RePlan: Reasoning-Guided Region Planning for Complex Instruction-Based Image Editing

Tianyuan Qu[1,2], Lei Ke[1], Xiaohang Zhan[1], Longxiang Tang[3], Yuqi Liu[2], Bohao Peng[2], Bei Yu[2], Dong Yu[1], and Jiaya Jia[3]

[1] Tencent AI Lab, China
[2] The Chinese University of Hong Kong, China
[3] The Hong Kong University of Science and Technology, China

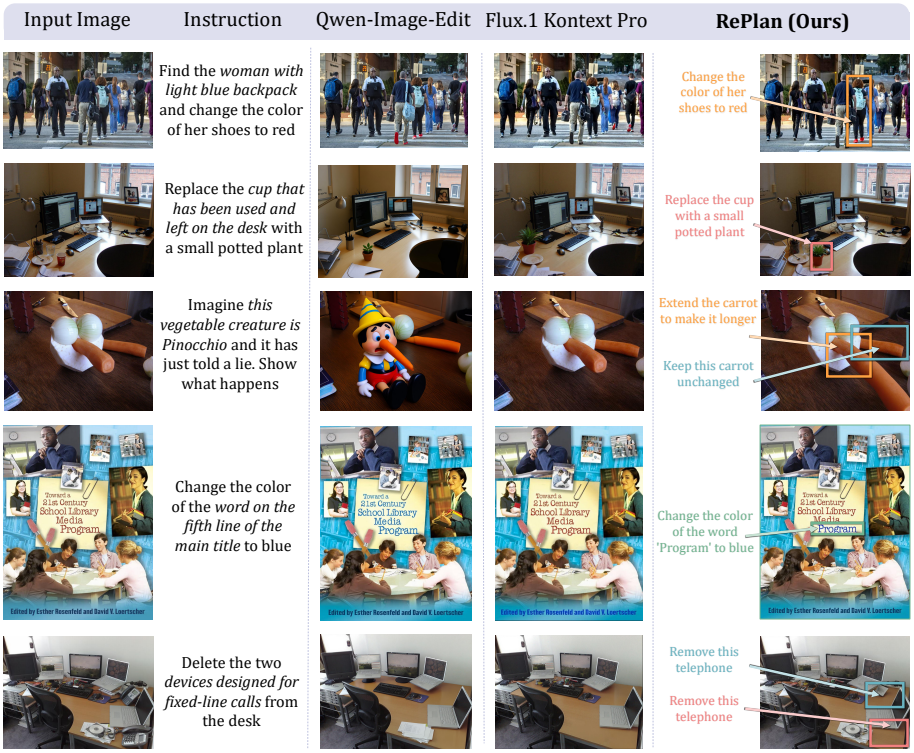

Fig. 1: We define **Instruction-Visual (IV) Complexity** as the challenges that arise from complex input images, intricate instructions, and their interactions—for example, cluttered layouts, fine-grained referring, and knowledge-based reasoning. Such tasks require models to conduct fine-grained visual reasoning. To address this, we propose **RePlan**, a framework that leverages the inherent visual understanding and reasoning capabilities of pretrained VLMs to provide region-aligned guidance for diffusion editing model, producing more accurate edits with fewer artifacts than open-source SoTA baselines [2, 25]. Zoom in for better view.

Abstract. Instruction-based image editing enables natural-language control over visual modifications, yet existing models falter under Instruction–Visual Complexity (IV-Complexity), where intricate instructions meet cluttered or ambiguous scenes. We introduce RePlan (Region-aligned Planning), a plan-then-execute framework that couples a vision–language planner with a diffusion editor. The planner decomposes instructions via step-by-step reasoning and explicitly grounds them to target regions; the editor then applies changes using a training-free attention-region injection mechanism, enabling precise, parallel multi-region edits without iterative inpainting. To strengthen planning, we apply GRPO-based reinforcement learning using 1K instruction-only examples, yielding substantial gains in reasoning fidelity and format reliability. We further present IV-Edit, a benchmark focused on fine-grained grounding and knowledge-intensive edits. Across IV-Complex settings, RePlan consistently outperforms strong baselines trained on far larger datasets, improving regional precision and overall consistency. Project page: `https://replan-iv-edit.github.io/`

Keywords: Instruction-based Image Editing · Attention Injection · Consistency · Large Vision-Language Model · Reinforcement Learning

1 Introduction

Instruction-based image editing has emerged as a core direction in multimodal AI, enabling users to flexibly modify images through natural language. Existing pure editing models [2,3,18,29] already produce diverse and high-quality visual effects, yet they still struggle with accurately grounding and executing edits within visually and linguistically complex scenarios, a challenge we formalize as **Instruction-Visual Complexity (IV-Complexity)**.

We define IV-Complexity as the intrinsic challenge that arises from the interplay between visual complexity, such as cluttered layouts or multiple similar objects, and instructional complexity, such as multi-object references, implicit semantics, or the need for world knowledge and causal reasoning. The interaction between these dimensions even amplify the challenge, requiring precise editing grounded in fine-grained visual understanding and reasoning over complex instructions. As the second row shown in Fig. 1, in a cluttered desk scene, the instruction "Replace the cup that has been used and left on the desk with a small potted plant" requires distinguishing the intended target among multiple similar objects and reasoning about implicit semantics that what counts as a "used" cup. This combined demand illustrates how IV-Complexity emerges when visual and instructional factors reinforce each other.

