
22:     $I^{(i)} \leftarrow \text{infer\_with\_DiT}(\text{RoI Editing}, I^{(i-1)}, \text{instr})$ 
23:  else if  $cat \in \{\text{Tone Transfer, Style Change}\}$  then
24:     $I^{(i)} \leftarrow \text{infer\_with\_DiT}(\text{Global Transformation}, I^{(i-1)}, \text{instr})$ 
25:  else
26:    raise ValueError("Invalid category: "cat")
27:  end if
28: end for
29: 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 coordinate [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])]

493

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}.

494

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

496

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 coordinate [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}.

497

498 **C More Quantitative Results**

Method	CLIP <sub>im</sub> ↑	CLIP <sub>out</sub> ↑	L1 ↓	DINO ↑	GPT <sub>IF</sub> ↑	GPT <sub>FC</sub> ↑	GPT <sub>AQ</sub> ↑	GPT <sub>avg</sub> ↑
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	<b>0.056</b>	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>	<b>4.76</b>	<b>4.72</b>
IEAP(Ours)	<b>0.928</b>	<b>0.278</b>	<b>0.056</b>	<b>0.917</b>	<b>4.68</b>	<b>4.84</b>	4.60	<u>4.71</u>

Table 4: Quantitative comparison results on AnyEdit Add test set.

Method	CLIP <sub>im</sub> ↑	CLIP <sub>out</sub> ↑	L1 ↓	DINO ↑	GPT <sub>IF</sub> ↑	GPT <sub>FC</sub> ↑	GPT <sub>AQ</sub> ↑	GPT <sub>avg</sub> ↑
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	<b>0.054</b>	<u>0.875</u>	<u>4.06</u>	<b>4.48</b>	<b>4.32</b>	<b>4.29</b>
IEAP(Ours)	<b>0.916</b>	<b>0.230</b>	<u>0.057</u>	<b>0.886</b>	<b>4.18</b>	<u>3.88</u>	<u>3.66</u>	<u>3.91</u>

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

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

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

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

Table 7: Quantitative comparison results on AnyEdit Action Change test set.

Method	CLIP <sub>im</sub> ↑	CLIP <sub>out</sub> ↑	L1 ↓	DINO ↑	GPT <sub>IF</sub> ↑	GPT <sub>FC</sub> ↑	GPT <sub>AQ</sub> ↑	GPT <sub>avg</sub> ↑
InstructPix2Pix	0.881	0.219	0.127	0.771	3.82	4.44	4.36	<u>4.21</u>
MagicBrush	0.902	<u>0.219</u>	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	<b>4.40</b>	4.09
ICEdit	<u>0.944</u>	0.213	<u>0.063</u>	<u>0.868</u>	3.28	<b>4.64</b>	4.30	4.07
IEAP(Ours)	<b>0.963</b>	<b>0.223</b>	<b>0.058</b>	<b>0.903</b>	<b>3.88</b>	<u>4.44</u>	<u>4.38</u>	<b>4.23</b>

Table 8: Quantitative comparison results on AnyEdit Relation test set.

Method	CLIP <sub>im</sub> ↑	CLIP <sub>out</sub> ↑	L1 ↓	DINO ↑	GPT <sub>IF</sub> ↑	GPT <sub>FC</sub> ↑	GPT <sub>AQ</sub> ↑	GPT <sub>avg</sub> ↑
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	<u>0.908</u>	<u>0.262</u>	<u>0.073</u>	<u>0.889</u>	<u>3.22</u>	<b>4.38</b>	<b>4.38</b>	<u>4.00</u>
ICEdit	0.895	0.253	0.074	0.841	3.14	4.28	4.26	<u>3.89</u>
IEAP(Ours)	<b>0.923</b>	<b>0.263</b>	<b>0.066</b>	<b>0.921</b>	<b>4.38</b>	<u>4.32</u>	<u>4.28</u>	<b>4.32</b>

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

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

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

Method	CLIP <sub>im</sub> ↑	CLIP <sub>out</sub> ↑	L1 ↓	DINO ↑	GPT <sub>IF</sub> ↑	GPT <sub>FC</sub> ↑	GPT <sub>AQ</sub> ↑	GPT <sub>avg</sub> ↑
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	<u>4.58</u>	<u>4.38</u>	<u>4.46</u>	<u>4.47</u>
UltraEdit	0.796	<b>0.257</b>	0.169	0.747	3.48	4.36	3.14	<u>3.66</u>
ICEdit	<u>0.799</u>	0.241	<u>0.166</u>	<u>0.757</u>	3.04	4.16	3.88	3.69
IEAP(Ours)	<b>0.801</b>	<u>0.243</u>	<b>0.165</b>	<b>0.759</b>	<b>4.74</b>	<b>4.68</b>	<b>4.70</b>	<b>4.71</b>

Table 11: Quantitative comparison results on AnyEdit Background Change test set.

