Edit-R2: Context-Aware Reinforcement Learning for Multi-Turn Image Editing

Yuxiao Ye[1,2*] **Haoran He**[1] **Fangyuan Kong**[2] **Xintao Wang**[2] **Pengfei Wan**[2]
Kun Gai[2] **Ling Pan**[1†]

[1]Hong Kong University of Science and Technology [2]KlingAI

Abstract

Text-guided image editing has advanced rapidly with diffusion models and unified multimodal foundation models. However, most existing methods remain confined to single-turn settings, overlooking the more realistic scenario of multi-turn in-context editing, where users iteratively refine an image through a sequence of instructions. In this setting, a model must follow each new instruction while preserving accumulated session-level constraints, challenged by two coupled failure modes: long-context dilution, where sparse textual constraints become difficult to recover from growing interleaved image-text histories, and state contamination, where earlier editing mistakes degrade subsequent generations. We introduce Edit-R2, a novel reinforcement learning post-training framework for unified multimodal models. Edit-R2 reconstructs the operative session intent, which effectively consolidates scattered historical constraints into an explicit reasoning trace before each editing turn. It further enables multi-turn RL over both reasoning and generation through a unified objective that jointly optimizes intent reconstruction generation in discrete text space and flow-matching image generation in continuous latent space, while a trajectory filtering mechanism suppresses corrupted rollouts to stabilize training under state contamination. To support systematic evaluation, we introduce MICE-Bench, a large-scale benchmark for multi-turn in-context editing with automated metrics for instruction following (IF), content consistency (CC), and global awareness (GA) over accumulated session constraints. Experiments show that Edit-R2 substantially improves multi-turn in-context editing and achieves competitive performance compared against strong baselines. Code is available at `https://github.com/yuxiaooye/Edit-R2`.

1 Introduction

Text-guided image editing has advanced rapidly with diffusion and unified multi-modal foundation models [28], which enables diverse natural-language edits such as object insertion, attribute manipulation, and semantic replacement. Despite its progress, the dominant formulation remains largely *single-turn*: given input image(s) and one instruction, the model is expected to produce one edited image [27, 36, 33, 37]. This isolated editing formulation does not reflect realistic editing workflows and leaves a key aspect of interactive editing underexplored. In practice, editing is often an iterative session: a user may first change an object, then adjust the background, and finally revise a previous preference, while expecting the system to preserve some of the earlier requirements and interpret new instructions without context misunderstanding. Therefore, a capable editing system should operate not only on the current image and instruction, but also on the accumulated session context for supporting session-level behavior and optimization.

*This work was conducted during the author's internship at KlingAI.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

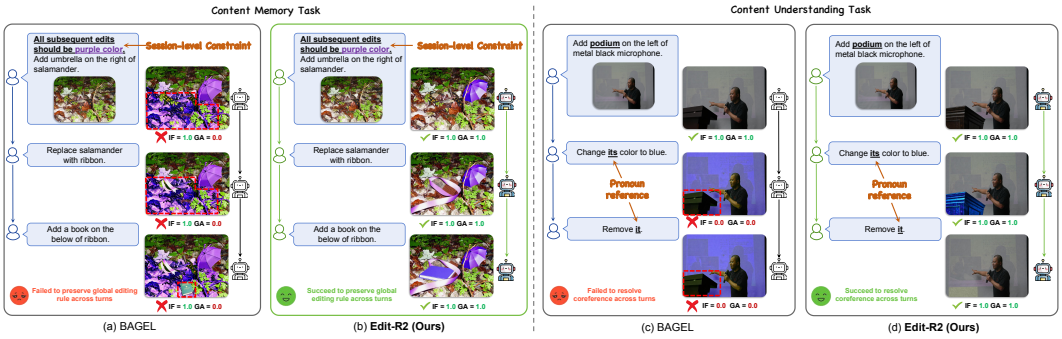


Figure 1: Overview of in-context editing. (a)-(b): *Content memory* task. A session-level constraint requires all added or modified elements to be purple. BAGEL violates this constraint throughout the session, while Edit-R2 faithfully adheres to it, achieving GA = 1.0 at every turn. (c)-(d): *Content understanding* task. The user intends to use "its" to refer to the podium added in the preceding turn 1. BAGEL fails to resolve the coreference, whereas Edit-R2 correctly edits the intended object at every turn. Instruction is simplified for demonstration.

We term this paradigm *multi-turn in-context image editing* (or ***in-context editing*** for brevity[3]): at each turn, the model receives an interleaved history of previous instructions and generated images, and must produce an edited image that follows the new instruction while maintaining consistency with constraints introduced in prior turns (see Figure 1). In multi-turn sessions, the current instruction may contain pronouns, ellipses, reversions, or preferences whose meaning depends on earlier turns. Meanwhile, some historical requirements should persist, whereas others may be overridden by later instructions. The current image alone may not reveal which previous constraints should remain active. Thus, the model must infer the operative session intent from a sparse, implicit, and dynamically updated image-text history. Despite its practical importance, this in-context capability remains underexplored in current image editing models, datasets, and training paradigms. Most benchmarks evaluate isolated input-instruction-output triples [16, 8], and existing multi-turn benchmarks [35, 31, 20, 6] remain *history-agnostic*: each turn can be solved from the immediately preceding image and the current instruction, without requiring explicit retention of earlier constraints.

Given the sequential nature of in-context editing, we formulate this as a session-level reinforcement learning (RL) problem and develop **Edit-R2**, a new RL paradigm that optimizes editing beyond traditional isolated local edits. Under this crucial scenario, each generated image becomes part of the state for future turns, and the value of an edit depends not only on its compliance with the current instruction, but also on its ability to preserve the context needed in later turns. This long-horizon dependency makes RL a natural fit. Although recent advances in RL have exhibited strong gains, they are largely limited to single-turn visual generation [13, 30, 38], where the policy is optimized against a static conditioning context [17, 11, 32]. Transitioning to this multi-turn paradigm poses a significant challenge, where *image editing models easily get lost in the long interleaved history, leading to severe long-context intent dilution* (see Figure 2). As the session progresses, the interleaved image-text context grows rapidly, which poses critical challenges for the policy to recover the active session intent. In unified models, this imbalance is further amplified by the heavy visual-token budget required to encode generated images [25], making critical textual constraints increasingly sparse and distant in the context across history. Furthermore, even when an edit appears acceptable for the current instruction, any missed or incorrect detail is carried into the next turn, where it can *mislead subsequent edits* and make the rollout unreliable for training.

Edit-R2 is a novel RL framework that addresses these critical bottlenecks with explicit session-intent recovery and trajectory-level stabilization. Built upon the unified model BAGEL [5], Edit-R2 first reconstructs the operative intent for the current turn via in-context chain-of-thought (IC-CoT), which reasons over the dialogue history and consolidates scattered historical constraints into a compact reasoning trace, mitigating the dilution of early textual constraints and providing the generator with a stable state representation. Crucially, IC-CoT cannot be optimized in isolation, as an apparently reasonable intent reconstruction is useful only if it leads to successful downstream image editing. To achieve the synergy, we jointly optimize IC-CoT generation and visual generation with

[3]Prior works [7, 27] use "in-context" to refer to single-turn editing conditioned on multiple reference images, which differs from our usage.

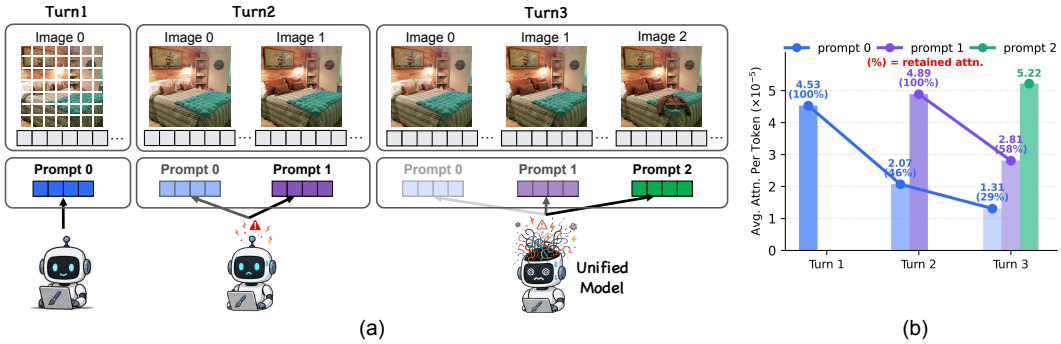


Figure 2: (a): Long-context issue for in-context editing. (b): Attention magnitude of the flow-matching generator across prompts: As editing progresses, attention to prompts from earlier turns weakens significantly, leading to context loss.

unified multi-turn RL, which couples the discrete text space of session-intent reconstruction with the continuous latent space of the flow-matching image generator. To prevent corrupted rollouts from dominating training and contaminating later training steps, Edit-R2 further introduces a trajectory filtering mechanism for stable optimization. Finally, to provide both the training environment and evaluation protocol required for multi-turn RL and resolve the evaluation bottleneck, we construct MICE-Bench, a large-scale benchmark for in-context editing covering *content memory* and *content understanding* tasks with a fully automated evaluation pipeline built on comprehensive metrics: instruction following (IF), content consistency (CC) [3], and newly introduced global awareness (GA) measuring compliance with accumulated session-level constraints. Experiments on MICE-Bench demonstrate that Edit-R2 substantially outperforms existing open-source models and achieves competitive performance against closed-source counterparts.