Recent progress in large-scale vision-language models (VLMs) [1, 4, 5, 15, 20, 24] has demonstrated strong capabilities in visual understanding and world-knowledge reasoning. A natural idea is therefore to transfer these strengths into instruction-based editing. Inspired by this, methods such as Qwen-Image [25], Bagel [9], and UniWorld [16] attempt to unify VLMs with image generation models, showing remarkable potential. However, these unified approaches typically

treat VLMs as semantic-level guidance encoders, which leads to coarse interaction with the generation model . As a consequence, even with massive training data, they still lag behind the fine-grained grounding and reasoning abilities that standalone VLMs can achieve. For example, while a VLM can correctly localize targets in a complex grounding task [17, 28], the corresponding editing model may fail to identify the same regions for modification under similar instructions.

We therefore pose the question of how the fine-grained perception and reasoning capacities of VLMs can be more effectively exploited to overcome IV-Complexity in image editing. Our key insight is that the interaction between VLMs and diffusion models should be refined **from a global semantic level to a region-specific level.** Rather than using VLMs merely as high-level semantic encoders, we harness their fine-grained perception and reasoning capabilities to generate region-aligned guidance that explicitly links decomposed instructions to target regions in the image. Building on this insight, we propose **RePlan**, a framework that couples VLMs with a diffusion-based decoder in a plan–execute manner: the VLM performs chain-of-thought reasoning to analyze the visual input and instruction, outputs structured region-aligned guidance, and the diffusion model faithfully executes this guidance to complete precise edits under IV-Complexity.

To accurately ground edits to the regions specified by the guidance, we propose a **training-free attention region injection** mechanism, which equips the pre-trained editing DiT [2, 21, 22] with precise region-aligned control and allows efficient execution across multiple regions in one pass. This avoids the image degradation issues of multi-round inpainting while reducing computation cost, offering a new perspective for controllable interactive editing. On top of this framework, we further enhance the planning ability of VLMs through GRPO reinforcement learning. Remarkably, with only ∼1k instruction-only examples, RePlan outperforms models trained on massive-scale data and computation when evaluated under IV-Complex editing task.

However, existing instruction-based editing benchmarks [18, 27] oversimplify editing scenarios by emphasizing images with salient objects and straightforward instructions. Such settings fail to reflect the real-world challenges and diverse user needs posed by IV-Complexity. To bridge this gap, we introduce **IV-Edit**, a benchmark specifically designed to evaluate instruction–visual understanding, fine-grained target localization, and knowledge-intensive reasoning.

We summarized our contributions as:

- **RePlan Framework:** We propose RePlan, which refines VLM–diffusion interaction to region-level guidance. With GRPO training from only a small set of instruction-only examples, RePlan outperforms state-of-the-art models trained on orders of magnitude more data in IV-Complexity scenarios.
- **Attention Region Injection:** We design a plug-and-play mechanism for MMDiT that enables accurate response to region-aligned guidance, while supporting multiple edits in only one-pass.
- **IV-Edit Benchmark:** We establish IV-Edit, the first benchmark tailored to IV-Complexity, providing a principled testbed for future research.

2 Related Work

Instruction-Based Image Editing. Instruction-driven image editing has advanced with diffusion-based methods. End-to-end approaches such as *InstructPix2Pix* [3,13] learn direct mappings from instructions to edited outputs, showing strong global editing but limited spatial reasoning. Inpainting-based pipelines first localize regions and then apply mask-guided editing [29], which improves locality but depends on fragile localization modules and struggles with reasoning-heavy instructions. More recent lines explore VLM-guided generation [9,25], but typically leverage VLMs only at a coarse level, underutilizing their fine-grained reasoning capabilities.

Vision–Language Models. Large VLMs [1, 5, 24] exhibit remarkable fine-grained perception [15,24] and complex reasoning abilities [12,19]. These strengths suggest great potential for boosting IV-Complex image editing.

Image Editing Benchmarks. Existing benchmarks such as Imgedit [27] and GEdit [18] mainly evaluate edits on images with salient subject and explicit instructions. Reasoning-oriented benchmarks like *KrisBench* [26] and *RISEBench* [30] move beyond direct commands, but their tasks still involve simple image compositions and fail to reflect the instruction-visual complexity of real-world image editing.

3 Method

3.1 Overview

We present a framework for complex instruction-based image editing that couples a vision–language model (VLM) with a diffusion-based decoder. The VLM interprets the input image and instruction, conducts chain-of-thought reasoning, and outputs region-aligned guidance. This guidance is executed by a DiT decoder through a training-free attention region injection, enabling one-pass, multi-region editing. The overall framework is illustrated in Fig. 2. To further strengthen the planner, we apply reinforcement learning with VLM-based feedback on post-edited results, achieving significant gains with only ∼1k instruction-only samples, without requiring paired images.

3.2 Region-aligned Editing Planner

Reasoning on Instructions. Given an input image $I \in \mathbb{R}^{H\times W\times 3}$ and a user instruction $\mathcal{T}$, the VLM planner first reasons about editing targets by combining image understanding with instruction anlysis. For ambiguous or abstract descriptions, the planner must ground high-level semantics into concrete visual effects. We further enhance this reasoning ability using the GRPO reinforcement learning algorithm (see Sec. 3.4).

