Inter-Edit: First Benchmark for Interactive Instruction-Based Image Editing

Supplementary Material

A. Additional Details for Inter-Edit Benchmark

A.1. Automated Pipeline for Training Set Construction

This section provides a further elaboration on the three-stage automated data generation pipeline introduced in Section 3.1 of the main paper, as illustrated in A1.

Stage 1: Diverse Image Generation. To assist the large language model (LLM) in composing diverse prompts for source image generation, we sample keywords from an extensive, large-scale lexicon that is systematically categorized into three components. The first category, **Subject**, enumerates a broad range of objects that may appear in the image. The second category, **Style**, specifies the visual aesthetic of the desired output; importantly, to reflect the predominance of real-world editing scenarios, we assign a dominant weight (90%) to the *photorealistic* style, ensuring that the produced data maintains a high degree of visual realism. The third category, **Extra Elements**, provides supplementary descriptors that emulate natural photographic conditions, including ambient attributes (e.g., time of day, weather, season), photographic parameters (e.g., lens type, aperture, focal length), and compositional factors (e.g., viewpoint, shot angle).

The stochastic combination of keywords from these three categories yields 850,212 distinct prompt skeletons, which the LLM further enriches with creative elaboration, thereby substantially increasing the diversity of the generated source data. In addition, we supply a curated list of 100 high-quality, manually written example prompts. During prompt construction, three examples are randomly selected and presented to the LLM as in-context references, which improves both the quality and stability of its outputs.

Stage 2: Multi-Type Instruction Synthesis. In Stage 2, we guide the multimodal large language model (MLLM) to generate an edit instruction corresponding to one of four randomly selected edit types, which collectively cover a wide spectrum of common image manipulation tasks. The instruction length is sampled from either under 10 words or under 40 words, and the instruction is required to explicitly specify both the target location and the intended edit in order to improve editing accuracy. These prompts are then fed into Q-Edit [50] for high-quality image editing. This full-image generation approach significantly outperforms re-drawing methods in terms of image coherence and naturalness. However, we empirically observed that the success rate of the Edit model (following the generated instructions) was lower than the MLLM's ability to accurately describe an edit given the image pair. Therefore, to enhance annotation quality, we introduced a Regenerate step. After a successful edit, we retain only the resulting diptych (source and edited images). This diptych is fed back into the MLLM, which then re-infers the edit, determining the precise region where the change occurred and providing a shorter and more accurate instruction under the known edit region. The edit region coordinates are output as a bounding box `[x_min, y_min, x_max, y_max]`. If the MLLM fails to describe the edit region or provide a valid, non-zero bounding box, it must output `[0, 0, 0, 0]`. Any data sample associated with a `[0, 0, 0, 0]` bounding box is subsequently discarded, ensuring that all samples remaining have been both evaluated and successfully localized.

During data generation, we observe that the MLLM's localization ability degrades when processing images with excessively high resolution, while overly low-resolution images suffer from substantial information loss, negatively affecting the quality of both edit generation and evaluation. To balance these factors, we introduce a standardized preprocessing step before all MLLM interactions—including prompt generation in Stage 2, subsequent edit localization and instruction regeneration, and evaluation in Stage 3. Specifically, each image is resized such that both dimensions are at most 960 pixels while preserving its original aspect ratio. This constraint maintains sufficient visual detail for analysis while enhancing the MLLM's spatial understanding and localization accuracy.

During the regeneration stage, the newly generated fine-grained instruction is additionally required to roughly match the original instruction length so as to preserve the proportional structure established in Stage 1. By contrast, the concise instruction—given that the edited region is already known—is only required to describe the content of the edit itself.

Stage 3: Data Filtering and Refining

For the filtering stage, we employ an MLLM-as-a-judge to decide whether the performed edit is successful. To minimize false positives (i.e., edits incorrectly labeled as successful), we enforce a strict output protocol. When the MLLM evaluator provides its Chain-of-Thought analysis and deems an edit successful (a `Success` evaluation), it is required to also output the coordinates of the edited region. If the edit is deemed unsuccessful (`Fail`, as illustrated in A2) or the output edited region fails to align with the output in Stage 2, the image pair is discarded.