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

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

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

Table 13: Quantitative comparison results on Expression test set.

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

Table 14: Quantitative comparison results on Material Change test set.

Method	CLIP <sub>im</sub> ↑	L1 ↓	DINO ↑	GPT <sub>IF</sub> ↑	GPT <sub>FC</sub> ↑	GPT <sub>AQ</sub> ↑	GPT <sub>avg</sub> ↑
InstructPix2Pix	<u>0.710</u>	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	<u>0.467</u>	4.02	4.8	<b>4.62</b>	<u>4.48</u>
ICEdit	0.706	<u>0.219</u>	<u>0.458</u>	<u>4.04</u>	<b>4.82</b>	4.36	<u>4.41</u>
IEAP(Ours)	<b>0.922</b>	<b>0.097</b>	<b>0.915</b>	<b>4.44</b>	4.64	<u>4.44</u>	<b>4.51</b>

Table 15: Quantitative comparison results on AnyEdit Style Change test set.

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

Table 16: Quantitative comparison results on AnyEdit Tone Transfer test set.

Method	CLIP <sub>im</sub> ↑	L1 ↓	DINO ↑	GPT <sub>IF</sub> ↑	GPT <sub>FC</sub> ↑	GPT <sub>AQ</sub> ↑	GPT <sub>avg</sub> ↑
InstructPix2Pix	0.815	0.134	0.647	<u>3.40</u>	4.04	<b>4.80</b>	<u>4.08</u>
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	<u>0.906</u>	<b>0.042</b>	<b>0.842</b>	2.98	<u>4.40</u>	3.40	3.59
IEAP(Ours)	<b>0.908</b>	<u>0.056</u>	<u>0.794</u>	<b>3.42</b>	<b>4.48</b>	<u>4.46</u>	<b>4.12</b>

Table 17: Quantitative comparison results on AnyEdit Counting test set.

Method	CLIP <sub>im</sub> ↑	L1 ↓	DINO ↑	GPT <sub>IF</sub> ↑	GPT <sub>FC</sub> ↑	GPT <sub>AQ</sub> ↑	GPT <sub>avg</sub> ↑
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	<u>0.825</u>	<b>0.167</b>	<b>0.669</b>	2.82	<u>4.38</u>	<u>4.38</u>	3.86
ICEdit	0.806	0.171	0.629	<u>3.56</u>	4.16	4.06	<u>3.93</u>
IEAP(Ours)	<b>0.833</b>	<u>0.169</u>	<u>0.662</u>	<b>3.88</b>	<b>4.44</b>	<b>4.52</b>	<b>4.28</b>

Table 18: Quantitative comparison results on AnyEdit Implicit Change test set.

Method	CLIP <sub>im</sub> ↑	L1 ↓	DINO ↑	GPT <sub>IF</sub> ↑	GPT <sub>FC</sub> ↑	GPT <sub>AQ</sub> ↑	GPT <sub>avg</sub> ↑
InstructPix2Pix	0.887	0.111	0.858	<b>4.30</b>	<u>4.50</u>	4.30	<b>4.37</b>
MagicBrush	0.900	0.100	0.874	4.12	<u>4.36</u>	<b>4.54</b>	4.34
UltraEdit	<u>0.922</u>	<b>0.077</b>	<u>0.911</u>	3.24	4.4	4.36	4.00
ICEdit	0.898	<u>0.079</u>	0.864	4.16	4.46	4.20	4.27
IEAP(Ours)	<b>0.938</b>	0.084	<b>0.925</b>	<u>4.18</u>	<b>4.56</b>	<u>4.38</u>	<b>4.37</b>

Table 19: Quantitative comparison results on AnyEdit Move test set.

Method	CLIP <sub>im</sub> ↑	CLIP <sub>out</sub> ↑	L1 ↓	DINO ↑	GPT <sub>IF</sub> ↑	GPT <sub>FC</sub> ↑	GPT <sub>AQ</sub> ↑	GPT <sub>avg</sub> ↑
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	<u>4.48</u>	4.10	3.20
UltraEdit	0.732	0.279	<b>0.147</b>	<b>0.843</b>	1.96	4.46	3.98	3.47
ICEdit	<b>0.810</b>	<b>0.289</b>	<u>0.155</u>	<u>0.811</u>	<b>4.18</b>	4.42	<b>4.68</b>	<b>4.43</b>
IEAP(Ours)	<u>0.788</u>	<u>0.285</u>	0.162	0.786	<u>3.96</u>	<b>4.58</b>	<u>4.06</u>	<u>4.20</u>

Table 20: Quantitative comparison results on AnyEdit Textual Change test set.