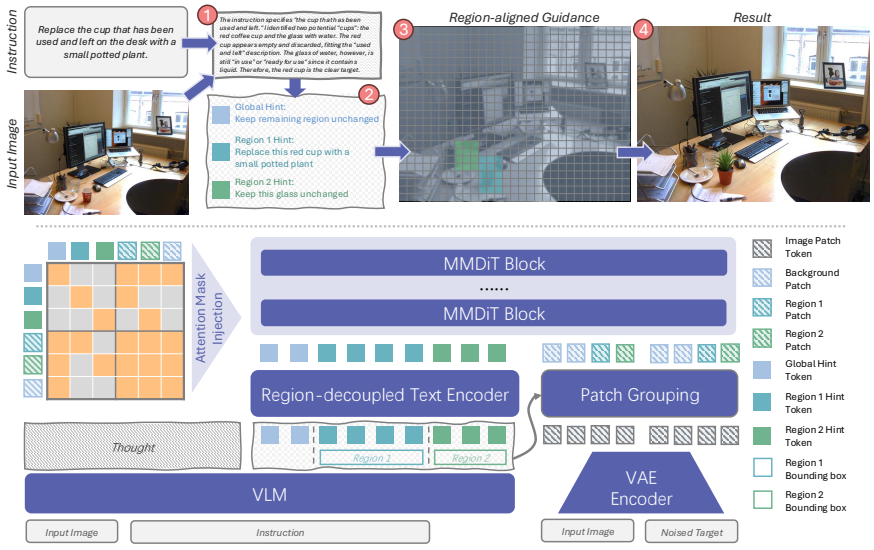

Fig. 2: Overview of our RePlan framework. The bottom part of the figure shows the overall architecture. Given an input image and text instruction, the VLM analyzes them via chain-of-thought reasoning and produces region-aligned guidance, where each guidance includes a region bbox and its editing hint. Each hint is futher encoded by a text encoder into a feature token, while image patch tokens are obtained by VAE encoding and grouped according to the region bounding boxes. A group-specific attention mechanism, detailed in Fig. 4, is proposed to allow MMDiT to generate the final edited image. The top part of the figure presents an editing examples.

Region-aligned Editing Planning. We decouple editing guidance into global and regional edits according to the targets and represent them as region–hint pairs $\{(B_k, h_k)\}_{k=0}^{K}$, where B_0 denotes the entire image (for global edits) with its associated hint h_0 (e.g., style or background adjustment), and B_k $(k \geq 1)$ are bounding boxes for local edits with corresponding hints h_k. Hints can also be negative instructing that a region remain unchanged, which helps prevent editing effects from unintentionally bleeding into neighboring areas.

<think>*The instruction specifies "the cup that has been used and left." I identified two potential "cups": the red coffee cup and the glass with water. The red cup appears empty and discarded, fitting the "used and left" description. The glass of water, however, is still "in use" or "ready for use" since it contains liquid. Therefore, the red cup is the clear target.* </think> <global> Keep remaining region unchanged </global> <region>[{"bbox_2d": [224, 372, 263, 431], "hint": "Replace this red cup with a small potted plant"}, {"bbox_2d": [175, 329, 220, 388], "hint": "Keep this glass unchanged"}]</region>

Fig. 3: Example of VLM output format

Output Format. We require the VLM to output structured text for convenient post-processing, with explicit markers separating reasoning, global edits, and

region guidance. Region guidance are expressed in JSON format. An example is shown in Fig. 3.

Interactivity. Explicit region planning enhances interpretability and controllability. When the automatically generated guidance is insufficient, users can adjust regions or the associated hints directly.

3.3 Training-free Attention Region Injection

Preliminary. Our method builds upon the MMDiT (Multimodal Diffusion Transformer) [2,11] framework for instruction-based image editing. MMDiT concatenates text, image, and latent tokens into a single sequence, which is processed jointly by Transformer self-attention, enabling rich cross-modal interaction without introducing extra modules.

The input consists of two modalities: the editing instructions $\mathcal{T}$, the original image I. They are embedded as

$$F^{\text{text}} = E_{\text{text}}(\mathcal{T}), \quad F^{\text{img}} = E_{\text{img}}(I). \tag{1}$$

The embeddings are concatenated into a unified input sequence:

$$X^{(0)} = [F^{\text{text}} \,\|\, F^{\text{img}} \,\|\, \mathbf{z}_t]. \tag{2}$$

Where $\mathbf{z}_t$ is the noised latentat step t. While full self-attention allows free information exchange across modalities, it also causes interference in multi-region editing: tokens from one region may attend to unrelated instructions, leading to *target confusion* or *instruction failure*.

Text encoding and grouping. We split the editing hints into one global hint h_0 associated with the full image B_0, and K local hints $\{h_k\}_{k=1}^{K}$ each associated with a bounding box B_k. Hints are separately encoded and concatenated as

$$F^{\text{text}} = [\, E_{\text{text}}(h_0) \,\|\, \cdots \,\|\, E_{\text{text}}(h_K) \,]. \tag{3}$$

We thus define token index groups $G_0^{\text{text}}, \ldots, G_K^{\text{text}}$ accordingly.

Image encoding and patch grouping. The image I is first processed by the VAE encoder to produce a spatial feature map F^{img} which can be reshaped into patch

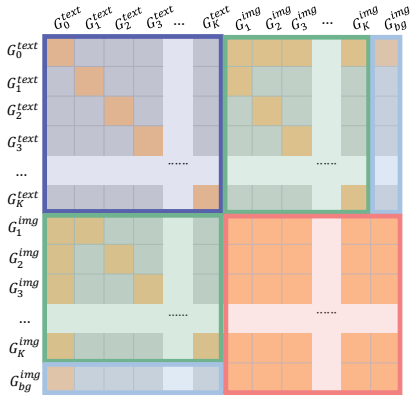

Fig. 4: Attention rule visualization. We use different highlight colors to indicate different rules, which correspond to Hint isolation, Region constraint, Background constraint and Image–latent full interaction.