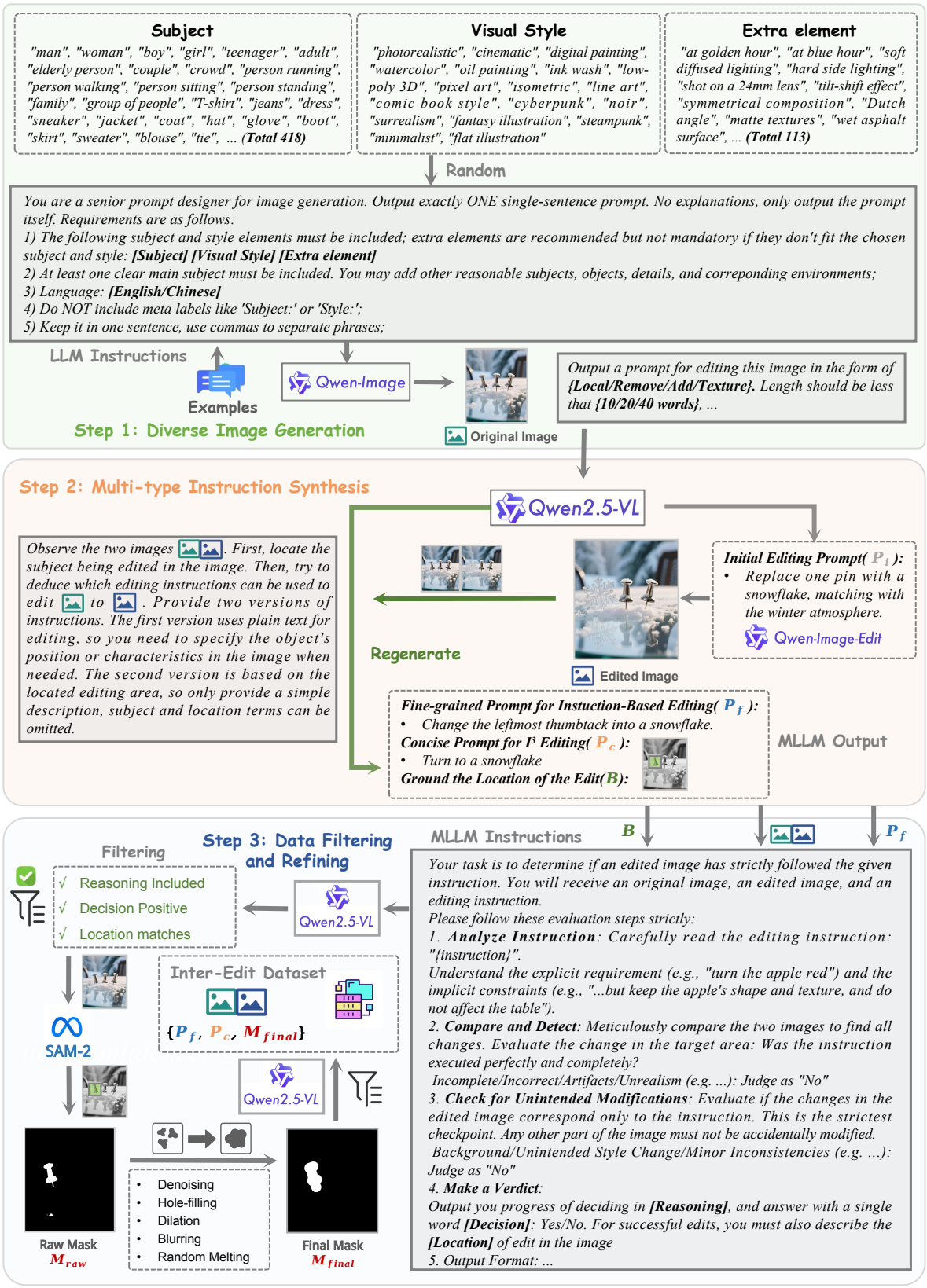

Figure A1. Detailed pipeline of training set construction

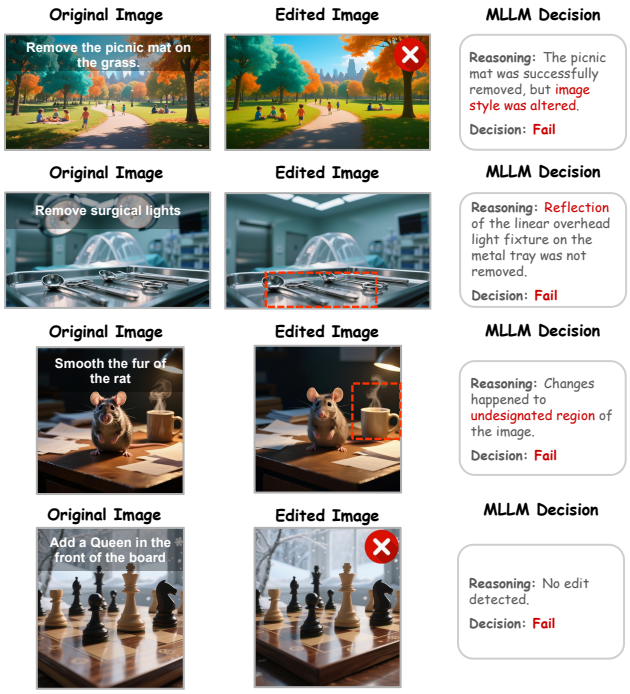

Figure A2. Examples from the entries filtered out by the MLLM. Red crosses mark editing failures, and red boxes highlight imperfect regions. Reasoning of the MLLM are shortened appropriately to be shown in the figure.

Utilizing the spatial prior provided by MLLM, Segment Anything 2 (SAM-2) [18] is first applied to segment the subject of edit. As noted in the main paper, the raw, pixel-level masks M_{raw} generated by SAM-2 are often fragmented and do not align with human intuition for specifying an edit region. We developed a multi-step post-processing pipeline to transform M_{raw} into a smooth, naturalistic mask. The process begins by applying a morphological **opening** operation (with a 5×5 kernel for 2 iterations) to eliminate small, spurious pixel noise. Following this, we use a binary **hole-filling** algorithm to ensure the main segmented region is solid. We then apply a morphological **closing** operation (using a larger 9×9 kernel for 2 iterations) to bridge small gaps and consolidate the mask. To simulate a more generous, human-drawn mask, we perform a large-scale **dilation** using a 101×101 elliptical structuring element. The sharp, aliased edges of this dilated mask are then smoothed by a strong 71×71 Gaussian **blur**, and the result is **binarized** (threshold at 127) to produce an intermediate smooth dilated mask, M_{final}.

A.2. Manually Annotated Inter-Edit Test Set

In this section, we first present a comprehensive overview of the complete process for constructing the Inter-Edit test set in Section A.2.1. Following that, Section A.2.2 provides a detailed description of the browser-based annotation tool we meticulously designed to streamline the annotation process for annotators.

A.2.1. Data Annotation Process

The annotation process for the test set is illustrated in Figure 2(b). The entire workflow involves three key steps: first, the extensive collection of source images; second, the generation of edited images to form image pairs; and finally, the manual inspection and annotation of the complete I^3E data pairs.

Source Image Collection. To ensure that the images align closely with those edited by real users, we select the main source images for the test set to be real-world images, with an annotation goal of 5,000 pairs. To guarantee the successful generation of the desired number of I^3E data pairs, we randomly sample 50,000 real images from the LAION dataset [42] as the primary set of source images. Additionally, to test the model's ability to handle challenging examples, we design several difficult subsets from the source image collection. These subsets primarily consist of two special real image subsets: (1) low-resolution images ($\leq 480px$), randomly sampled from the LAION dataset, and (2) images with low aesthetic scores, specifically images with an average aesthetic score below 4, sampled from the AVA dataset [34]. Furthermore, we create two special subsets of generated images: (1) source images in various artistic styles, and (2) source images containing multiple identical objects. For the generated image subsets, the prompts are generated by Qwen3-32B [52], tailored to the characteristics of each subset, while the images are produced by Qwen-Image [50] based on the generated prompts and random image sizes. The size rules align with the training set, covering various dimensions. Each subset collects 2,500 images, with the annotation goal set at 250 pairs for each subset.

Generation of Edited Images. After collecting the source images, we adopt a procedure similar to that used in constructing the training set to generate edited images. Specifically, fine-grained edit instructions are first generated based on the visual content of each source image and a randomly sampled editing type, improving the likelihood of successful edits. Each instruction, together with its corresponding source image, is then fed into Q-Edit [50] to produce the edited results. The generated data are subsequently filtered using Qwen2.5VL-72B [2], following the same filtering strategy as in the training stage, except that mask-related components are excluded from the input. Only the samples that pass this filtering step proceed to the manual annotation phase.