Our contributions are summarized as follows: **(i)** We formulate multi-turn in-context image editing as a session-level RL problem and introduce MICE-Bench, the first large-scale automated benchmark for this setting, accompanied by a new GA metric that explicitly quantifies session-level constraint compliance. **(ii)** We propose Edit-R2, a novel RL post-training framework that reconstructs the active session intent with IC-CoT and jointly optimizes intent reconstruction and visual generation across discrete text and continuous latent spaces, with trajectory filtering for stable multi-turn training. **(iii)** Extensive experiments show that Edit-R2 substantially improves BAGEL **(+18% IF, +18% GA)** and **(+8% IF, +15% GA)**, achieving competitive performance over strong models.

2 Related Work

Instruction-based Image Editing Models. InstructPix2Pix [2] pioneered the learning-based image editing, subsequent models [23, 16] such as Qwen-Image-Edit [26] and FLUX.1 Kontext [10] extend with stronger architectures and larger-scale training. Recently, image editing has evolved rapidly from single-turn [27, 36] to multi-turn paradigms. Several training-free methods aim to mitigate quality degradation across successive turns [12, 23]. Auto-regressive models including Nano-Banana-2 [4], GPT-Image-1 [19], OmniGen [29], and BAGEL [5] enable in-context editing. Despite these advances, existing models struggle to resolve the long-horizon challenges inherent to in-context editing.

Image Editing Benchmarks. Existing benchmarks can be categorized by the degree of temporal dependency they encode (summarized in Table 6). **(i) Single-turn.** Most early benchmarks target isolated editing instructions [8, 31, 16]. **(ii) Multi-turn.** MagicBrush [35], Seed-Data-Edit [6] provide multi-turn corpora [20], while MSE-Bench [22] and EdiVal-Bench [3] offer evaluations. However, all of them treat each turn as independent task and requires no retention of session-level context. **(iii) Multi-turn in-context.** The need for in-context image-editing evaluation has recently been recognized. [31, 34] establish the long-range dependencies across turns with several human-crafted examples. However, neither work provides a systematic and scalable evaluation protocol.

Reinforcement Learning for Generative Models. Flow-GRPO [13] and Dance-GRPO [30] adopt group relative policy optimization by reformulating the sampling of diffusion and rectified flows via Stochastic Differential Equations (SDEs), and Diffusion-NFT [38] applies direct preference optimization. In image editing, EditScore [17], Edit-R1 [11] employ RL in single-turn settings [32]. Works such as [14, 9] jointly optimize reasoning module and image generator through RL, but remain restricted to single-turn tasks. All of the methods are history-agnostic. In contrast, Edit-R2 is one of the first RL post-training framework for in-context editing.

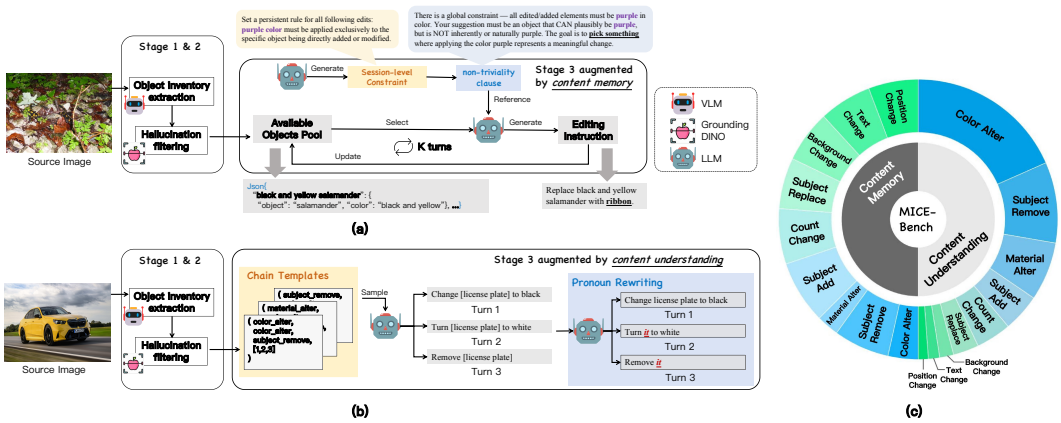


Figure 3: Overview of MICE-Bench. (a)-(b): Data construction pipeline. (c): Task category distribution. Detailed statistics are shown in Table 10.

3 MICE-Bench: Multi-Turn In-Context Editing Benchmark

We introduce **MICE-Bench** (**M**ulti-turn **I**n-**C**ontext **E**diting Benchmark), comprising 720 multi-turn in-context editing instances, each featuring a three-turn editing session. MICE-Bench is specifically designed to evaluate model's capabilities to resolve long-horizon dependencies featuring *Content Understanding* and *Content Memory* tasks (demonstrated in Figure 3).

Base pipeline. We build upon EdiVal-Agent [3], which generates multi-turn editing instructions from unlabeled images in three stages: (1) object inventory extraction via VLM, (2) hallucination filtering with GroundingDINO [15], and (3) object-pool-aware instruction generation that maintains inter-turn coherence (see Appendix E.1). We enhance the pipeline with two in-context tasks.

Content memory task. It tests whether a model can adhere to *session-level constraints* applied uniformly across the editing session (e.g., "all added objects must be *red*"). We inject such constraints into the Stage 3 instruction generator via a constrained LLM prompt, with a *non-triviality clause* to prevent degenerate object proposals that trivially satisfy the constraint. See Appendix E.2 for details.

Content understanding task. It tests whether a model can resolve pronominal coreferences across turns (e.g., "change *its* color"). We apply instruction *chain templates* to ensure consecutive turns share a common anchor object, then perform post-hoc pronoun rewriting via an LLM. This design ensures pronoun references are both well-grounded and naturalistic. See Appendix E.3 for details.

Evaluation metrics. Beyond standard instruction-following (IF) and content-consistency (CC) [3], we introduce **Global-Awareness (GA)**, a new measure specifically designed to quantify the capabilities of resolving in-context challenges. GA evaluates the model's ability to respect session-level constraints and resolve cross-turn coreferences across the editing session, with dedicated evaluation criteria. Full definitions are provided in Appendix E.5.

4 Method

4.1 Formulation: In-context Image Editing as an MDP

We formulate in-context image editing as a finite-horizon Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, R, K)$, where K denotes the number of editing turns. At each turn $t \in \{0, 1, \ldots, K-1\}$, the state $s_t = (Y_0, T_0, Y_1, T_1, \ldots, Y_t, T_t) \in \mathcal{S}$ denotes the full interleaved editing history, where Y_0 is the reference image, $\{Y_i\}_{i=1}^{t}$ are previously generated images, and $\{T_i\}_{i=0}^{t}$ are corresponding instructions. Crucially, s_t collects the entire session history, enabling the policy to reason about long-horizon dependencies. The action $a_t \in \mathcal{A}$ at turn t is the edited image $a_t \triangleq Y_{t+1} = \pi_\theta(s_t)$. At the turn level, we treat the image generation process as a single macro-action, with intra-turn optimization detailed in Section 4.2. The transition function is deterministic: $s_{t+1} = s_t \oplus (Y_{t+1}, T_{t+1})$, concatenating the current state with the new image and the next instruction. A scalar reward $r_t = R(s_t, a_t)$ is received after each turn, reflecting the quality of Y_{t+1}. We define a compound reward as follows:

$$r_t = w_{\text{IF}} \cdot \text{IF}(Y_t, T_t, Y_{t+1}) + w_{\text{CC}} \cdot \text{CC}(Y_0, Y_{t+1}) + w_{\text{GA}} \cdot \text{GA}(s_t, Y_{t+1}), \quad (1)$$