## D More Visualization Results

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

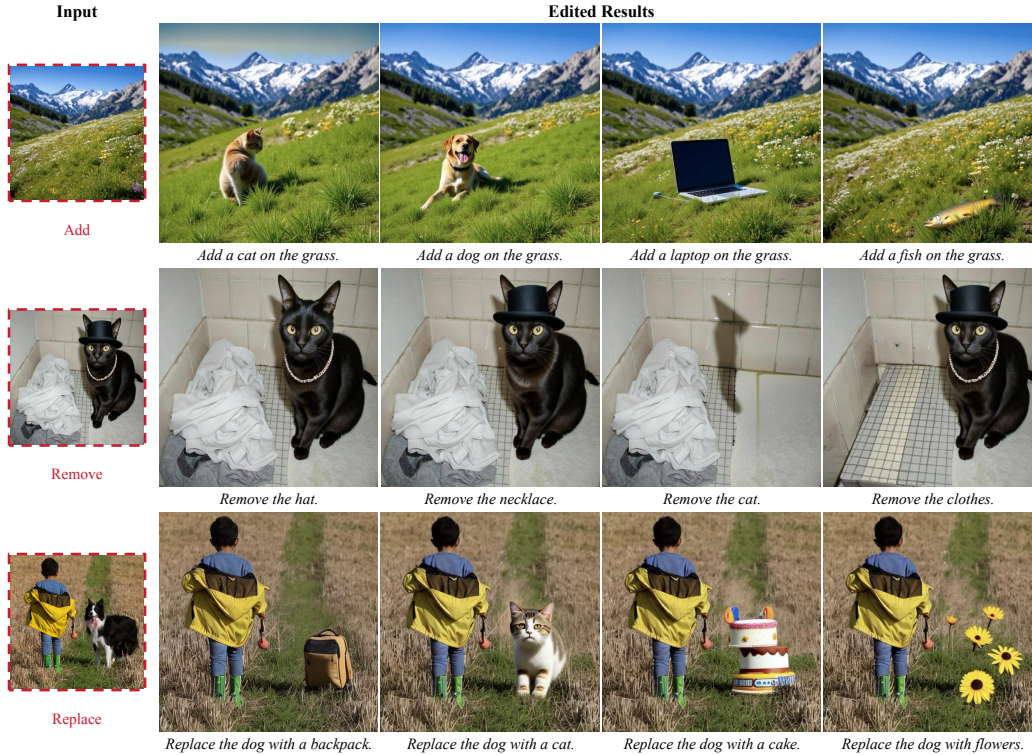


Figure 8: More Visualization Results.