Manual Annotation Process. Upon obtaining the triplet data consisting of source images, fine-grained edit instructions, and edited images, the manual annotation process begins. Annotators use a carefully designed front-end inter-

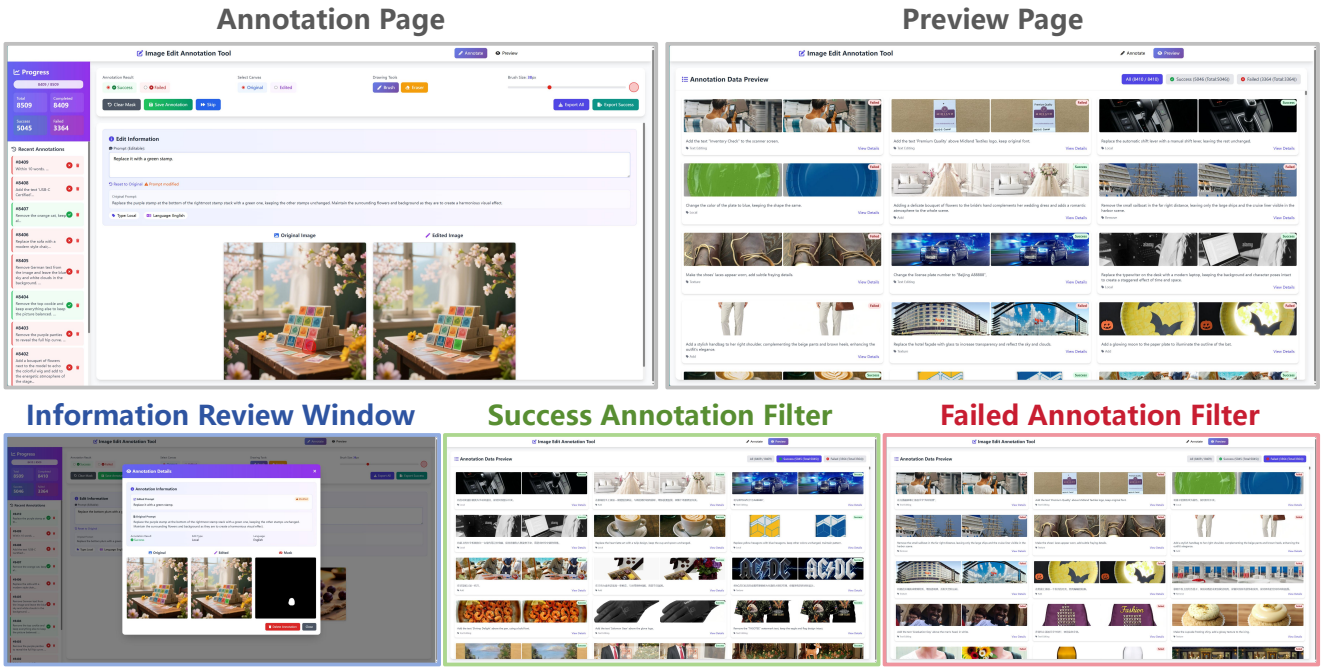

Figure A3. Screenshot of the front-end demonstration of the annotation tool. It mainly includes two functional pages: the annotation page and the preview page. On the annotation page, clicking on a recently annotated data item allows users to review the corresponding detailed annotation information. On the preview page, clicking on the corresponding data item opens the same detail window. At the top of the preview page, users can filter and view data labeled as successful/failed separately.

face to perform the annotation. The annotation process follows these steps: first, annotators assess whether the edited result aligns with the expected outcome. After reviewing the edited image and understanding its intent, and confirming that the data in any of the triplet elements are accurate, the annotator intuitively draws a mask on the source image. They then refine the edit instruction based on the known mask. Multiple annotators participate in this process, with each annotator assigned the same number of annotation targets, ensuring that the data within each subset is evenly distributed. This guarantees that the annotation results reflect a broad user perspective, minimizing biases in the annotations.

A.2.2. Annotation Tool

We have developed a lightweight, browser-based annotation interface for curating and quality-controlling the *Inter-Edit* human-annotated test set. The tool standardizes how annotators inspect paired images (original vs. edited), mark spatial regions with coarse masks, refine the edit instructions based on the images and mask content, and record binary judgments (`Success`/`Failed`). It can be run locally or on a shared server with a single command, automatically saving progress and providing comprehensive preview and correction functionalities to ensure consistency in annotation. The front-end screenshot of the annotation tool is shown in Figure A3, with further details provided in the following paragraph.

Interface Overview. The top navigation bar provides two main pages: *Annotate* and *Preview*. The *Annotate* page serves as the main workspace, displaying the original and edited images side by side, with the following components: (i) a canvas selector (`Original`/`Edited`), (ii) drawing tools (`Brush`/`Eraser`) and adjustable brush size, (iii) a mandatory binary judgment selector, and (iv) an instruction panel with editable prompt and metadata (edit type, language). The left sidebar displays real-time statistics—`Total`, `Completed`, `Success`, and `Failed`—as well as a dynamic list of recent annotations. Each recent entry includes its prompt text and result tag (`Success` or `Failed`), allowing annotators to quickly review or remove erroneous records via a one-click delete icon.

Annotation and Review Workflow. For each image pair, annotators (1) read the instruction and examine both images, verifying whether the original fine-grained annotation instruction is correct. If the entire triplet is correct and a mask annotation is required, the binary judgment is selected as `Success`; otherwise, it is marked as `Failed`. Only the editing pairs marked as successful will proceed to the next annotation step; (2) draw a coarse, intuitive mask on the original or edited image; (3) input a refined textual instruction; and (4) review the annotation results and confirm their accuracy. The toolbar provides three key actions: `Clear`

Figure A4. Examples from the Inter-Edit test set. Entries are chosen from the designed challenging subset.

`Mask` (reset the canvas), `Save Annotation` (store and move to the next sample), and `Skip` (defer uncertain samples). Saved annotations immediately update the progress bar and recent-history sidebar, enabling annotators to verify their submission without leaving the page.

Preview and Management Page. The *Preview* page offers global visibility into all labeled data. It supports category filters (`All`, `Success`, `Failed`) and displays annotations as compact cards containing both images, the corresponding prompt, and a color-coded status label. Clicking any card opens a detailed dialog showing three aligned images—the original, edited, and annotated mask—along with metadata such as fine-grained/refined prompt, binary judgment result, edit type, and language. This unified view allows annotators or supervisors to inspect label quality and confirm spatial accuracy. If an annotation is found to be incorrect, users can delete it directly from either the recent-history sidebar or the detailed preview window; the item is then returned to the pending queue for re-labeling, ensuring dataset integrity.