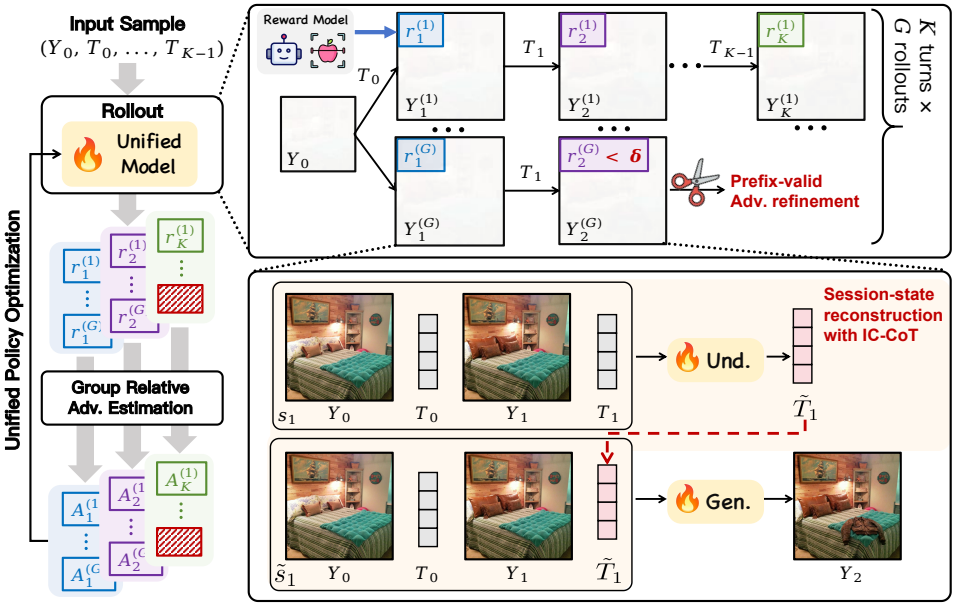


Figure 4: Overview of Edit-R2.

where IF $\in \{0, 1\}$, CC $\in [0, 1]$, and GA $\in \{0, 1\}$ denote instruction-following, content-consistency, and global-awareness metrics, which we directly adopt as reward components for RL training. The objective is to find θ maximizing the expected cumulative reward $\max_\theta \mathbb{E}_{\pi_\theta}[\sum_{t=0}^{K-1} r_t]$.

4.2 Edit-R2

Building on the aforementioned MDP formulation, Edit-R2 targets the key challenge of context loss in multi-turn image editing: the policy must act from a long interleaved history, while the decision-relevant information is a compact but latent set of active session constraints. Edit-R2 therefore learns to reconstruct a compact session state before generation and optimizes both the reconstruction and the visual edit with a unified trajectory-level RL objective. The overview of Edit-R2 is in Figure 4.

Session-state reconstruction with IC-CoT. Long-context attention dilution makes it progressively harder for the image generator to resolve long-horizon dependencies in the session history. Passing the full s_t directly to the generator is a naive approach, but the growing token imbalance and RoPE distance decay cause attention to earlier turns to diminish, rendering significant unfaithful editing at later turns. BAGEL's native reasoning mechanism can strengthen single-turn editing, but is fundamentally inadequate for session-aware reasoning across a historical context. Prior works of advanced CoT reasoning for visual generation [18, 21] similarly restrict to single-turn scenarios, where the reasoning scope is limited to a fixed prompt rather than an evolving session. We therefore introduce **IC-CoT** (In-Context Chain-of-Thought), a session-aware reasoning module that explicitly reasons over the full interleaved history s_t and produces an enhanced instruction $\tilde{T}_t$ before each generation step. By distilling the relevant session context directly into the current instruction, IC-CoT compresses critical historical signals and effectively shortens the distance between the generation module and the information it needs. At each turn t, the understanding expert π_θ^{und} receives the full interleaved context s_t together with a structured question Q_{CoT} (see Appendix F), and produces a CoT trace followed by an enhanced instruction $\tilde{T}_t$:

$$(\tilde{T}_t,\ \mathbf{w}_t,\ \log \pi_\theta(\mathbf{w}_t)) = \pi_\theta^{\text{und}}(s_t, Q_{\text{CoT}}), \tag{2}$$

where $\mathbf{w}_t$ is the CoT token sequence and their log-probabilities $\log \pi_\theta(\mathbf{w}_t)$ is retained for the subsequent policy update. The enhanced $\tilde{T}_t$ replaces the raw T_t, yielding the refined state $\tilde{s}_t = s_{t-1} \oplus (Y_t, \tilde{T}_t)$, which is then passed to the flow-matching generation expert:

$$(Y_{t+1},\ \mathbf{z}_t,\ \log \pi_\theta(\mathbf{z}_t)) = \pi_\theta^{\text{gen}}(\tilde{s}_t), \tag{3}$$

where $\mathbf{z}_t$ denotes the latent trajectory of the flow-matching process.

Algorithm 1 Edit-R2

Input: Reference image Y_0, edit instructions $\{T_t\}_{t=0}^{K-1}$, unified model π_θ, reward function R.
Initialize: $\mathcal{D} \leftarrow \emptyset$; $s_0 \leftarrow (Y_0, T_0)$
for $t \in \{0, 1, \ldots, K-1\}$ **do**
    Model performs IC-CoT $(\tilde{T}_t, \mathbf{w}_t, \log \pi_\theta(\mathbf{w}_t)) = \pi_\theta^{\text{und}}(s_t, Q_{\text{CoT}})$
    Model performs image editing $(Y_{t+1}, \mathbf{z}_t, \log \pi_\theta(\mathbf{z}_t)) = \pi_\theta^{\text{gen}}(\tilde{s}_t)$
    State transition $s_{t+1} = \tilde{s}_t \oplus \big(Y_{t+1},\ T_{t+1}\big)$
    Compute reward $r_t = R(s_t, a_t)$ (Eqn. 1)
    if $r_t < \delta$ **then**
        break
    Collect samples $\tau_t = (\tilde{s}_t, \mathbf{w}_t, \log \pi_\theta(\mathbf{w}_t), \mathbf{z}_t, \log \pi_\theta(\mathbf{z}_t), r_t)$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{\tau_t\}$
Repeat the for-loop above to collect G rollouts
Compute group normalized advantages $\{A_t\}$ from rewards in $\mathcal{D}$ (Eqn. 5)
for each $\tau_t \in \mathcal{D}$ **do**
    Optimize π_θ with $\mathcal{L}_{\text{unified}}(\theta)$ (Eqn. 6).
Return: Updated π_θ

Prefix-valid advantage refinement. During early training, the IC-CoT module may produce incorrect rewrites that misinterpret user intent or hallucinate non-existent content, causing the downstream generation to deviate severely from the intended edit. Continuing a rollout beyond such a failure is harmful: the resulting images carry no meaningful training signal, yet VLM-based reward models are prone to assigning false-positive scores to degenerate outputs, thereby producing noisy gradients in optimization. We therefore terminate rollout i at the first turn where the reward falls below a threshold δ, discarding all subsequent turns. Formally, we define the *active set* at turn t as the collection of rollouts that have not been terminated up to and including turn t:

$$\Omega_t = \{\, i \in \{1, \ldots, G\} \mid r_{t'}^{(i)} \geq \delta,\ \forall\, t' \leq t \,\}. \tag{4}$$

The advantage estimate is then refined to normalize exclusively over active rollouts:

$$A_t^{(i)} = \frac{r_t^{(i)} - \text{mean}_{j \in \Omega_t}\big(r_t^{(j)}\big)}{\text{std}_{j \in \Omega_t}\big(r_t^{(j)}\big)}, \quad i \in \Omega_t. \tag{5}$$

This prevents low-quality trajectories from distorting the normalization baseline, and reduces unnecessary computation by avoiding generation for rollouts that have already diverged.

Unified multi-turn policy optimization. IC-CoT and the flow-matching generator are tightly coupled, as the IC-CoT trace determines a reconstructed session state, and the generator subsequently realizes a visual edit conditioned on that state. However, the training signal is only observed after image generation: the quality of a CoT rewrite is only observable through the downstream image it produces, yet the generator can only benefit from IC-CoT if the rewrites are well-calibrated to its generation capabilities. Training them in isolation with separate objectives would break this feedback loop. We therefore treat each turn as a joint decision of the IC-CoT token sequence and the flow trajectory for effective edits to reinforce not only the visual trajectory that produced the image, but also the preceding session reconstruction that made the edit possible. Given this factorization, we instantiate the policy updated with Unified-GRPO [14] and extends it to the multi-turn scenario by tackling the core challenge of context loss. The pseudo-code for Edit-R2 is in Algorithm 1. Refer to Appendix F for detailed training loss.

5 Experiments

5.1 Setup

Evaluation benchmarks and metrics. We primarily evaluate on **MICE-Bench**, which measures models' ability to resolve long-horizon challenges in in-context editing, with IF, CC, and GA as evaluation metrics. We report *marginal task success rate* for IF and GA: For a given turn, the proportion of previously successful prompts that continue to succeed at current turn. Results are reported in the average over 4 runs. We also report results on EdiVal-Bench [3] and GEdit-Bench [16] for generalization capability (see Appendix C).

Table 1: Results on MICE-Bench. Avg = (IF + CC + GA) / 3. Best and second-best results among open-source models are in **bold** and underlined, respectively. Closed-source model scores that fall below Edit-R2 are marked in red. †Models not originally support in-context editing.