tokens $\{f_{i,j}\}_{i=1..M,\, j=1..N}$. Each editing region B_k is then mapped into the patch grid to collect the corresponding group:

$$G_k^{\text{img}} = \{f_{i,j} \mid (i,j) \in B_k\}, \quad k = 1, \ldots, K. \tag{4}$$

while the background group is defined as patches not belong to any region groups:

$$G_{\text{bg}}^{\text{img}} = \{f_{i,j}\}_{i,j} \setminus \bigcup_{k=1}^{K} G_k^{\text{img}}. \tag{5}$$

Attention mask manipulation. In each attention layer, we impose a binary mask $M \in \{0,1\}^{|X|\times|X|}$ controlling which tokens can attend to which others. The mask follows five intuitive rules, as also visualized in Fig. 4:

1. **Intra-group interaction.** Tokens within the same group (text, image, or latent) are fully connected. This ensures that local context is preserved inside each modality or region.
2. **Hint isolation.** Different text groups G_a^{text} and G_b^{text} $(a \neq b)$ are not allowed to see each other. This prevents regional instructions from contaminating one another and avoids semantic conflicts.
3. **Image–latent full interaction.** All image and latent tokens remain globally connected. This ensuring global stylistic coherence and smooth boundaries. Meanwhile, the effect can extend beyond the bounding box when necessary.
4. **Region constraint.** Tokens belonging to region B_k $(u \in G_k^{\text{img}})$ may only attend to their own hint tokens G_k^{text} and the global instruction G_0^{text}. In this way, local edits are precisely guided by their designated hints while still aligned with the global change.
5. **Background constraint.** Background tokens $u \in G_{\text{bg}}^{\text{img}}$ can only attend to the global text group G_0^{text}. This keeps the untouched background in sync with the global instruction without being polluted by local edits.

Together, these rules ensure that (i) text instructions are disentangled, (ii) image spaces preserve global coherence, and (iii) each regional hint remains focused on its designated area. As a result, MMDiT can execute multiple region-level edits in parallel, enhancing efficiency while avoiding the accumulated errors of multi-round inpainting, and further supports region-level negative prompts.

3.4 Structured Planning and Reasoning with GRPO

We perform reinforcement learning on the pretrained vision–language model (VLM) to improve its planning capabilities for complex instruction-based editing. Specifically, we use Qwen2.5-VL 7B [1] as the VLM planner, and Flux Kontext Dev [2] as the diffusion image decoder.

We adopt Group Relative Policy Optimization (GRPO) [23], which updates the planner by comparing the relative quality of multiple edited outputs generated from the same instruction.

Since rewards rely on valid image outputs, errors in plan formatting can disrupt decoding and lead to distorted reward signals. To address this, we employ a two-stage training strategy: we first focus on improving plan validity and reasoning quality, and then introduce image-level rewards to refine planning behavior.

Both stages use only 1k complex instruction editing samples we generated for supervised alignment before reinforcement learning.

Stage 1: Format and reasoning learning. In the first stage, GRPO training provides only format-related rewards to ensure structured plan generation and coherent reasoning:

- **Tag format reward.** A regular expression parser checks whether the output follows the tag structure in Fig. 3. A valid structure yields a positive reward; otherwise zero.
- **Region format reward.** The content inside the `<region>` tag is parsed as JSON, including the outer list and each inner dictionary. Valid JSON yields a positive reward; otherwise zero.
- **Reasoning quality reward.** The length of the content inside the `<think>` tag is measured, and the reward increases with length from zero up to a capped maximum.

The Stage 1 reward can be computed as:

$$R^{(1)} = R_{Ta} + R_F + R_R.$$

Stage 2: Planning learning. In the second stage, plans are decoded into images, and a larger VLM provides image-level evaluation. We adopt Qwen2.5-VL 72B as the reward model:

- **Target** (R_T): Whether the edit is applied to the specified area.
- **Effect** (R_E): Whether the visual change matches the instruction.
- **Consistency** (R_C): Reservation of irrelevant regions and global style.

To prevent reward hacking (e.g., maximizing consistency by making no edits), consistency is re-weighted by effect: $R'_C = R_C \cdot R_E$. Finally, the Stage 2 reward is:

$$R^{(2)} = R_T + R_E + R'_C + \lambda R^{(1)},$$

where λ is a small weight that preserves format reliability.

4 Data Construction and Benchmark

4.1 Task Setting

IV-Complexity highlights the difficulty of faithfully grounding user intent in rich visual contexts, where instructions involve complex referring expressions