Configuration and Deployment. All behaviors are governed by a single configuration file, which specifies data paths, mask storage directories, server host/port, preview limits (to reduce transmission and browser load), etc. Configuration changes take effect upon restart. The interface is browser-compatible and optimized for both individual and collaborative use. In addition to preview limits, we also restrict the number of annotation tasks forwarded from the server to the frontend, defaulting to 100. That is, each time the browser refreshes, a new set of 100 tasks will be automatically assigned based on the target IP address. Annotations that annotators delete due to errors will be prioritized for re-labeling. This also reduces transmission and browser load, ensuring smooth annotation and avoiding multiple annotators working on the same data simultaneously.

I^3E Design Philosophy. This tool encourages annotators to intuitively draw *imprecise but natural* masks to simulate real user interaction in I^3E, promoting faster annota-

Table A1. Distribution and description of edit categories in the Inter-Edit training set.

Category	Percentage	Description
Local	37.1%	Involves substituting one object for another or altering an object's attributes (e.g., "change the man's shirt from blue to red").
Add	28.4%	Inserts a new object into the scene (e.g., "add a pair of sunglasses to her face").
Remove	28.0%	Entails erasing an object from the image (e.g., "remove the person on the left").
Texture	6.5%	Alters an object's visual characteristics, such as material or pattern, without affecting its underlying structure (e.g., "cover the wall with brick texture").

tion and better reflecting practical spatial intent. Editable instructions ensure that the masks and precise instructions align with human editing habits while preserving semantic fidelity. The combined functionality—masking, judging, reviewing, and correcting—supports reliable, high-quality test set curation, aligned with our evaluation framework.

Benchmark Outputs. Each record is automatically saved as a structured JSON entry paired with the corresponding mask image. Annotators can export all labeled data or only `Success` samples for benchmark preparation. The exported artifacts—paired images, fine-grained/refined prompts, masks, and validated outcomes—constitute the ground-truth supervision for the *Inter-Edit* test benchmark.

A.3. Additional Information of the Dataset

A.3.1. Details of the Inter-Edit Training Set

The final Inter-Edit training set comprises 1,099,964 image-editing pairs. The distribution of these pairs across the four primary editing categories is as follows: Local (37.1%), Add (28.4%), Remove (28.0%), and Texture (6.5%). A detailed breakdown of each category, including its definition and distribution, is presented in Table A1.

In Figure A5, we visualize the most frequent meaningful keywords within the editing instructions for each category. Notably, due to differences in data sources and characteristics between the training and test sets, the high-frequency keywords are not identical across the two splits.

The masks generated for the editing regions cover an average of 14.79% of the total image area. Due to a series of morphological operations applied during generation, masks for very small objects are expanded; consequently, only 25% of the masks cover less than 5% of the image. Furthermore, 90% of all masks are smaller than 28.55% of the image area. This distribution indicates that our generation pipeline successfully avoids creating oversized edit regions (e.g., those occupying half the image or more), validating that positional constraints in such scenarios are both feasible and meaningful.

The initial image editing prompts generated by the MLLM [2] have an average length of 17 words. Following a "Regenerate" process, this distribution shifted: approximately 34% of prompts are 10 words or fewer, while 25% exceed 37 words. This suggests the MLLM provides detailed descriptions, though these lengths may be verbose for typical user interaction. To better suit the I^3E task, we employed the MLLM again to condense the prompts based on the image pair content. After this refinement, the average instruction length was reduced to 8 words, which we believe significantly enhances the usability of the I^3E model.

A.3.2. Details of the Inter-Edit Test Set

The Inter-Edit test set contains 6,250 image pairs, with corresponding editing instructions provided in both Chinese and English. For all evaluations in the main paper, we use only the English instructions to ensure a fair comparison for models that do not support Chinese. Furthermore, recognizing that model capabilities for "text editing" (e.g., changing text on a sign) vary significantly, we isolated this subcategory from the broader **Local** edit type. This allows for more granular and fair model-specific evaluations.

Analysis of the manually annotated masks in the test set shows that the intended edit region covers, on average, 13.24% of the total image area. This size is consistent with typical localization-dependent editing scenarios, demonstrating that our test set generation method aligns well with real-world use cases.

To increase the evaluation challenge, we introduced several challenging subsets into the test set, as illustrated in A4:

- **Artistic Styles:** We used a T2I model [50] with controlled prompts to generate edits on images with diverse artistic styles.
- **Low-Resolution Images:** Sourced from the LAION dataset, these images are ($\leq$ 480px) and have undergone post-processing.
- **Low Aesthetic Scores:** We included images from the AVA dataset [34], a professional photography aesthetics dataset where each work is rated by multiple human critics. We filtered for images with an average score of 4.0 or lower (on a 1-10 scale), which constitute approximately 3.13% of the data.
- **Ambiguous Edits:** We generated scenarios with inherent ambiguity (e.g., images containing multiple instances of the same target object, such as "the person on the left") by controlling prompts in a T2I model [50].

A.4. Additional Information of Evaluation Metrics

A critical preprocessing step ensures all calculations are performed on a consistent resolution. For each sample, the

Table A2. Comparison of Open-Source Image Editing Datasets

Property	InstructPix2Pix	MagicBrush	HQ-Edit	UltraEdit	OmniEdit	ImgEdit	Inter-Edit (Ours)
Size	450K	10K	200K	4M	1.2M	1.2M	1.1M
Mask	No	Yes	No	Yes	No	Yes	**Yes (Refined)**
High Res	No	Yes	Yes	No	Yes	Yes	**Yes**
Edit Types	General	General	General	Defined	Defined	Defined	**Defined**
Bounding Box	No	No	No	No	No	Yes	**Yes**
Dual Prompts	No	No	No	No	No	No	**Yes**

Table A3. Comparison of Image Editing Benchmarks (Test Sets)

Property	InstructPix2Pix	MagicBrush	EditBench	Emu Edit Bench	Inter-Edit (Ours)
Test Set Size	*N/A*	1,053	240	3,589	**6,250**
Source	AI-Generated	Human-Annotated	Mixed	Mixed	**Human-Annotated**
Mask-Based	No	Yes	Yes (Inpainting)	No	**Yes**
High Res	No	Yes	Yes	Yes	**Yes**
Specialized Subset	No	No	No	No	**Yes**

Figure A5. Keywords frequency statistics in the instructions of Inter-Edit training set (Left) and Benchmark (Right) across various instruction types. For better illustration, the relative proportions in the image are accurate, but the other keywords have been merged and reduced in size to avoid taking up excessive space.

source image (I_s), edited image (I_e), ground truth image (I_{gt}), and mask (M) are all resized to a uniform base size, set to 512×512 in our experiments.