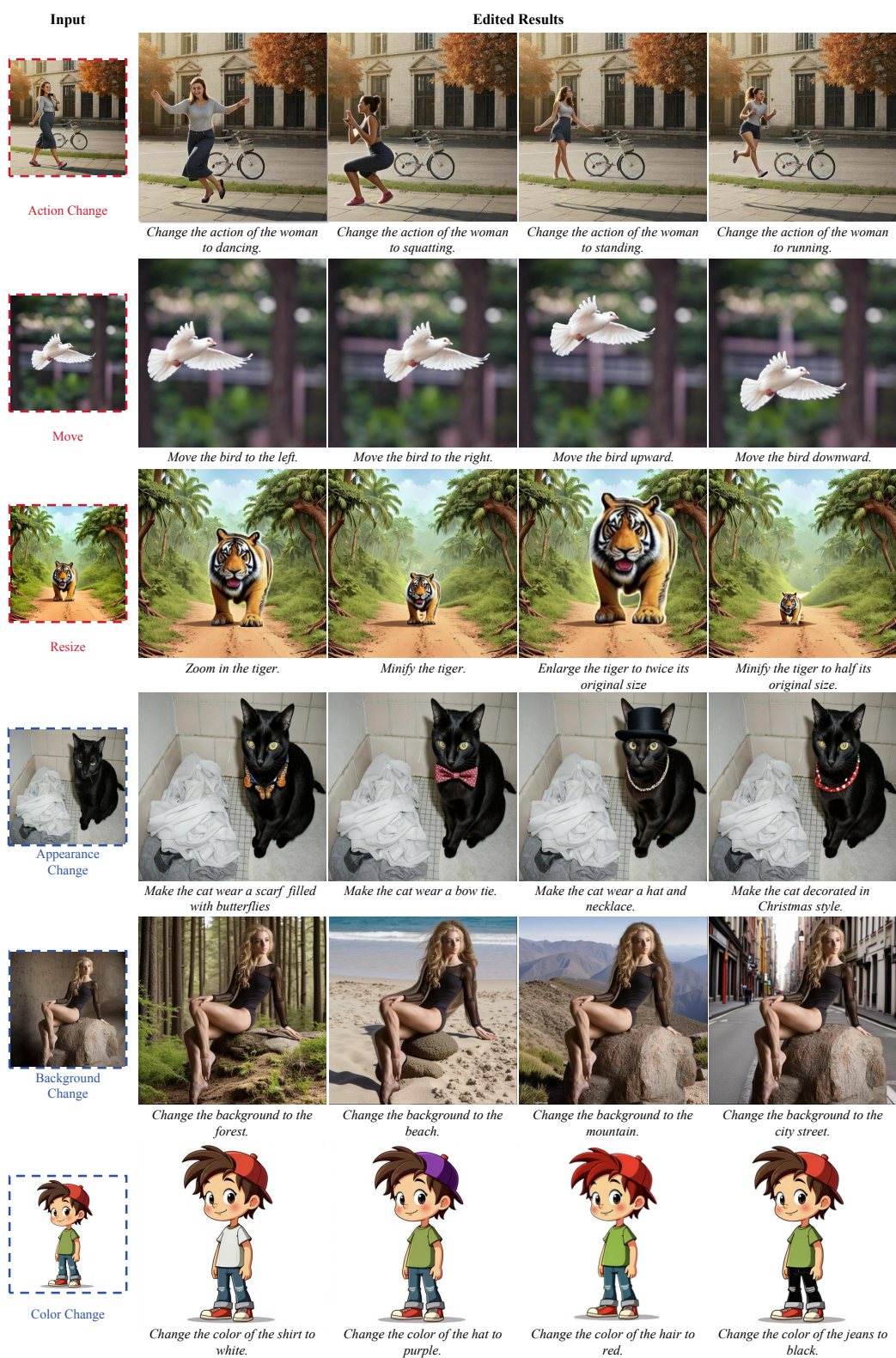


Figure 9: More Visualization Results.