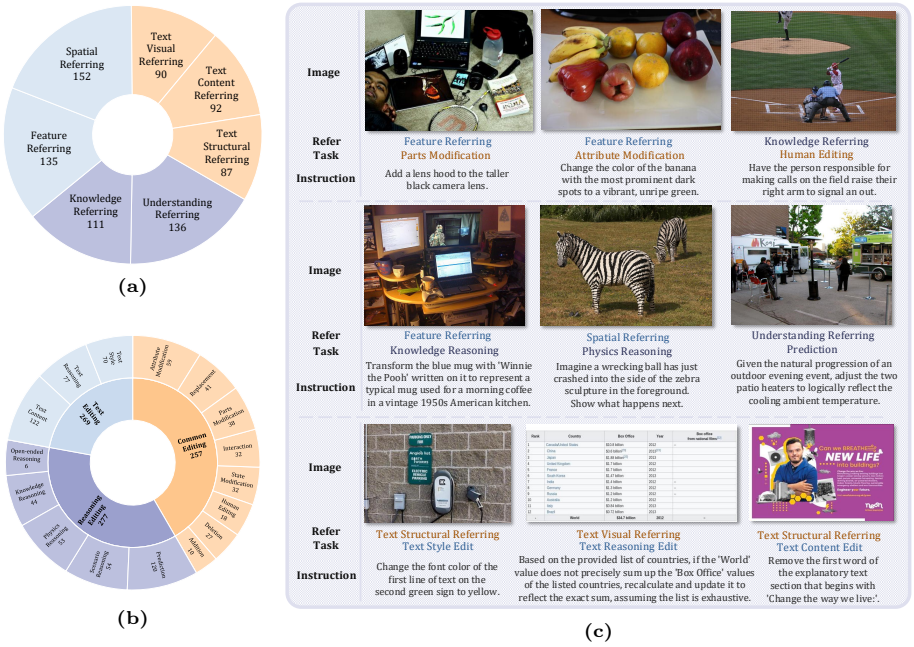

Fig. 5: Overview of our IV-Edit Benchmark. (a) and (b) respectively shows the distribution of referring types and task types across the dataset. IV-Edit is explicitly designed to reflect the IV-Complexity challenge, where user instructions require aligning fine-grained language with rich and diverse visual contexts. (c) presents visual examples spanning a wide range of real-world scenarios and fine-grained instruction intents—including spatial, structural, and reasoning-intensive edits. Each instruction can be decomposed into a referring expression and a task type.

and require fine-grained reasoning to align language with the correct visual regions. To reflect these challenges, we design IV-Edit to be domain-diverse and reasoning-sensitive, covering two complementary scenarios: (1) complex real-world photo editing and (2) text-related image editing (e.g., posters, slides, charts, and tables). Across both scenarios, we intentionally prioritize non subject-dominated images with semantics distributed across multiple objects/regions, reducing single-salient-object shortcuts and encouraging robust instance disambiguation. Correspondingly, our instructions emphasize detailed grounded understanding and frequently require reasoning under contextual and real-world constraints (e.g., spatial relations, functional/knowledge cues, and structured text/table semantics).

Specifically, each instruction is constructed by composing a *referring expression* and an *editing task*. This compositional formulation induces diverse combinations of grounding mechanisms and edit intents, and enables systematic evaluation of instruction following under varied visual domains and reasoning

requirements. We consider 7 referring types and 16 task types, as illustrated in Fig. 5a and Fig. 5b; full definitions are provided in Appendix.

4.2 Benchmark Statistics

With careful filtering and manual verification, IV-Edit comprises around 800 high-quality instruction–image pairs collected from multiple public data sources (see Appendix), covering heterogeneous visual domains and editing effects. This scale enables comprehensive evaluation at manageable cost. On average, instructions contain 21 words, and 182 examples involve edits over multiple target regions, increasing diversity via multi-target grounding. The distributions of referring types and task types are summarized in Fig. 5a and Fig. 5b, while Fig. 5c shows representative cases. Additional statistics are provided in Appendix.

4.3 Evaluation Protocol

Since the input images are not dominated by a single subject, traditional metrics that compute global semantic similarity based on CLIP are not well suited. We introduce Gemini2.5-Pro as a fine-grained evaluator, assigning 5-point ratings to image pairs before and after editing along the following four dimensions:

- **Target**: Whether the target region is correctly located and modified, avoiding confusion between editing objects. When multiple different targets need to be edited, whether there are no omissions.
- **Consistency**: Whether the content outside the edited region remains unchanged, without editing effects spilling into unrelated areas. In addition, whether the overall style before and after editing remains stable.
- **Quality**: The visual quality of the editing results. Regardless of the editing instructions, whether the edited image contains artifacts or style conflicts between different regions.
- **Effect**: Whether the visual effect of the editing instruction is accurately achieved.

5 Experiments

5.1 Results on IV-Edit Benchmark

Metric. We evaluate along four dimensions from Sec. 4 using Gemini-2.5-Pro. The Overall score is the simple average of these dimensions. To avoid inflated Consistency when no edits are made, we introduce a Weighted score that weights Consistency by Effect, defined as Weighted $= \sum_{\text{samples}}(\text{Target} + \text{Quality} + \text{Effect} + \text{Effect} \times \text{Consistency})/4$, with all scores ranging from 1 to 5.

Evaluation Setting. We conduct evaluations on a total of two closed source models and six open source models. The closed source models include GPT-4o [14] and Gemini-2.5-Flash-Image [6] (also referred to as nano banana). The open source models evaluated are InstructPix2Pix [3], Uniworld [16], Bagel [9](using the think mode), Qwen-Image [25], Flux.1-Kontext-dev [2], Flux.2-Klein-9B, and our proposed RePlan. For our RePlan framework, we conduct evaluations by applying it to Flux.1-Kontext-dev, Qwen-Image-Edit and Flux.2-Klein-9B, all of which share the MMDiT architecture.

Table 1: Quantitative comparison of open-source and proprietary image editing models across four evaluation dimensions, with Overall and Weighted scores also reported. **RePlan** consistently yields significant improvements when applied to different MMDiT backbones, and achieves the best Consistency and Overall among open-source models.

Model	Quality ↑	Target ↑	Effect ↑	Cons. ↑	Overall ↑	Weighted ↑
Gemini-Flash-Image	3.89	4.11	3.93	2.89	3.71	3.44
GPT-4o	3.61	4.02	3.78	1.77	3.30	3.07
InstructPix2Pix	2.47	2.47	1.90	1.40	2.06	1.48
Uniworld-V1	3.26	2.89	2.18	1.46	2.45	1.84
Bagel-Think	3.44	3.47	2.93	2.33	3.05	2.46
Flux.1-Kontext-dev	3.93	3.34	2.73	2.88	3.22	2.49
+ RePlan (Ours)	**4.16**	**3.47**	**2.59**	**3.64**	**3.46**	**2.55**
Qwen-Image-Edit	3.47	3.72	3.24	1.79	3.05	2.62
+ RePlan (Ours)	**3.86**	**3.77**	**3.16**	**3.24**	**3.51**	**2.91**
Flux.2-Klein-9B	4.05	3.81	3.31	2.61	3.45	2.95
+ RePlan (Ours)	**4.28**	**4.04**	**3.41**	**3.96**	**3.92**	**3.33**