Image resizing for I_s, I_e, and I_{gt} utilizes LANCZOS in-

```
Your task is to serve as a rigorous, objective judge of an image editing task. You will be
provided with four inputs: 1) the Original Image, 2) the Editing Instruction (text), 3) the
Editing Area (Mask) and 4) the Edited Image. You must analyze the result and provide a
strict evaluation based on four distinct criteria. For each criterion, you must assign an
integer score from 1 to 10.

Core Scoring Directive: Use the Full Scoring Range
This is a critical instruction. You must utilize the entire 1-10 scoring range and avoid
clustering your scores (e.g., only giving 7s, 8s, and 9s). A score of 1 represents a
complete and total failure in that category. A score of 10 represents flawless, indisputable
 perfection. Use the intermediate scores (2-9) to reflect the full spectrum of quality. A
mediocre or average result should be scored around a 5 or 6, not a 7 or 8. Your scores must
be thoughtfully distributed.

Evaluation Criteria:
You must score the following four criteria from 1 to 10.
1. Edit Success (1-10)
Definition: How successfully was the edit technically executed?
10 (Perfect): The edit is perfectly and accurately completed. The target area is edited
cleanly, and there are no unreasonable alterations, artifacts, or halos in other parts of
the image. 1 (Failure): The edit is fundamentally incorrect, failed to execute, or no
meaningful edit was performed. Intermediate: Deduct points based on the severity of
technical flaws, such as residual artifacts, unnatural blending, or incomplete edits.
2. Instruction Alignment (1-10)
Definition: How well does the edited image follow the user's text instruction and Mask Area?
10 (Perfect): The edit perfectly and completely fulfills all aspects and nuances of the
instruction. 1 (Failure): The edit completely disregards, misinterprets, or acts contrary to
 the instruction. Intermediate: Score based on the degree of alignment. For example, if the
instruction is "make the apple red and shiny" and the edit only makes it red but not shiny,
the score must be lowered accordingly.
3. Naturalness (1-10)
Definition: How realistic and seamlessly integrated is the edit?
10 (Perfect): The edited region is indistinguishable from the original image. There are no
stylistic discrepancies. Lighting, shadows, textures, and reflections are perfectly
consistent and physically realistic. All details are seamlessly blended, and the edit is
impossible to detect. 1 (Failure): The edit is clumsy, jarring, and obviously artificial.
The edited portion is immediately and easily identifiable as a "photoshop." Intermediate:
Deduct points for any unnatural elements, such as inconsistent lighting, mismatched styles
or textures, or visible seams/edges.
4. Aesthetics (1-10)
Definition: How visually appealing is the result of the edit?
10 (Perfect): The edit is highly aesthetic, visually pleasing, and enhances the image's
overall appeal (while still respecting the instruction and the original image's style). 1 (
Failure): The edit is ugly, jarring, or visually unpleasant. Intermediate: Score based on
the overall beauty and visual harmony of the edited result.

Output Format:
Provide your evaluation in the following structured format. First, provide a concise
rationale for each of your four scores. Then, list the scores.
Rationale
Edit Success: [Your reasoning for the score.]
...
Final Scores
Edit Success: [1-10]
...
```

Figure A6. Prompt for Editing Evaluation

terpolation to maintain image detail. In contrast, the binary mask M is resized using NEAREST interpolation to ensure its boundaries remain discrete and are not blurred.

Global Similarity ($\mathcal{S}_{global}$) We compute $\mathcal{S}_{global}$ using the LPIPS distance [60]. Consistent with standard evaluation practices, we employ the AlexNet [19] backbone for the LPIPS model. Prior to being fed into the network, the edited image I_e and ground truth I_{gt} are converted to tensors and normalized to the range $[-1, 1]$ (using a mean and standard deviation of 0.5). $\mathcal{S}_{global}$ is the resulting distance, where lower values indicate better global similarity.

Regional Similarity ($\mathcal{S}_{in}$ and $\mathcal{S}_{out}$) For the regional metrics, we utilize the official Alphaclip implementation [45] with a ViT-B/16 visual backbone. The images (I_s, I_e, I_{gt}) are processed using the standard CLIP [39] image preprocessor.

The alpha masks, M and $(1 - M)$, are processed according to the specific requirements of the Alpha-CLIP visual encoder. The binary masks (with values $\{0, 1\}$) are first rescaled to $\{0, 255\}$. They are then resized to 224×224, converted to tensors, and normalized using a mean of $\mu = 0.5$ and a standard deviation of $\sigma = 0.26$. The final feature vectors $E_\alpha(\cdot, \cdot)$ are extracted from the model's `visual` encoder. $\mathcal{S}_{in}$ and $\mathcal{S}_{out}$ are then computed as the cosine simularity between the corresponding feature pairs, as shown in Equations (2) and (3).

Boundary Discontinuity Score (BDS) The BDS is calculated using the images resized to 512×512. The process is as follows:

1. The edited image I_e is converted to a single-channel grayscale image.
2. A gradient magnitude map, $\mathcal{G}(I_e)$, is computed using Sobel filter with a 3×3 kernel. The final magnitude is calculated as:

$$\mathcal{G}(I_e) = \sqrt{(\text{Sobel}_x(I_e))^2 + (\text{Sobel}_y(I_e))^2} \quad (6)$$

3. The inner (T_{in}) and outer (T_{out}) transition bands are generated using morphological `erode` and `dilate` functions, as defined in Equation (4). A key implementation parameter is the kernel k; we use a square 11×11 kernel for these operations.
4. The final BDS is computed as the absolute difference between the mean gradient magnitude within T_{in} and T_{out}, following Equation (5) (where I_{out} is I_e). A score of 0.0 is returned if either transition band is empty (e.g., if the mask is too small or covers the entire image).

VQA Scores we employ a powerful MLLM [1] for automated assessment. We formulate a validation prompt that assess editing performance in four aspects: Edit Success, Instruction Alignment, Naturalness, and Aesthetics. Detailed prompt is given in A6

B. Additional Details for Experiments

All data synthesis and baseline model training are conducted on eight H800 GPUs.