Method	Turn 1				Turn 2				Turn 3			
	IF	CC	GA	Avg	IF	CC	GA	Avg	IF	CC	GA	Avg
Closed-source models												
GPT-Image-1 [high] [19]	66.31	75.22	61.29	67.61	60.42	68.10	59.93	62.82	62.06	60.55	52.76	58.46
Nano-Banana-2 [4]	62.65	87.38	59.97	70.00	49.53	82.55	46.31	59.46	57.98	76.37	52.43	62.26
Open-source models												
Qwen-Image-Edit [2511]† [26]	**58.62**	81.20	**52.84**	64.22	46.47	77.58	46.50	56.85	25.04	72.46	20.76	39.42
Flux.1-Kontext [dev]† [10]	43.21	73.25	38.28	51.58	32.56	71.51	32.63	45.57	33.90	68.53	31.87	44.77
OmniGen [29]	44.44	56.84	38.11	46.46	37.51	50.44	32.69	40.21	40.97	47.41	36.76	41.71
VINCIE [22]	34.77	71.51	31.67	45.98	34.68	63.37	31.92	43.32	43.78	58.02	26.90	42.90
BAGEL [5]	51.38	**89.12**	44.72	61.74	42.12	**85.34**	39.56	55.67	50.18	**79.90**	43.59	57.89
BAGEL (w/ think)	45.90	88.39	42.37	58.89	43.66	83.58	40.03	55.76	47.80	77.88	48.14	57.94
BAGEL (w/ IC-CoT)	51.88	87.80	46.30	62.00	46.84	83.82	46.07	58.91	50.82	78.21	46.79	58.61
Edit-R2	54.82	88.62	50.73	**64.72**	**60.77**	84.33	**58.17**	**67.76**	**58.63**	78.51	**58.96**	**65.37**

Table 2: Ablation study of Edit-R2 on MICE-Bench. Best results are in **bold**. TG = Text-GRPO; PV = Prefix-valid advantage refinement.

Method	Turn 1				Turn 2				Turn 3			
	IF	CC	GA	Avg	IF	CC	GA	Avg	IF	CC	GA	Avg
Edit-R2 (w/o IC-CoT, TG, PV)	53.99	88.29	50.06	64.11	49.32	82.45	46.28	59.35	55.20	72.26	52.01	59.82
Edit-R2 (w/o TG, PV)	52.65	**88.86**	47.05	62.85	51.53	**84.72**	51.37	62.54	56.50	**78.99**	55.76	63.75
Edit-R2 (w/o PV)	54.49	88.77	50.56	64.61	55.31	84.36	53.42	64.36	58.08	77.95	57.09	64.71
Edit-R2	**54.82**	88.62	**50.73**	**64.72**	**60.77**	84.33	**58.17**	**67.76**	**58.63**	78.51	**58.96**	**65.37**

Baselines and training details. We compare against closed-source models include Nano-Banana-2 [4] and GPT-Image-1 [high] [19]. For open-source models, we consider those with native in-context editing support, including OmniGen [29], VINCIE [22], BAGEL [5] and its variants. We also evaluate editing-specific models Qwen-Image-Edit [26] and Flux.1 Kontext [10]. Since they do not inherently support in-context editing, we construct their input for each turn t by concatenating $(Y_0, T_0, \ldots, T_t)$ to approximate the in-context setting for fair comparison. We build Edit-R2 upon BAGEL [5] and fine-tune with LoRA. We use a training set comprising 10K in-context editing instances, generated via the same pipeline as MICE-Bench. The prefix-valid advantage refinement threshold δ is set to 0.5. Full details including hyperparameters are provided in Appendix A.

5.2 Main Results

We summarize key observations for results on MICE-Bench (see Table 1). **Edit-R2 achieves the best average performance across all turns among open-source models.** On Turn 1, Edit-R2 performs competitively among models with native in-context editing support. We note that Qwen-Image-Edit demonstrates strong single-turn instruction following capabilities, reflecting its specialization in this setting. **On the more challenging Turn 2 and Turn 3, Edit-R2 outperforms all models on average.** Qwen-Image-Edit and Flux.1-Kontext suffer severe performance degradation due to their inability to leverage the full session history. In contrast, Edit-R2 improves substantially over BAGEL, with IF and GA gains of approximately **(+18%, +18%)** at Turn 2 and **(+8%,+15%)** at Turn 3. These results demonstrate Edit-R2's strong performance retention on long-horizon editing tasks. Notably, GPT-Image-1 exhibits a pronounced CC decline across turns. As qualitative analysis reveals in Section 5.5, this reflects a tendency toward free-form generation with limited pixel fidelity.

5.3 Ablation Study

Table 2 reports the incremental contribution of each component in Edit-R2. The base variant without IC-CoT, TG, and PV establishes a strong foundation via Flow-GRPO alone, yet its gains are primarily concentrated at Turn 1 and diminish at later turns where the long-context challenge becomes severe. Incorporating IC-CoT yields consistent improvements at Turn 2 and Turn 3, as session-aware instruction reconstruction reduces the reasoning burden on the flow-matching module and mitigates context dilution across turns. Further incorporating Text-GRPO—which jointly optimizes the IC-CoT module under the same reward signal—delivers additional gains at later turns, confirming that explicitly optimizing the rewriting policy amplifies its effectiveness (see Section 5.4 for analysis).

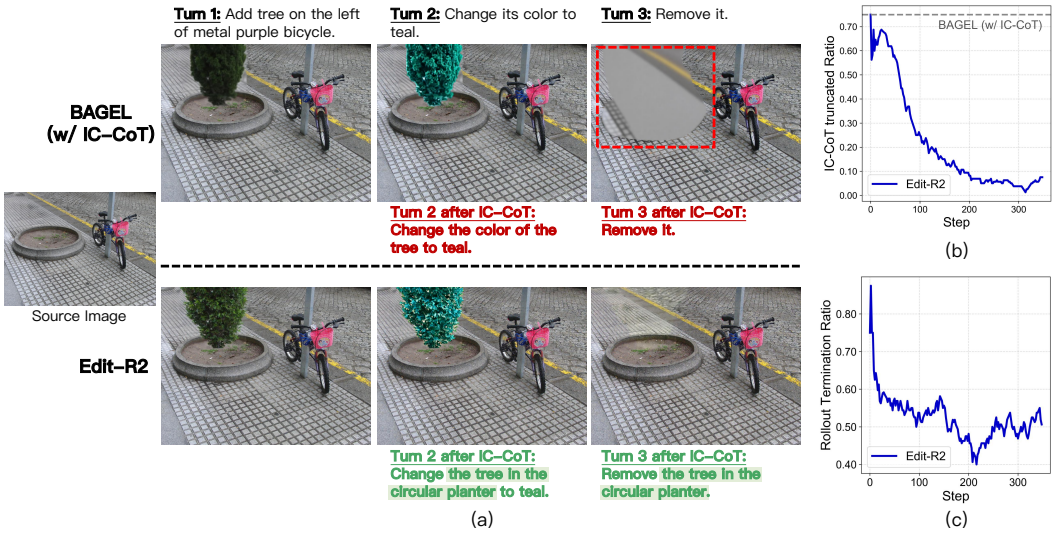


Figure 5: (a) Example of emergent IC-CoT behavior. After Edit-R2 training, IC-CoT produces more precise session-aware instruction rewrites (e.g., enriching "the tree" with spatial context "in the circular planter"). (b)-(c): Both the IC-CoT truncated ratio and rollout termination ratio decline steadily across training steps, indicating that Edit-R2 progressively stabilizes IC-CoT and reduces error propagation across turns.

Finally, PV prevents erroneous intermediate edits from corrupting the normalization baseline and propagating noise into later turns, achieving the best overall performance.

5.4 In-depth Analysis

Emergent instruction rewriting via Edit-R2 training. Following Edit-R2 training, IC-CoT exhibits emergent behavior (Figure 5(a)): it accurately identifies the target object and enriches its spatial description (e.g., "in the circular planter"), enabling more precise editing. In contrast, the untrained IC-CoT generates excessive redundant reasoning, leading to output truncation and failure to complete the instruction rewriting task, resulting in poor visual quality. As training evolves, both the IC-CoT truncated ratio and rollout-termination ratio decline steadily (Figure 5(b)(c)), confirming that Edit-R2 effectively stabilizes IC-CoT generation and validates the role of prefix-valid advantage refinement in preventing corrupted trajectories from distorting policy optimization.

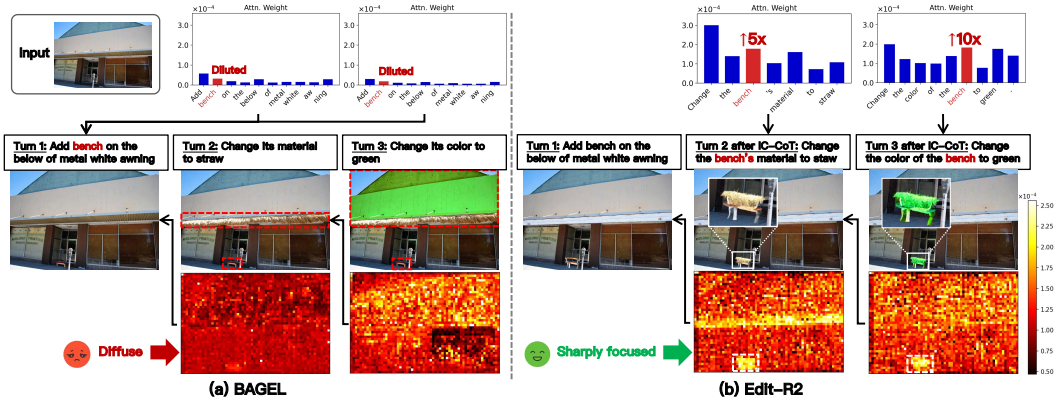


Figure 6: Attention visualization comparing BAGEL and Edit-R2. BAGEL gets lost in the long interleaved history: attention to critical tokens from earlier turns is weakened, resulting in misdirected and diffuse visual attention that leads to erroneous edits. In contrast, Edit-R2 accurately captures the operative session intent (attending $5\times$ and $10\times$ more strongly to the key object token "bench" at turns 2 and 3 respectively) and produces spatially focused, faithful edits.