Quantitative Analysis. Tab. 1 reports the evaluation results. The accuracy of handling referring expressions is measured from two perspectives: Target and Consistency. Target emphasizes recall, capturing the model's ability to semantically localize the editing object, while Consistency emphasizes precision, reflecting fine-grained localization at the regional level. For Target, Qwen-Image and Bagel perform strongly by leveraging VLMs, which better resolve complex semantic referring and the underlying intentions in instructions. Regardless of which is used as the MMDiT backbone, RePlan shows a significant performance improvement, highlighting its superior reasoning capability in analyzing and grounding IV-complex instructions.

RePlan especially achieves a clear advantage in Consistency, benefiting from directional regional injection that prevents editing spillover into semantically similar region, a common drawback of semantic-level guidance methods.

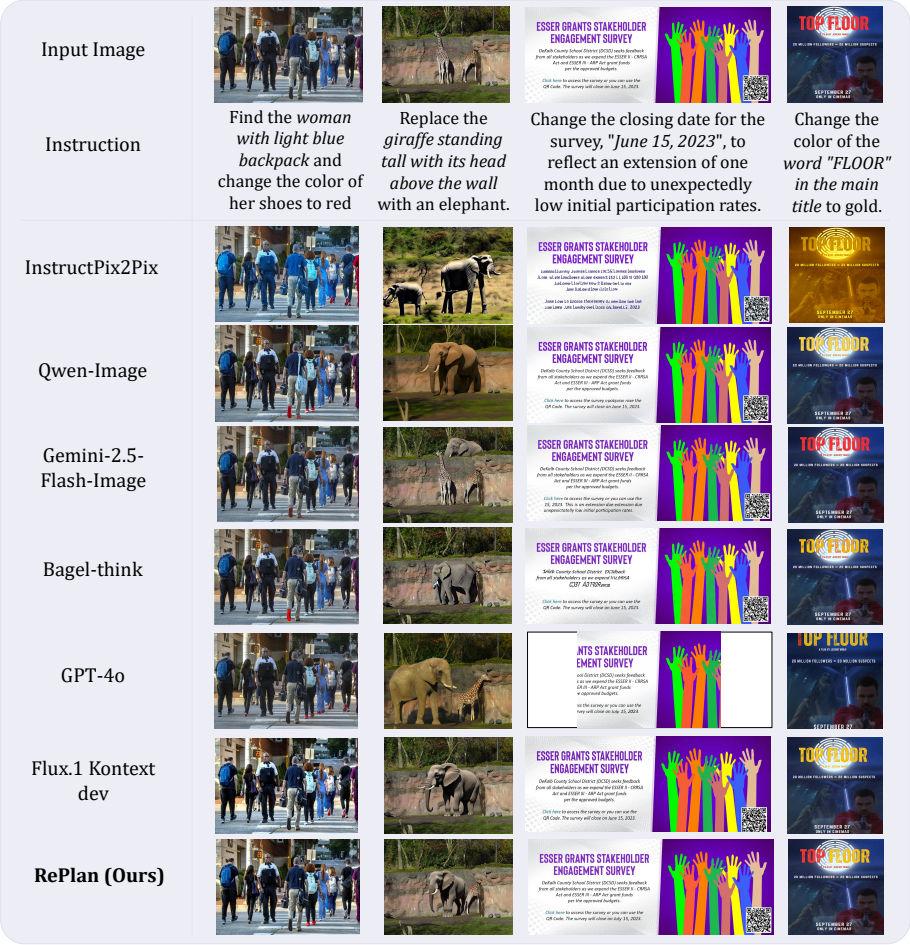

Fig. 6: Editing results comparison. We use Flux.1-Kontext-dev as the backbone of RePlan. Notably, GPT-4o enforces fixed aspect ratios, leading to unavoidable cropping for non-standard images. Zoom in for better view.

Qualitative Analysis. By comparing the editing results in Fig. 6, we observe that our RePlan demonstrates clear advantages in accurately localizing the target editing regions. In contrast, other existing methods tend to suffer from editing spillover into semantically similar areas, a problem that persists even in state-of-the-art proprietary models. In addition, RePlan shows stronger reasoning ability for handling indirect instructions; for example, in the third case it not only identifies the word "June" in the image but also infers that the next month is "July." More comparative results can be found in Appendix.

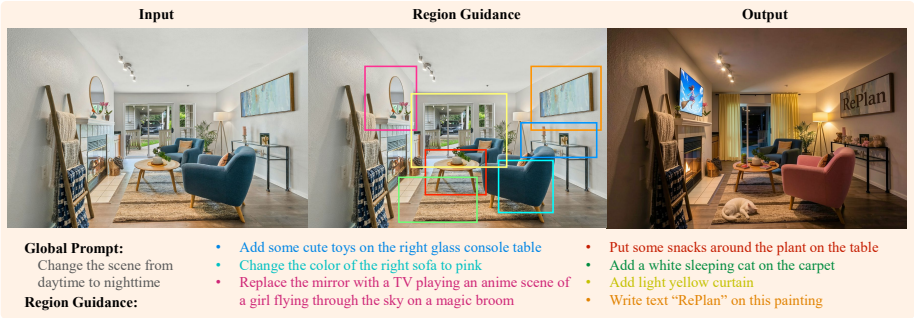

Fig. 7: Parallel editing across numerous regions. Multiple regional instructions together with a global modification are edited simultaneously in a single forward. The edits remain well localized without spillover while maintaining globally coherent visual appearance, effectively avoiding issues such as cross-region interference and instruction omissions commonly encountered by existing models in highly complex scenarios.