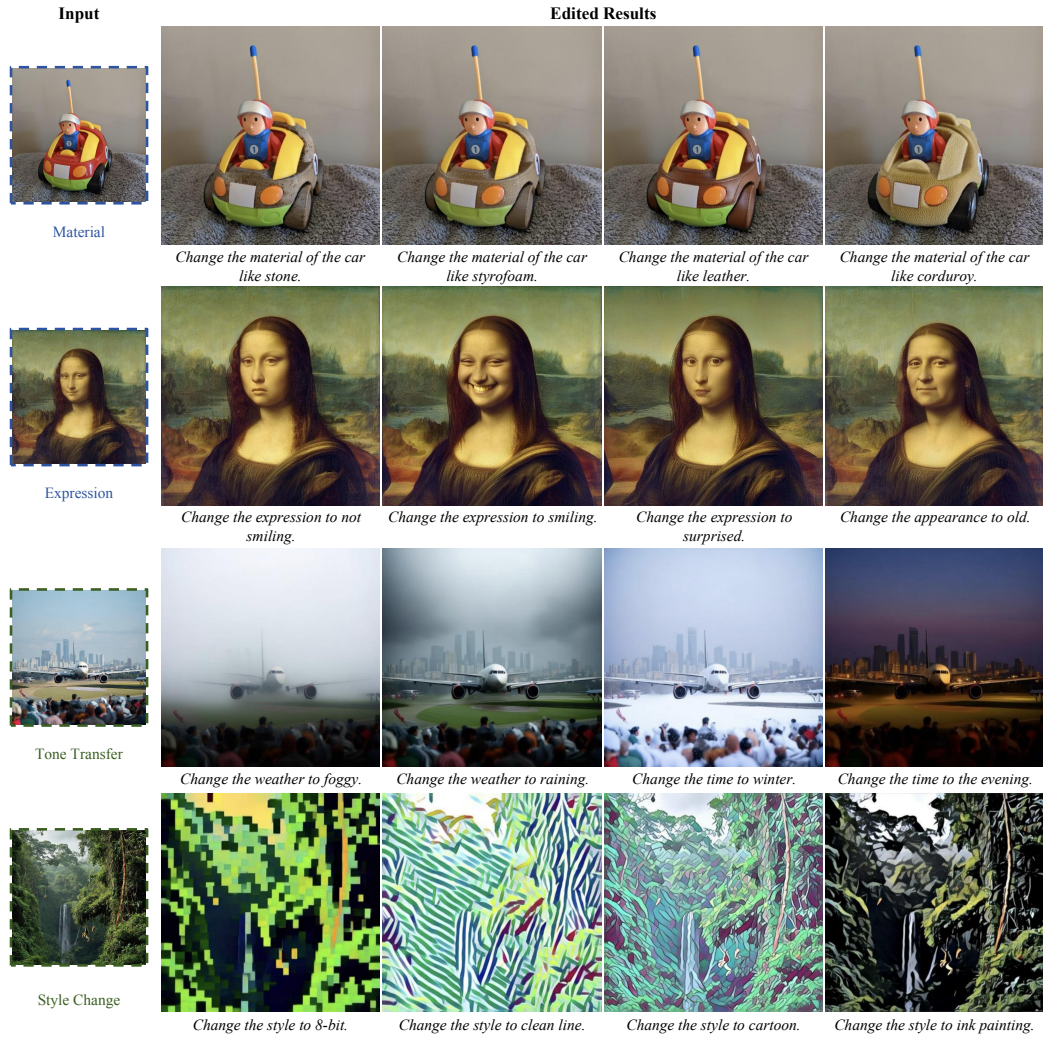


Figure 10: More Visualization Results.

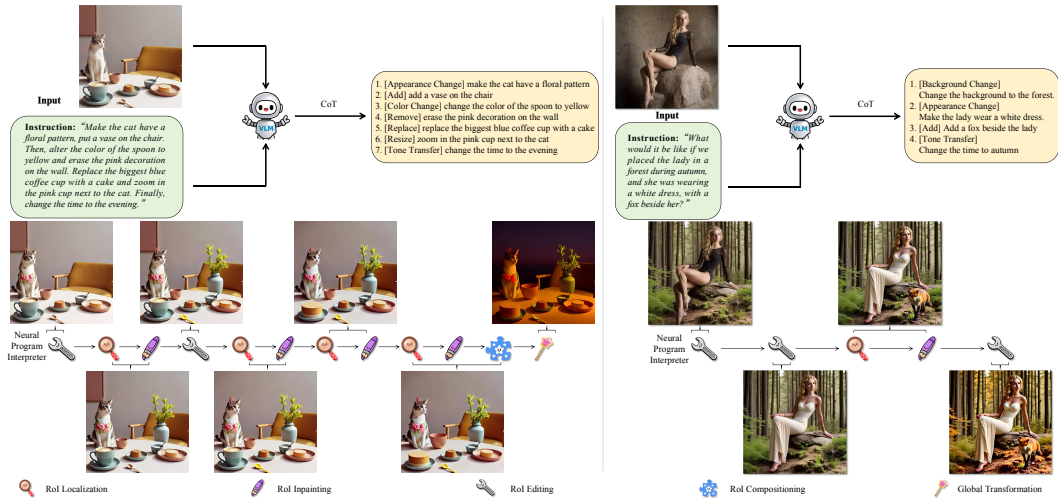


Figure 11: More Detailed Visualization Processes of the pipeline.



## E Analysis and Discussions

### E.1 Runtime Performance Analysis

We evaluate the time required for each atomic operation of IEAP on a single NVIDIA H100 GPU. Empirical measurements indicate that the RoI Localization stage requires approximately 3 s to 5 s per operation. Other editing primitives, including RoI Inpainting, RoI Editing, RoI Compositing, and Global Transformation, each consumes roughly 7 s to 9 s per operation.

Consequently, a complete multi-step edit involving  $k$  atomic operations exhibits a total latency of

$$T_{\text{total}} = \sum_{i=1}^k T_i \quad \text{with} \quad T_i = \begin{cases} 3 \text{ s to } 5 \text{ s}, & \text{if operation}_i = \text{RoI Localization,} \\ 7 \text{ s to } 9 \text{ s}, & \text{otherwise.} \end{cases}$$

While this per-operation cost precludes real-time interactivity, it remains acceptable for batch-oriented workflows in digital content creation, scientific visualization, and other offline editing scenarios.

### 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 s to 9 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,

549 and accessibility technology by generating descriptive visual representations for users with visual  
550 impairments, demonstrate the framework’s broader societal utility and interdisciplinary relevance.

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

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