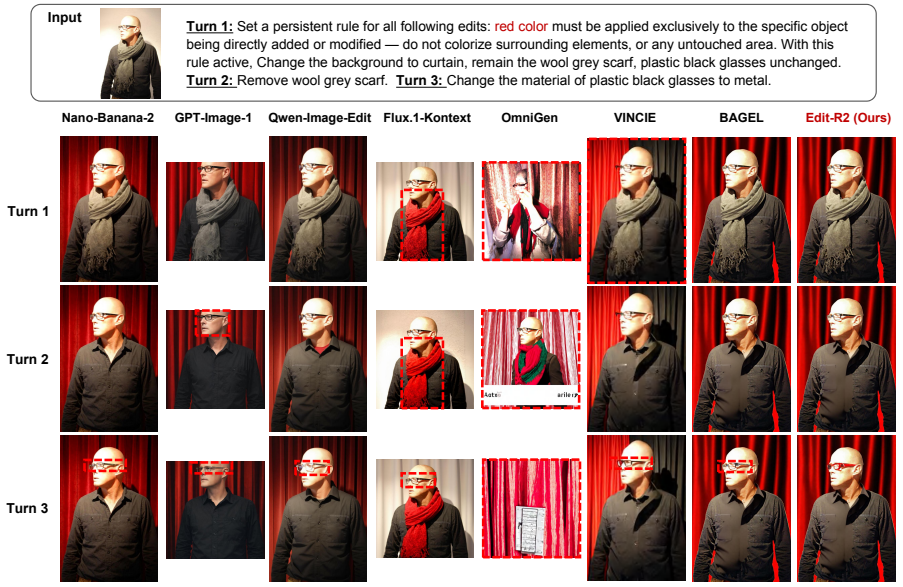


Figure 7: Qualitative comparison on *Content Memory* task. Erroneous areas are in red boxes.

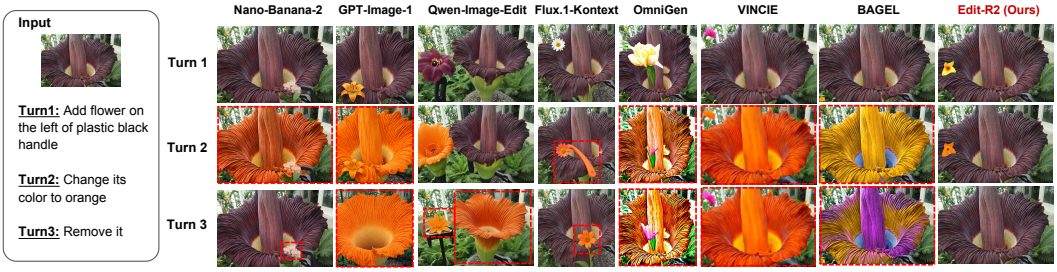


Figure 8: Qualitative comparison on *Content Understanding* task. Erroneous areas are in red boxes.

Attention analysis: Edit-R2 recovers session intent lost by BAGEL. As shown in Figure 6, we visualize text attention at the middle layer (where object-level semantic grounding is observed) and visual attention (over VAE tokens) at the last transformer layer during the velocity-prediction forward pass. BAGEL often fails to localize the intended editing region, with misdirected and diffuse visual attention, consistent with its weak text attention on key object tokens from earlier turns where RoPE distance decay substantially weakens the signal. In contrast, Edit-R2 exhibits sharply focused visual attention on target regions, empowered by IC-CoT which distills critical session context into the enhanced instruction. These observations align with a text-to-visual grounding mechanism: intermediate-layer textual anchoring enriches the query with object-level semantics, which in turn guides precise spatial localization in later visual attention layers.

5.5 Qualitative Results

Figure 7 presents qualitative results on the content memory task. All other models completely ignore the session-level constraint "red color must be applied" at turn 3. In contrast, Edit-R2 successfully adheres to the constraint and changes the color of glasses to red, demonstrating improved long-horizon constraint awareness. Figure 8 demonstrates the content understanding task. All other models fail to correctly recolor the flower added in the previous edit to the user-specified color at turn 2, instead altering the color of unrelated regions. They also fail to correctly remove it at turn 3. Please refer to Appendix D for more qualitative results.

5.6 Generalization to Longer Editing Sessions

To evaluate whether the gains of Edit-R2 extend beyond the three-turn training horizon, we construct an additional evaluation set of 200 six-turn editing sessions (100 Content Memory and 100 Content

Table 3: Longer-horizon evaluation on six-turn editing sessions. Models are evaluated at Turns 4-6 using the same protocol as Table 1. Best results are in **bold**.

Method	Turn 4 Avg	Turn 5 Avg	Turn 6 Avg
BAGEL	36.65	38.36	28.47
Edit-R2 (3-turn training)	47.84	47.21	30.52
Edit-R2 (6-turn continual training)	**51.57**	**51.16**	**33.39**

Understanding sessions) following the MICE-Bench construction protocol. Table 3 reports the average performances over IF,CC and GA at the three unseen turns. The original Edit-R2 checkpoint, trained exclusively on three-turn sessions, consistently outperforms BAGEL at Turns 4-6. This demonstrates that the learned policy transfers beyond its training horizon, although the smaller gain at Turn 6 indicates that degradation over very long histories remains an open challenge. We further curate 2,000 six-turn training samples and continue training the three-turn checkpoint. This further improves the performances at later turns, showing Edit-R2's potential to be extended naturally to longer trajectories.

5.7 Impact of Reward Weights

We study the roles of instruction following (IF), content consistency (CC), and global awareness (GA) by varying the GA reward weight and distributing the remaining weight equally between IF and CC. As shown in Table 4, increasing the GA weight improves GA while slightly reducing IF and CC; decreasing it produces the opposite trend. Across nonzero GA weights, the average score changes only modestly, supporting equal weighting as a simple and balanced default. In contrast, removing GA in reward reduces GA performance by 9.24 and 10.22 at Turns 2 and 3, respectively, compared with the default equal-weight setting, while yielding only marginal improvements in IF and CC. The resulting average-score drops of 2.32 and 2.81 points demonstrate that GA provides a non-redundant signal for preserving session-level constraints.

Table 4: Reward-weight ablation on MICE-Bench.

	Turn 1				Turn 2				Turn 3			
Reward weights	IF	CC	GA	Avg	IF	CC	GA	Avg	IF	CC	GA	Avg
$0.2\,\mathrm{IF} + 0.2\,\mathrm{CC} + 0.6\,\mathrm{GA}$	53.43	87.95	**52.10**	64.49	58.93	83.66	**59.95**	67.51	56.73	77.95	**60.55**	65.08
$\frac{1}{3}\,\mathrm{IF} + \frac{1}{3}\,\mathrm{CC} + \frac{1}{3}\,\mathrm{GA}$	54.82	88.62	50.73	64.72	60.77	84.33	58.17	**67.76**	58.63	78.51	58.96	**65.37**
$0.4\,\mathrm{IF} + 0.4\,\mathrm{CC} + 0.2\,\mathrm{GA}$	56.10	89.29	49.28	**64.89**	61.58	84.95	56.20	67.58	59.42	79.04	56.95	65.14
$0.5\,\mathrm{IF} + 0.5\,\mathrm{CC}$	**57.62**	**89.74**	47.05	64.80	**62.09**	**85.31**	48.93	65.44	**59.70**	**79.26**	48.74	62.56

6 Conclusion

We investigate multi-turn in-context image editing, a practical yet underexplored paradigm where the model must maintain session-level consistency across an evolving dialogue history. We propose **Edit-R2**, comprising IC-CoT for session intent reconstruction and unified multi-turn RL that jointly optimizes reasoning and image generation, alongside **MICE-Bench** for systematic evaluation. Edit-R2 yields substantial gains and achieves competitive performance against strong models.

Limitations. MICE-Bench covers two representative in-context editing scenarios. It is interesting for future directions to further capture the breadth of user intent in real-world interactive editing based on MICE-Bench, scale Edit-R2 to explore a broader taxonomy of in-context editing task categories.

Acknowledgments

This work was supported by KlingAI.

References

[1] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[2] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18392–18402, 2023.