Data Synthesis Details. During the construction of the *Inter-Edit* dataset, Qwen3-32B [52] serves as the large language model (LLM) with a batch size of 12 and no reasoning mode enabled. On average, generating one prompt for a source image takes 0.77 s per GPU. Qwen2.5VL-72B [2] is employed as the multimodal large language model (MLLM) and performs the most steps in the pipeline, each with a batch size of 4. First, the MLLM determines an initial editing instruction based on the source image, which requires 1.24 s per GPU. After obtaining the edited image, the model performs two tasks—relabeling instructions with varying lengths and localizing the edited region using a bounding box—each taking 2.36 s per GPU. In the final filtering stage, the MLLM evaluates all information within each data pair and outputs both the decision and its reasoning to enhance precision, which is the most time-consuming step, averaging 3.65 s per GPU.

For image generation, we employ Qwen-Image [50] as the source image generator and Q-Edit [50] as the image editing model, both recognized as the most powerful open-source models in their respective domains. To preserve visual fidelity, we apply only *flash attention* [8] for acceleration, avoiding techniques such as step distillation [10, 32] that may degrade image quality. During source image generation, the model receives a synthesized prompt and randomly samples one of nine aspect ratios (16:9, 3:2, 4:3, 1:1, 3:4, 2:3, 9:16). The resolution is set to approximately 1328×1328 pixels, matching the model's optimal setting. Each image is generated with 50 denoising steps using default parameters, requiring 28.75 s per GPU on average. For editing, Q-Edit takes the source image and textual instruction as input, maintaining the same aspect ratio and using a resolution around 1024×1024 pixels. Each edited image is produced through 40 denoising steps with default parameters, taking an average of 31.16 s per GPU. Although constructing the million-scale training set is a lengthy process, it only needs to be done once. All synthesized data will be publicly released, eliminating the need for repetition.

Model Training Details. For all baseline methods, Q-Edit serves as the pretrained backbone. The training setup is consistent across methods: the batch size per GPU is set to 1 with 4-step gradient accumulation. We use the Prodigy optimizer [33] with warm-up and bias correction enabled, a weight decay of 0.01, and a total of 50,000 training steps. The input source image and mask are resized to maintain the original aspect ratio while ensuring a total pixel count

Table A4. Quantitative results of RNI, CIA, and CJT on the full test set, as well as the Chinese and English subsets.

	Method	Global & Boundary		Regional Fidelity		VQA Scores					Human
		LPIPS ↓	BDS ↓	$\mathcal{S}_{in}$ ↑	$\mathcal{S}_{out}$ ↑	S_{edit} ↑	S_{nat} ↑	S_{aes} ↑	S_{align} ↑	S_{VQA} ↑	Eval. ↑
Full	RNI (Full)	0.193	9.545	0.973	0.977	6.499	5.789	6.203	7.145	6.409	6.619
	CIA (Full)	0.260	5.219	0.964	0.954	5.583	5.573	5.941	6.502	5.956	6.129
	CJT (Full)	0.247	5.129	0.973	0.963	6.506	5.835	6.198	6.886	6.356	6.672
Chinese	RNI (Chinese)	0.195	8.605	0.970	0.979	6.456	5.792	6.181	7.416	6.461	6.461
	CIA (Chinese)	0.261	4.903	0.961	0.958	5.782	5.588	5.929	6.429	5.932	6.102
	CJT (Chinese)	0.251	4.822	0.970	0.964	6.673	5.837	6.157	6.848	6.379	6.624
English	RNI (English)	0.191	10.485	0.976	0.974	6.541	5.785	6.224	7.174	6.431	6.672
	CIA (English)	0.259	5.534	0.966	0.950	5.384	5.557	5.952	6.574	5.979	6.156
	CJT (English)	0.242	5.435	0.976	0.961	6.338	5.833	6.239	6.923	6.333	6.720

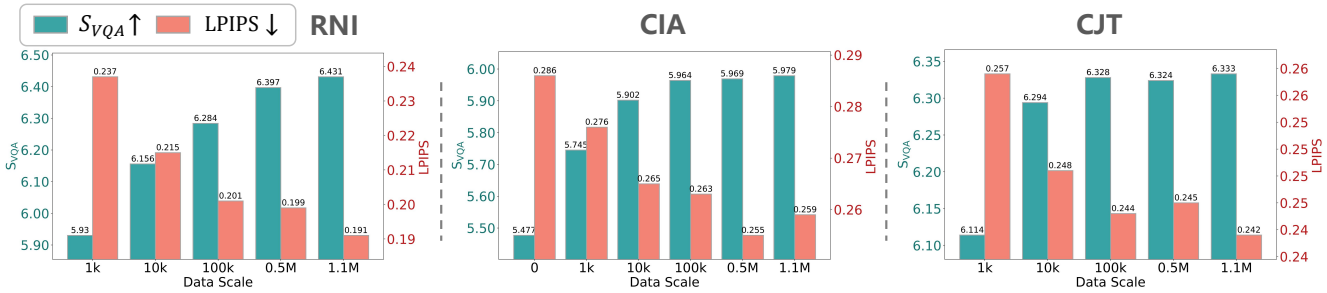

Figure A7. Performance trends of the three proposed methods under different training data scales. RNI continues to benefit from larger datasets, while CIA and CJT reach saturation with comparatively smaller amounts of data.

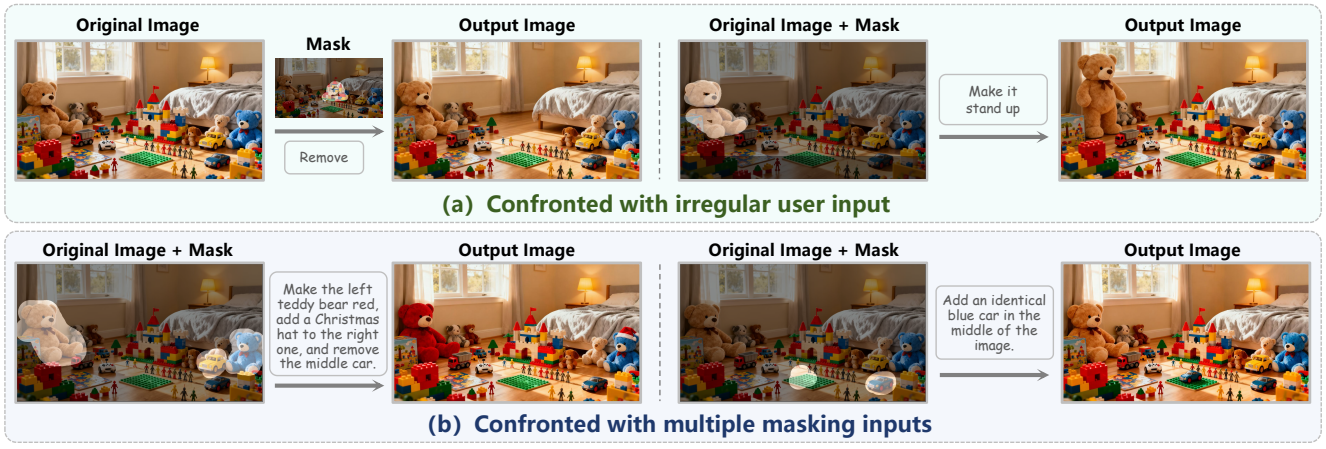

Figure A8. Performance of the proposed baseline CJT method on the I^3E task when encountering atypical user-scribbled masks. (a) Results obtained when the input mask contains irregular strokes and discontinuities. (b) Results when the user provides multiple disjoint masked regions for editing.

close to 1024×1024.