Human Preference Study. To complement automatic metrics, we conduct a blinded human preference study on IV-Edit. We randomly sample 100 instruction-image pairs and compare the editing results produced by Flux.1-Kontext-dev and our RePlan (with Flux.1-Kontext-dev as backbone). For each pair, we present the two anonymized outputs in randomized left/right order. We recruit 12 human experts, who are unaware of the model identities, and ask them to vote for (i) which result better follows the editing instruction, or (ii) a tie if they are comparable in instruction-following quality. We aggregate votes per sample via majority vote to obtain the final label.

Overall, RePlan is preferred in 35% of the cases, Flux.1-Kontext-dev is preferred in 15% of the cases, and the remaining 50% are judged as ties. Excluding ties, RePlan wins 70% of the non-tie cases, providing additional human preference evidence for the effectiveness of RePlan.

Scalable Multi-Region Editing. Fig. 7 demonstrates that the proposed attention injection mechanism scales to more regions. Multiple regional instructions and a global lighting adjustment are executed simultaneously in a single forward pass, enabling efficient parallel editing. Even with numerous regions, edits remain well localized without affecting neighboring areas, while the global adjustment is applied coherently across the entire image.

5.2 Ablation Study

Ablation on Planner. We test RePlan on IV-Edit using Gemini2.5-Pro and Qwen2.5-VL [1] as planners without RL. As shown in Tab. 2, both lag behind the RL-trained planner. Manual inspection reveals that Gemini2.5-Pro, though strong in reasoning, often produces bbox errors, while Qwen2.5-VL struggles with

Table 2: Ablation studies on VLM region guidance planner and RL training strategy.

Zero-shot VLM Region Planner			**Reasoning and Staged RL Training**		
Model	Overall ↑	Weighted ↑	Model	Overall ↑	Weighted ↑
Gemini2.5-pro	2.95 (-0.51)	1.93 (-0.62)	w/o reasoning	3.31 (-0.15)	2.49 (-0.06)
Qwen2.5-VL 7B	2.60 (-0.86)	1.63 (-0.92)	Uni Stage RL	3.42 (-0.04)	2.51 (-0.04)
RePlan (Kontext)	3.46	2.55	RePlan (Kontext)	3.46	2.55

Table 3: Average inference latency (seconds) on IV-Edit with different numbers of regions. Here we use Flux.1-Kontext-dev as the MMDiT backbone.

Method	**Acceleration**	**Mask**	**1**	**2**	**3**	**4**	**5**
Kontext (multi-turn)	FlashAttention2	✗	44.2	88.4	132.6	176.8	221.0
Replan (w/o accel.)	SDPA	✓	83.1	89.3	95.7	104.7	113.8
Replan	**Rule-group Attn.**	✓	**44.2**	**46.1**	**48.8**	**51.5**	**54.1**

hint decomposition and format compliance. These results highlight the necessity of RL for a reliable planner.

Ablation on Reasoning. To assess the role of CoT reasoning, we remove it and train the VLM to directly output region-aligned guidance (Tab. 2). Performance drops markedly without reasoning. Combined with the planner ablation, this underscores its importance in analyzing instructions and producing effective guidance.

Ablation on RL Stage. We further evaluate the two-stage RL strategy (Sec. 3.4) by skipping the first-stage format learning. Results show that the full two-stage scheme not only achieves higher final scores under the same training steps, but also delivers superior sample efficiency. This validates both the effectiveness and efficiency of the strategy.

5.3 Inference Latency

Rule-group Attention Acceleration. Our attention injection imposes structured visibility constraints. Implementing them as a generic dense mask can prevent efficient attention kernels (e.g., FlashAttention [7, 8]) and significantly increase HBM traffic.

We leverage that the number of rule groups k is small ($k \ll n$, typically < 10) and visibility depends only on group membership. Instead of storing an explicit $n \times n$ mask, each token t carries two bitsets $\mathbf{K}_t, \mathbf{Q}_t \in \{0,1\}^k$ defined as

$$(\mathbf{K}_t)_i = \mathbb{I}[t \in G_i], \qquad (\mathbf{Q}_t)_i = \mathbb{I}[t \text{ is allowed to attend to } G_i],$$

where G_i denotes rule group i and $\mathbb{I}[\cdot]$ is the indicator function. Visibility is computed on-the-fly as

$$\text{visible}(t, s) \iff (\mathbf{Q}_t \,\&\, \mathbf{K}_s) \neq \mathbf{0},$$

where & denotes bitwise AND. We implement this with FlexAttention [10] by fusing the visibility test into the attention kernel. The extra memory drops from $O(n^2)$ (explicit mask) to $O(n\lceil k/W \rceil)$ machine words, where W is the word size in bits; for small k, the added compute cost is negligible.

Latency Comparison. Tab. 3 reports end-to-end inference latency on IV-Edit. Rule-group attention significantly reduces the overhead of masked attention, achieving a consistent $\sim 2\times$ speedup over a naive masked SDPA implementation. With a single region, RePlan with masking matches the latency of the no-mask FlashAttention2 baseline, indicating that our rule-group attention preserves the efficiency of FlashAttention-style kernels. As the number of regions increases, latency grows only mildly, mainly due to longer text sequences.

Compared with multi-turn iterative editing, RePlan is substantially more efficient for multi-region scenarios. Iterative pipelines are inherently sequential, leading to near-linear latency growth, whereas RePlan processes all regions jointly in a single pass via injected visibility constraints, maintaining high efficiency as the number of regions increases.