[3] Tianyu Chen, Yasi Zhang, Zhi Zhang, Peiyu Yu, Shu Wang, Zhendong Wang, Kevin Lin, Xiaofei Wang, Zhengyuan Yang, Linjie Li, et al. Edival-agent: An object-centric framework for automated, fine-grained evaluation of multi-turn editing. *arXiv preprint arXiv:2509.13399*, 2025.

[4] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, 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.

[5] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

[6] Yuying Ge, Sijie Zhao, Chen Li, Yixiao Ge, and Ying Shan. Seed-data-edit technical report: A hybrid dataset for instructional image editing. *arXiv preprint arXiv:2405.04007*, 2024.

[7] Runze He, Yiji Cheng, Tiankai Hang, Zhimin Li, Yu Xu, Zijin Yin, Shiyi Zhang, Wenxun Dai, Penghui Du, Ao Ma, et al. Re-align: Structured reasoning-guided alignment for in-context image generation and editing. *arXiv preprint arXiv:2601.05124*, 2026.

[8] Mude Hui, Siwei Yang, Bingchen Zhao, Yichun Shi, Heng Wang, Peng Wang, Yuyin Zhou, and Cihang Xie. Hq-edit: A high-quality dataset for instruction-based image editing. *arXiv preprint arXiv:2404.09990*, 2024.

[9] Siqi Kou, Jiachun Jin, Zetong Zhou, Ye Ma, Yugang Wang, Quan Chen, Peng Jiang, Xiao Yang, Jun Zhu, Kai Yu, et al. Think-then-generate: Reasoning-aware text-to-image diffusion with llm encoders. *arXiv preprint arXiv:2601.10332*, 2026.

[10] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

[11] Zongjian Li, Zheyuan Liu, Qihui Zhang, Bin Lin, Feize Wu, Shenghai Yuan, Zhiyuan Yan, Yang Ye, Wangbo Yu, Yuwei Niu, et al. Uniworld-v2: Reinforce image editing with diffusion negative-aware finetuning and mllm implicit feedback. *arXiv preprint arXiv:2510.16888*, 2025.

[12] Yucheng Liao, Jiajun Liang, Kaiqian Cui, Baoquan Zhao, Haoran Xie, Wei Liu, Qing Li, and Xudong Mao. Freqedit: Preserving high-frequency features for robust multi-turn image editing. *arXiv preprint arXiv:2512.01755*, 2025.

[13] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025.

[14] Jie Liu, Zilyu Ye, Linxiao Yuan, Shenhan Zhu, Yu Gao, Jie Wu, Kunchang Li, Xionghui Wang, Xiaonan Nie, Weilin Huang, et al. Unigrpo: Unified policy optimization for reasoning-driven visual generation. *arXiv preprint arXiv:2603.23500*, 2026.

[15] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024.

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

[17] Xin Luo, Jiahao Wang, Chenyuan Wu, Shitao Xiao, Xiyan Jiang, Defu Lian, Jiajun Zhang, Dong Liu, et al. Editscore: Unlocking online rl for image editing via high-fidelity reward modeling. *arXiv preprint arXiv:2509.23909*, 2025.

[18] Yuanhuiyi Lyu, Chi Kit Wong, Chenfei Liao, Lutao Jiang, Xu Zheng, Zexin Lu, Linfeng Zhang, and Xuming Hu. Understanding-in-generation: Reinforcing generative capability of unified model via infusing understanding into generation. *arXiv preprint arXiv:2509.18639*, 2025.

[19] OpenAI. Introducing 4o image generation. `https://openai.com/index/introducing-4o-image-generation/`, 2025. Accessed: 2025-06-22.

[20] Yusu Qian, Eli Bocek-Rivele, Liangchen Song, Jialing Tong, Yinfei Yang, Jiasen Lu, Wenze Hu, and Zhe Gan. Pico-banana-400k: A large-scale dataset for text-guided image editing. *arXiv preprint arXiv:2510.19808*, 2025.

[21] Luozheng Qin, Jia Gong, Yuqing Sun, Tianjiao Li, Mengping Yang, Xiaomeng Yang, Chao Qu, Zhiyu Tan, and Hao Li. Uni-cot: Towards unified chain-of-thought reasoning across text and vision. *arXiv preprint arXiv:2508.05606*, 2025.

[22] Leigang Qu, Feng Cheng, Ziyan Yang, Qi Zhao, Shanchuan Lin, Yichun Shi, Yicong Li, Wenjie Wang, Tat-Seng Chua, and Lu Jiang. Vincie: Unlocking in-context image editing from video. In *The Fourteenth International Conference on Learning Representations*, 2025.

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

[24] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

[25] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features, 2025.

[26] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

[27] Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, et al. Omnigen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025.

[28] Keming Wu, Zuhao Yang, Kaichen Zhang, Shizun Wang, Haowei Zhu, Sicong Leng, Zhongyu Yang, Qijie Wang, Sudong Wang, Ziting Wang, Zili Wang, Hui Zhang, Haonan Wang, Hang Zhou, Yifan Pu, Xingxuan Li, Fangneng Zhan, Bo Li, Lidong Bing, Yuxin Song, Ziwei Liu, Wenhu Chen, Jingdong Wang, Xinchao Wang, Xiaojuan Qi, Shijian Lu, and Bin Wang. Visual generation in the new era: An evolution from atomic mapping to agentic world modeling, 2026.

[29] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13294–13304, 2025.

[30] Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

[31] Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. Imgedit: A unified image editing dataset and benchmark. *arXiv preprint arXiv:2505.20275*, 2025.

[32] Zixuan Ye, Quande Liu, Cong Wei, Yuanxing Zhang, Xintao Wang, Pengfei Wan, Kun Gai, and Wenhan Luo. Visual-aware cot: Achieving high-fidelity visual consistency in unified models. *arXiv preprint arXiv:2512.19686*, 2025.

[33] Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Anyedit: Mastering unified high-quality image editing for any idea. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26125–26135, 2025.

[34] Haochen Zhang, Animesh Sinha, Felix Juefei-Xu, Haoyu Ma, Kunpeng Li, Zhipeng Fan, Meng Dong, Xiaoliang Dai, Tingbo Hou, Peizhao Zhang, et al. Non-markov multi-round conversational image generation with history-conditioned mllms. *arXiv preprint arXiv:2601.20911*, 2026.

[35] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36:31428–31449, 2023.

[36] Zechuan Zhang, Ji Xie, Yu Lu, Zongxin Yang, and Yi Yang. Enabling instructional image editing with in-context generation in large scale diffusion transformer. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[37] Haozhe Zhao, Xiaojian Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. Ultraedit: Instruction-based fine-grained image editing at scale. *Advances in Neural Information Processing Systems*, 37:3058–3093, 2024.

[38] Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. Diffusionnft: Online diffusion reinforcement with forward process. *arXiv preprint arXiv:2509.16117*, 2025.

A Implementation Details

Details for experimental setup. During evaluation, both IF and GA employ Qwen3-VL-235B-A22B-Instruct for more aligned evaluation. Refer to Appendix E.5 for human agreement study of judgers. During training, we adopt a lightweight Qwen3-VL-32B-Instruct for IF and GA to accelerate online RL training. We further employ a self-ensemble strategy over the reward model to stabilize the reward signal, and apply KL regularization to prevent reward hacking. Refer to Appendix C for detailed analysis.

Hyper-parameters. We fine-tune BAGEL using LoRA with rank 64 and scaling factor 128. LoRA adapters are applied to all attention projections and MLP layers in both the generation expert and the understanding expert, as the unified training objective (Section 4.2) requires gradient flow through both experts. We optimize with AdamW at a learning rate of 10^{-4}. For GRPO, we sample a group of G=6 images per prompt from a shared initial noise. Training is distributed across 4×6 NVIDIA H200 GPUs with a total batch size of 48. On each node, 2 GPUs are reserved for deploying the reward models. We apply policy ratio clipping with threshold ϵ=10^{-5} and KL regularization with coefficient β=0.01 against the frozen reference policy. The composite reward combines instruction following (w_{IF}=0.33), content consistency (w_{CC}=0.33), and global awareness (w_{GA}=0.33). Each training episode spans $K = 3$ editing turns. During rollout, we use 15 denoising steps with classifier-free guidance scales of 4.0 (text) and 2.0 (image). Table 5 summarizes the key hyperparameters.

Incremental KV-Cache Update. BAGEL naturally supports text-image interleaved history reuse via KV-Cache. During sequential editing rollout, the contextual history of preceding turns is restored from the KV-Cache at each turn. Upon completion of the current editing step, the result is incrementally updated into the KV-Cache, i.e., $s_{t+1} = \tilde{s}_t \oplus \big(Y_{t+1},\ T_{t+1}\big)$. This native reuse mechanism eliminates the need to reconstruct the full context from scratch at every turn, which constitutes a key motivation for adopting BAGEL as the base model of Edit-R2.

Table 5: Summary of training hyperparameters.