For the RNI method, the ControlNet [59] module consists of six double blocks copied from pretrained Transformer layers, and the mask is fed into the model at the same resolution as the source image. In the CIA method, the mask boundary is extracted and overlaid onto the source image as a red contour line with a pixel width of 3. Both CIA and CJT integrate LoRA modules into all linear layers of the attention blocks within the Diffusion Transformer of Q-Edit, with the LoRA rank set to 32. During testing on *Inter-Edit*, the mask and source image are fed into the model following the same resolution configuration as in training. For other comparison methods, inference is conducted at their respective recommended resolutions.

C. Additional Results

C.1. Additional Quantitative Results

Complete Metrics for Baseline Methods. Since most existing methods in related areas support only English outputs, we report performance exclusively on the English subset of the Inter-Edit dataset in Table 1 of the main paper to ensure fair comparison. However, all three of our proposed methods natively support both Chinese and English. To facilitate future bilingual research and enable fairer evaluations, we provide in Table A4 the full performance of our methods on the entire Inter-Edit dataset as well as on its Chinese and English subsets. The overall results remain stable across languages, with English inputs exhibiting slightly better performance than Chinese.

Effect of Training Scale on Model Performance. Figure A7 illustrates the performance variations of our three designed methods under different training data scales. All methods exhibit a consistent trend—larger training datasets generally lead to better performance. However, their responses to data scaling differ. For the RNI method, performance steadily improves as the training data increase, showing a continued upward trend even at the million-scale level, suggesting potential for further gains with additional data. In contrast, the CIA and CJT methods reach near-peak performance with relatively small amounts of training data, and their improvements saturate as the dataset expands. In practice, one can flexibly select the appropriate method according to the available amount of training data.

C.2. Additional Qualitative Results

Robustness to Irregular and Complex Mask Inputs. In practical interactive editing scenarios, user-drawn masks often contain irregular strokes, discontinuities, or multiple spatially separated regions corresponding to different editing requests. To assess robustness under such realistic conditions, we further evaluate the proposed CJT method on these challenging inputs. As illustrated in Fig. A8, our method demonstrates strong inferential capability when confronted with unconventional masks—such as hollow or fragmented scribbles—and is able to produce editing results that align well with user intent. Surprisingly, despite the absence of explicit training for handling multiple interactive editing requirements simultaneously, the model is still capable of performing such multi-region edits in a zero-shot manner, yielding outputs that meet expectations.

Additional Applications. Based on the powerful editing capabilities of Q-Edit [50], out model further improved on precisely locate and complete more complex image editing tasks, while consistently maintaining visual and semantic consistency in unedited regions. As shown in Figure A9, our approach facilitates a diverse array of editing tasks, such as eliminating background crowds and modifying subject attributes in daily photos. It excels in complex scenarios containing multiple subjects, enabling easy localization and editing while strictly confining changes to the target region. Maintaining the advantages of the original model, our method enables the generation of style-consistent text within sketched boundaries. These capabilities can be chained in multi-turn workflows to accommodate elaborate editing needs.

D. Limitations and Future Work

Limitations. Although the proposed pipeline achieves strong performance, it still exhibits several limitations. First, the training data are primarily automatically generated through cascaded models, and although we incorporate a filtering stage to ensure high-quality samples, this process may still introduce the intrinsic preferences and biases of the underlying models into the dataset. Second, the pseudo–hand-drawn masks obtained via semantic segmentation and morphological post-processing only approximate coarse user annotations; real scribbles often contain irregular strokes, discontinuities, and diverse drawing habits that are not fully captured by these simulated masks. Finally, while the method itself is capable of handling iterative modifications, the benchmark does not include evaluation protocols or metrics for multi-turn editing, leaving the performance of repeated interactive edits insufficiently assessed. These limitations suggest promising directions for future work on data realism, scribble modeling, and evaluation design.

Future Work. Looking forward, the I^3E paradigm offers several compelling avenues for further exploration. Figure A10 illustrates three feasible directions. A natural extension is to incorporate controllable regional guidance intensity, enabling users to adjust how strongly the model adheres to the spatially specified region. This provides a more fine-grained and expressive editing experience while remaining highly user-friendly. For instance, users may increase the guidance strength when edits must remain strictly confined within the scribbled area, or decrease it when they prefer the modification to naturally propagate to surrounding similar elements—all without switching to a different model. Another promising direction is the integration of controllable visual elements, such as explicitly specified objects to be added, which can guide the model toward generating more predictable and visually coherent results. In addition, enriching the interaction interface with negative scribble regions offers a simple yet effective mechanism for selectively suppressing edits in areas where users wish to preserve the original content. These directions highlight the strong potential of the I^3E framework to drive the development of more flexible, intuitive, and user-centric image editing systems, and we believe they will inspire future research on controllable and region-aware visual generation.