6 Conclusion

We introduce IV-Complexity, a new challenge from cluttered visuals and ambiguous instructions. Existing methods depend on coarse semantic guidance, limiting fine-grained control. To address this, we propose RePlan, which uses VLM-based region reasoning with diffusion models and a training-free attention injection for precise parallel edits. RePlan is applicable to various MMDiT-based backbones, consistently bringing significant improvements without compromising efficiency. We also release IV-Edit, the first benchmark for IV-Complexity, providing a principled testbed for real-world instruction-based editing.

Acknowledgements

This work was supported in part by the Research Grants Council under the Areas of Excellence scheme grant AoE/E-601/22-R.

References

1. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
2. Batifol, S., Blattmann, A., Boesel, F., Consul, S., Diagne, C., Dockhorn, T., English, J., English, Z., Esser, P., Kulal, S., et al.: Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. arXiv e-prints pp. arXiv–2506 (2025)

3. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 18392–18402 (2023)
4. Chen, Z., Wang, W., Cao, Y., Liu, Y., Gao, Z., Cui, E., Zhu, J., Ye, S., Tian, H., Liu, Z., et al.: Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. arXiv preprint arXiv:2412.05271 (2024)
5. Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Zhong, M., Zhang, Q., Zhu, X., Lu, L., et al.: Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 24185–24198 (2024)
6. Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al.: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261 (2025)
7. Dao, T.: Flashattention-2: Faster attention with better parallelism and work partitioning. arXiv preprint arXiv:2307.08691 (2023)
8. Dao, T., Fu, D., Ermon, S., Rudra, A., Ré, C.: Flashattention: Fast and memory-efficient exact attention with io-awareness. Advances in neural information processing systems **35**, 16344–16359 (2022)
9. Deng, C., Zhu, D., Li, K., Gou, C., Li, F., Wang, Z., Zhong, S., Yu, W., Nie, X., Song, Z., et al.: Emerging properties in unified multimodal pretraining. arXiv preprint arXiv:2505.14683 (2025)
10. Dong, J., Feng, B., Guessous, D., Liang, Y., He, H.: Flex attention: A programming model for generating optimized attention kernels. arXiv preprint arXiv:2412.05496 **2**(3), 4 (2024)
11. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-first international conference on machine learning (2024)
12. Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y.: Deep learning, vol. 1. MIT Press (2016)
13. Hui, M., Yang, S., Zhao, B., Shi, Y., Wang, H., Wang, P., Zhou, Y., Xie, C.: Hq-edit: A high-quality dataset for instruction-based image editing. arXiv preprint arXiv:2404.09990 (2024)
14. Hurst, A., Lerer, A., Goucher, A.P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al.: Gpt-4o system card. arXiv preprint arXiv:2410.21276 (2024)
15. Lai, X., Tian, Z., Chen, Y., Li, Y., Yuan, Y., Liu, S., Jia, J.: Lisa: Reasoning segmentation via large language model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9579–9589 (2024)
16. Lin, B., Li, Z., Cheng, X., Niu, Y., Ye, Y., He, X., Yuan, S., Yu, W., Wang, S., Ge, Y., et al.: Uniworld: High-resolution semantic encoders for unified visual understanding and generation. arXiv preprint arXiv:2506.03147 (2025)
17. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
18. Liu, S., Han, Y., Xing, P., Yin, F., Wang, R., Cheng, W., Liao, J., Wang, Y., Fu, H., Han, C., et al.: Step1x-edit: A practical framework for general image editing. arXiv preprint arXiv:2504.17761 (2025)

19. Liu, Y., Peng, B., Zhong, Z., Yue, Z., Lu, F., Yu, B., Jia, J.: Seg-zero: Reasoning-chain guided segmentation via cognitive reinforcement. arXiv preprint arXiv:2503.06520 (2025)
20. Liu, Y., Qu, T., Zhong, Z., Peng, B., Liu, S., Yu, B., Jia, J.: Visionreasoner: Unified visual perception and reasoning via reinforcement learning. arXiv preprint arXiv:2505.12081 (2025)
21. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4195–4205 (2023)
22. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
23. Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al.: Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300 (2024)
24. Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., et al.: Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. arXiv preprint arXiv:2409.12191 (2024)
25. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., Yin, S.m., Bai, S., Xu, X., Chen, Y., et al.: Qwen-image technical report. arXiv preprint arXiv:2508.02324 (2025)
26. Wu, Y., Li, Z., Hu, X., Ye, X., Zeng, X., Yu, G., Zhu, W., Schiele, B., Yang, M.H., Yang, X.: Kris-bench: Benchmarking next-level intelligent image editing models. arXiv preprint arXiv:2505.16707 (2025)
27. Ye, Y., He, X., Li, Z., Lin, B., Yuan, S., Yan, Z., Hou, B., Yuan, L.: Imgedit: A unified image editing dataset and benchmark. arXiv preprint arXiv:2505.20275 (2025)
28. Yu, L., Poirson, P., Yang, S., Berg, A.C., Berg, T.L.: Modeling context in referring expressions. In: European conference on computer vision. pp. 69–85. Springer (2016)
29. Zhang, K., Mo, L., Chen, W., Sun, H., Su, Y.: Magicbrush: A manually annotated dataset for instruction-guided image editing. Advances in Neural Information Processing Systems **36**, 31428–31449 (2023)
30. Zhao, X., Zhang, P., Tang, K., Zhu, X., Li, H., Chai, W., Zhang, Z., Xia, R., Zhai, G., Yan, J., et al.: Envisioning beyond the pixels: Benchmarking reasoning-informed visual editing. arXiv preprint arXiv:2504.02826 (2025)