Hyperparameter	**Value**
Training	
LoRA rank / alpha	64 / 128
Optimizer	AdamW
Learning rate	1×10^{-4}
Group size	6
Total train batch size	48
Policy clip threshold (ϵ)	1×10^{-5}
KL coefficient (β)	0.01
Reward	
w_{IF} / w_{CC} / w_{GA}	0.33 / 0.33 / 0.33
Prefix-valid advantage threshold (δ)	0.5
Sampling & Generation	
Editing turns per episode (K)	3
Denoising steps (train / eval)	15 / 50
CFG scale (text / image)	4.0 / 2.0

B Additional Related Work

Comparison of MICE-Bench and existing benchmarks for image editing are summarized in Table 6. "In-Context" denotes whether the benchmark requires retention of session-level constraints across turns, beyond the immediately preceding image and instruction.

C Additional Experimental Results

Results on the EdiVal-Bench. To verify the generality of our approach beyond the in-context editing setting, we further evaluate all models on the EdiVal-Bench [3] that targets history-agnostic multi-turn editing. Results are reported in Table 7.

Table 6: Comparison of benchmarks for image editing evaluation.

Benchmark	#Size	#Sub-Tasks	Automated	Multi-Turn	In-Context
GEdit-Bench [16]	606	11	✗	✗	✗
ImgEdit-Bench [31]	811	14	✓	✗	✗
MagicBrush [35]	1053	5	✗	✓	✗
MSE-Bench [22]	100	13	✗	✓	✗
EdiVal-Bench [3]	572	9	✓	✓	✗
MICE-Bench (ours)	720	9	✓	✓	✓

Table 7: Results on the EdiVal-Bench. IF = EdiVal-IF, CC = EdiVal-CC. Avg = per-turn average of (IF + CC) / 2.

	Turn 1			Turn 2			Turn 3		
Method	IF	CC	Avg	IF	CC	Avg	IF	CC	Avg
BAGEL [5]	56.47	**88.38**	72.43	33.39	**79.29**	56.34	16.43	**72.95**	44.69
Edit-R2	**57.99**	88.10	**73.05**	**35.14**	78.53	**56.84**	**18.53**	72.04	**45.29**

Results on GEdit-Bench. Table 8 reports evaluation results on the GEdit-Bench [16] benchmark. Notably, Edit-R2 also improves on EdiVal-Bench and GEdit-Bench, indicating that the learned capabilities generalize beyond the training distribution.

Training Dynamics. Learning curves of Edit-R2 are demonstrated in Figure 9. Edit-R2 enables a steady training at each turn.

Ablations on reward weights. Since we focus on a multiple-reward objective, determining the weight of each reward is crucial. As CC measures the similarity between the input image and the edited image, we observe that incorporating CC in the overall reward can implicitly regularize the model and prevent quality degradation. Specifically, we compare the output images generated under $r = 0.33 * \text{IF} + 0.33 * \text{CC} + 0.33 * \text{GA}$, and $r = 0.5 * \text{IF} + 0.5 * \text{GA}$. As shown in Figure 10, CC helps prevent the generation of blurry and noise images, effectively mitigating the degradation of image quality.

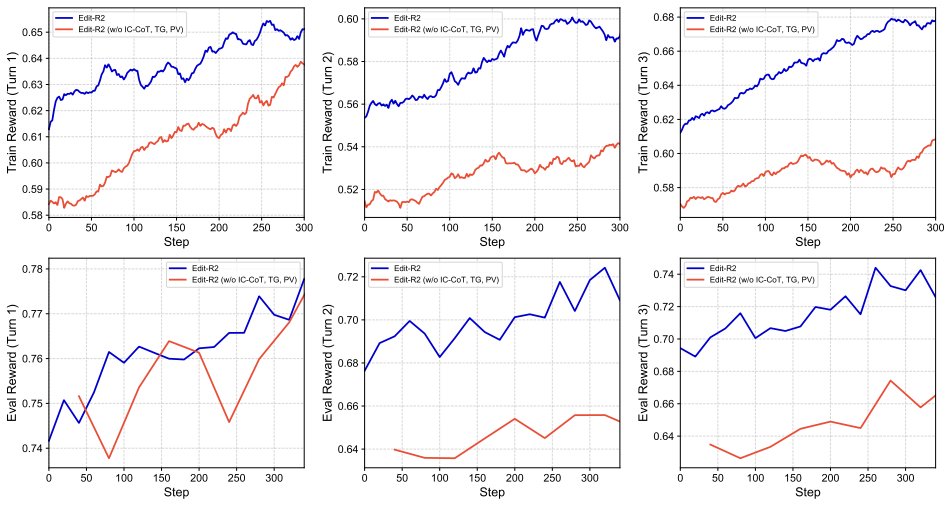


Figure 9: Training dynamics. Average rewards of IF, GA and CC are reported at each turn. Compared to the variant trained with Flow-GRPO alone, full Edit-R2 achieves more pronounced gains at turns 2 and 3, driven by IC-CoT and prefix-valid advantage refinement.

Table 8: Results on GEdit-Bench-EN (Full set). G_SC = Semantic Consistency, G_PQ = Perceptual Quality, G_O = Overall Score.

Method	G_SC	G_PQ	G_O
BAGEL [5]	7.36	6.83	6.52
Edit-R2	**7.55**	**6.99**	**6.75**



Figure 10: Ablations on reward weights. Incorporating CC in reward significantly mitigating the degradation of image quality.

Ablations on self-ensembled reward model. In our initial experimental attempts, we observed that since IF and GA both rely on a VLM for scoring, the resulting reward signals inevitably carry non-trivial noise and cause training to be highly unstable, consistent with observations in prior work [17]. To address this, we adopt a self-ensemble strategy: the VLM reward model replaces greedy decoding by stochastic sampling with temperature 0.6, and averages the scores from four independent forward passes to obtain the final reward signal. Figure 11 presents training curves in the single-reward setting ($r = \text{IF}$). Even in this comparatively simple configuration, a greedy-sampling VLM used as IF fails to achieve effective training at turn 1: performance degrades monotonically throughout training. In contrast, the model trained with the Avg@4 self-ensemble reward achieves steady and consistent improvement. Consequently, all experiments in this paper adopt the Avg@4 self-ensemble strategy as the default configuration for both IF and GA.

D Additional Qualitative Results

Additional qualitative results are demonstrated in Figure 12 and Figure 13.

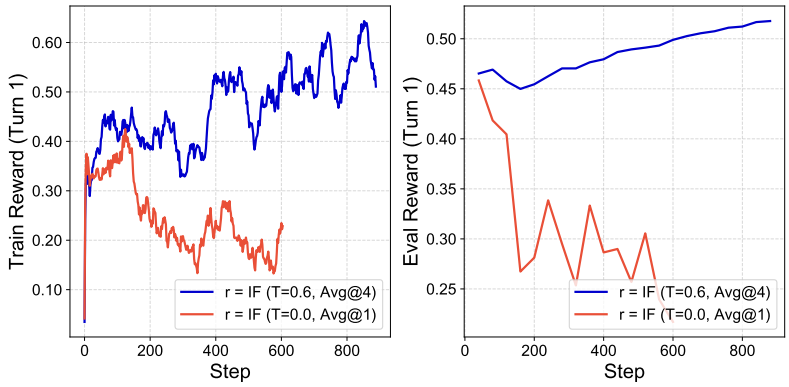


Figure 11: Ablations on self-ensembled reward model IF. T denotes VLM decoding temperature; Avg@4 denotes the average score over 4 forward passes.

E Details for MICE-Bench

E.1 Base Pipeline Details

E.1.1 Stage 1: Object Inventory Extraction

Given an unlabeled input image, we prompt a VLM to enumerate all clearly visible objects. For each object, the VLM describes its semantic type, dominant color, material, visible text, instance count, and foreground/background status. The output format is a structured JSON record:

- **Keys** follow the template `"{material} {color} {object}"` (e.g., `"wooden brown table"`).
- **Values** encode the attribute fields listed above.
- A dedicated `"All Objects"` field aggregates the full inventory into a single string for downstream consumption.

This structured representation constitutes the initial *object pool* $\mathcal{P}_0$ that grounds all subsequent instruction generation.

E.1.2 Stage 2: Hallucination Filtering

VLMs are known to hallucinate objects absent from the image [3]. We pass each candidate object name through GroundingDINO [15], an open-set object detector, using the object description as the text query. An object is retained only if at least one bounding box exceeds a confidence threshold. We additionally filter out detections whose bounding-box area exceeds 90% of the image (normalized coordinates) to suppress spurious full-image matches. Each retained object is augmented with the number of detected instances and bounding-box coordinates for subsequent evaluation.

E.1.3 Stage 3: Object-Pool-Aware Instruction Generation

Given the verified object pool $\mathcal{P}_0$, an LLM generates a sequence of editing instructions over $K = 3$ turns. At each turn, the LLM samples from a predefined taxonomy of edit types shown in Table 10.