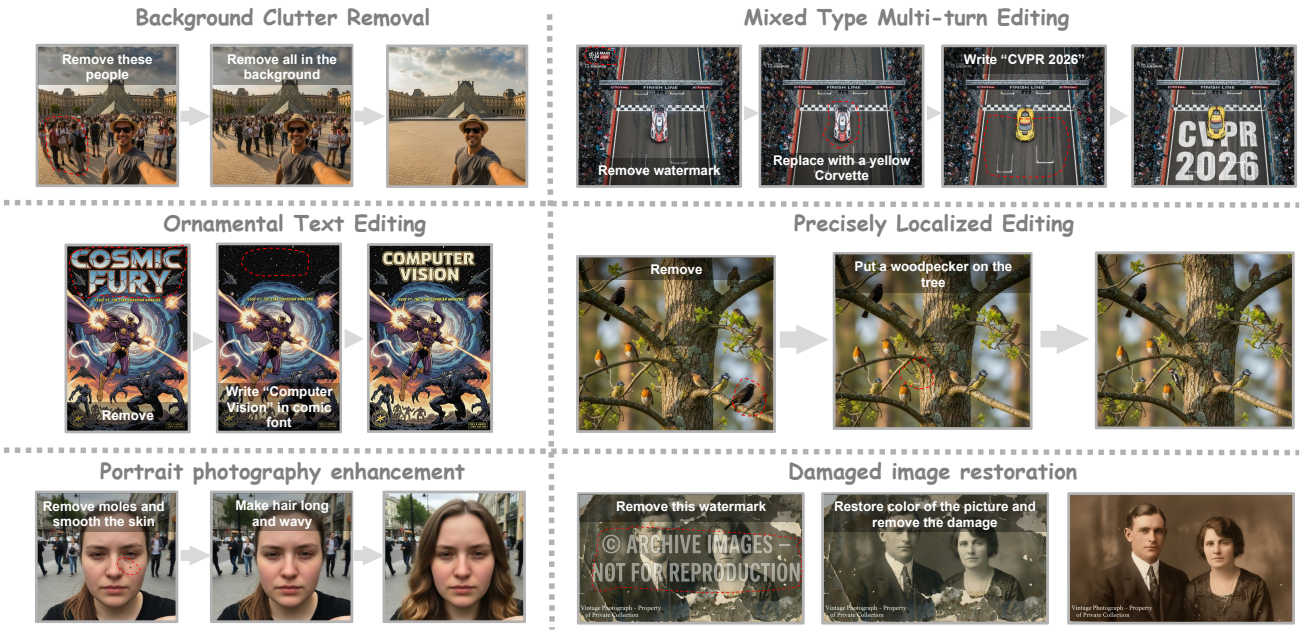

Figure A9. Additional application examples of the proposed baseline CJT method. The red dotted lines indicate the outer boundary of the user-scribbled mask. Without any extra tuning, our method can accomplish a wide variety of practically meaningful tasks.

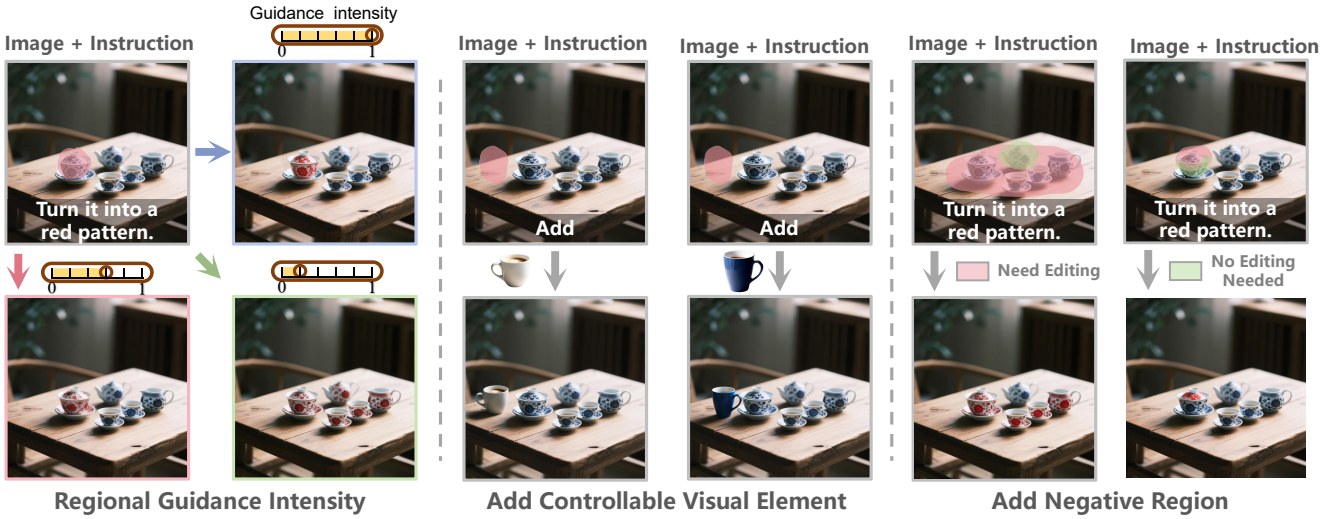

Figure A10. Three potential extensions of the I^{3}E paradigm: regional guidance intensity, controllable visual element addition, and negative scribble regions.

E. Ethical Considerations

Misuse Risk Management. The proposed baseline methods (RNI, CIA, CJT) empowers users with advanced capabilities for precise image editing, thus presenting substantial creative potential. However, the high degree of realism achievable by these methods also introduces significant risks related to misuse, including the generation of misleading or falsified imagery. Such capabilities could potentially be exploited for malicious purposes, such as fabricating evidence, spreading misinformation, or invading personal privacy.

To mitigate these risks, we require users to explicitly consent to adhere to ethical guidelines and legal regulations prior to using the technology. Users must agree not to employ the model for malicious activities or unethical objectives, including but not limited to falsifying evidence or deliberately disseminating misleading information. Ad-

ditionally, the dissemination of the model framework will incorporate educational materials emphasizing responsible usage and clear warnings against unethical applications.

Data Protection and Privacy in Inter-Edit Dataset.

The construction of the Inter-Edit dataset strictly adheres to data privacy regulations and ethical standards. The source images in our dataset are derived from two primary origins: publicly available datasets (specifically LAION [42]) and model-synthesized imagery. For the web-sourced images, we respect the original licensing and distribution terms. Furthermore, during the data processing pipeline, we employed a rigorous filtering mechanism using MLLM [2] to scrutinize the content. This step ensures the exclusion of images containing sensitive Personally Identifiable Information, offensive material, or non-consensual content. Regarding the manual annotation phase, we implemented strict protocols to protect the privacy of our annotators. Although we collected demographic information (e.g., gender and age) to ensure diversity, all annotator data is anonymized and stored separately from the dataset annotations. No individual annotator can be identified from the released dataset or the accompanying metadata. We are committed to maintaining the integrity of the dataset and will provide a channel for individuals to request the removal of their data should any privacy concerns arise post-release.

Social Responsibility and Impact. We acknowledge the profound societal implications associated with advanced image editing capabilities, recognizing their potential to trigger cultural, social, or legal controversies. The I^3E baseline models emphasizes responsible creation and editing practices. Users are advised to exercise prudence and heightened sensitivity when editing content that may be culturally sensitive, socially contentious, or legally complex. We encourage users to adopt a responsible and ethically aware stance, promoting constructive, transparent, and socially beneficial applications of this technology.