The key mechanism is the dynamically maintained object pool $\mathcal{P}_t$, updated after each turn to reflect the expected scene state:

- **Removal / Replacement.** The target object is deleted from $\mathcal{P}_t$; for replacements, the new object is inserted with specified attributes.
- **Addition.** The new object is inserted with default attributes.
- **Attribute modification.** The object entry is updated (e.g., color or material), and the object key is renamed accordingly so that subsequent instructions reference the modified entity.

By enforcing that every instruction at turn $t+1$ only references objects in $\mathcal{P}_t$, the pipeline guarantees that no instruction can, for example, alter the color of an object removed in a prior turn. This eliminates contradictory sequences and ensures logical coherence.

E.2 Details for Content Memory Task

E.2.1 Session-Level Constraint Hints

Below we reproduce the full prompt templates used by the instruction-generation LLM when a session-level global constraint is active. Each template is injected into the corresponding edit-type prompt (e.g., *subject addition*, *subject replacement*, *background change*). The placeholder `{value}` is instantiated with the sampled constraint value at runtime (e.g., "brown" for color, "fluffy" for material).

Color constraint hint.

```
IMPORTANT: There is a global constraint -- all edited/added elements
must be ‘‘{value}’’ in color.  Your suggestion must be an object
that CAN plausibly be {value}, but is NOT inherently or naturally
{value}.  For example, if the constraint color is ‘‘brown’’, do NOT
suggest objects like ‘‘wood’’, ‘‘log’’, ‘‘dirt’’, or ‘‘chocolate’’ that
are naturally brown -- instead suggest objects like ‘‘vase’’, ‘‘hat’’,
‘‘chair’’, or ‘‘umbrella’’ that can be {value} but are not typically
associated with that color.  The goal is to pick something where
applying the color ‘‘{value}’’ represents a meaningful change.
```

Material constraint hint.

```
IMPORTANT: There is a global constraint -- all edited/added elements
must use ‘‘{value}’’ material.  Your suggestion must be an object
that CAN plausibly be made of {value}, but is NOT inherently or
naturally made of {value}.  For example, if the constraint material
is ‘‘fluffy’’, do NOT suggest objects like ‘‘teddy bear’’, ‘‘pillow’’, or
‘‘cotton candy’’ that are naturally fluffy -- instead suggest objects
like ‘‘chair’’, ‘‘lamp’’, or ‘‘vase’’ that can be made fluffy but are
not typically associated with that material.  The goal is to pick
something where applying ‘‘{value}’’ material represents a meaningful
transformation.
```

Example: subject addition prompt with constraint hint. The following shows the complete prompt sent to the LLM for a *subject addition* task when a color constraint ($v =$ brown) is active:

```
Suggest an object to add {position} of {reference_object}.

IMPORTANT: There is a global constraint -- all edited/added elements
must be ‘‘brown’’ in color.  Your suggestion must be an object
that CAN plausibly be brown, but is NOT inherently or naturally
brown.  For example, if the constraint color is ‘‘brown’’, do NOT
suggest objects like ‘‘wood’’, ‘‘log’’, ‘‘dirt’’, or ‘‘chocolate’’ that
are naturally brown -- instead suggest objects like ‘‘vase’’, ‘‘hat’’,
‘‘chair’’, or ‘‘umbrella’’ that can be brown but are not typically
associated with that color.  The goal is to pick something where
applying the color ‘‘brown’’ represents a meaningful change.

Respond with ONLY the object name.
```

E.2.2 Non-Triviality Design Rationale

A naïve constraint prompt (e.g., simply stating "all objects must be brown") causes the LLM to propose objects whose inherent nature already matches the target attribute—"log," "wood," or "beaver" when brown is specified—rendering the constraint vacuous as a test of editing ability. The non-triviality clause explicitly instructs the LLM to avoid such trivial associations, ensuring the constraint genuinely challenges the editing model rather than being satisfied by the original scene content.

E.2.3 Constraint-Aware Operation Suppression

When a session-level constraint specifies a particular color or material, the corresponding *color alter* or *material alter* operations are suppressed in subsequent turns to prevent contradictory instructions (e.g., a global "red" constraint followed by "change the cup to blue").

E.3 Details for Content Understanding Task

E.3.1 Instruction Chain Templates

We curate a set of templates, each specifying a fixed sequence of three edit types and a same-object constraint indicating which turns share a common anchor object. Representative templates include:

- $(\textit{color alter},\ \textit{color alter},\ \textit{subject remove},\ [1, 2, 3])$: all three turns target the same entity.
- $(\textit{subject add},\ \textit{color alter},\ \textit{subject replace},\ [1, 2])$: only the first two turns share an anchor.
- $(\textit{material alter},\ \textit{position change},\ \textit{color alter},\ [2, 3])$: turns 2 and 3 share an anchor.

Templates with partial overlap increase sample diversity while still guaranteeing a clear referent for pronoun substitution.

E.3.2 Pronoun Rewriting Examples

Given a three-turn sequence with explicit object references, an LLM rewrites turns 2 and 3 (turn 1 is unchanged, lacking preceding context). The rewriting prompt includes the original image, all three instructions, and anchor-object hints identifying the primary operand per turn. The LLM applies the following substitution rules:

- `it` / `them`: used when the referent coincides with a prior turn's target object.
- `there`: used when referring to the spatial location of a removed entity.
- `its`: used for possessive attribute references.

Table 9 provides representative examples for each pronoun type.

Table 9: Representative pronoun rewriting examples.

Turn	Before Rewriting	After Rewriting
	Example 1: "it" for object coreference	
1	Change the red cup to blue.	Change the red cup to blue.
2	Move the blue cup to the table.	Move **it** to the table.
3	Remove the blue cup.	Remove **it**.
	Example 2: "there" for spatial reference	
1	Remove the green chair in the corner.	Remove the green chair in the corner.
2	Add a lamp to the corner.	Add a lamp **there**.
3	Change the lamp to gold.	Change **its** color to gold.
	Example 3: "its" for possessive reference	
1	Change the material of the door to iron.	Change the material of the door to iron.
2	Change the iron door to black.	Change **its** color to black.
3	Remove the iron door.	Remove **it**.

E.4 Category Statistics

Detailed task description of MICE-Bench is shown in Table 10.

In-Context Task	Editing Instruction Type	Count
Content Memory	Subject Add	157
	Subject Remove	140
	Count Change	138
	Text Change	131
	Background Change	130
	Subject Replace	127
	Position Change	125
	Color Alter	75
	Material Alter	57
Content Understanding	Color Alter	397
	Subject Remove	202
	Material Alter	146
	Subject Add	94
	Count Change	88
	Subject Replace	64
	Background Change	35
	Text Change	28
	Position Change	26

Table 10: Task distribution of MICE-Bench. Each category (Content Memory and Content Understanding) contains 1,080 editing instructions across 360 three-turn instances, distributed among nine instruction types.

E.5 Details for Evaluation Metrics

IF: Instruction Following. IF $\in \{0, 1\}$ assesses whether each edit instruction T_t is successfully executed by comparing the previous-turn image Y_t with the current edited image Y_{t+1}. Depending on the instruction type, it dispatches different verification tools. For symbolically verifiable instructions (e.g., adding, removing, or repositioning objects), GroundingDINO [15] is used to check the editing outcome against deterministic criteria. For instance, verifying a *count change* instruction "Change the number of dogs to 5" by counting exactly five detected bounding boxes for "dog" in Y_{t+1}. For semantically verifiable instructions (e.g., changing color or background), a VLM [1] is instead used to judge whether the intended change has been applied.

CC: Content Consistency. CC $\in [0, 1]$ measures the preservation of non-target content between the base image Y_0 and the current image Y_{t+1}. It consists of two components: *background consistency*, which compares the background regions (excluding all object bounding boxes) across the two images, and *per-object consistency*, which compares the appearance of each unchanged object individually. Both are computed using DINO-based feature similarity [24]. The final CC score is the average of these two components.

GA: Global Awareness. While IF and CC are sufficient for evaluating history-agnostic multi-turn editing, where each instruction depends only on the immediately preceding image, they fall short for our proposed in-context editing setting. In in-context editing, a model must maintain a session-level understanding of the user's global intent across the entire editing history, rather than treating each turn in isolation. To capture this capability, we introduce GA, it takes the full editing history s_t and the output image of current turn Y_{t+1} as input. Employing a VLM, GA $\in \{0, 1\}$ evaluates whether the *content memory* and *content understanding* challenges are correctly resolved.

Specifically, when evaluating a multi-turn session, GA conducts a strict, turn-by-turn sequential verification. For each turn t, it evaluates whether the current instruction is correctly executed when transforming Image Y_t to Image Y_{t+1}, while also ensuring compliance with any session-level global constraints established in earlier turns. If any single turn fails either the local instruction or the global constraint, the evaluation terminates immediately, and the overall GA score for the session is recorded as zero. This strict sequential criterion guarantees that the metric captures not only the correctness of each isolated edit, but also the model's ability to maintain global constraint awareness

Table 11: Agreement analysis of different GA evaluators with human annotations.

Evaluator	Agreement (%)
CLIP_dir	58.9
GA (Qwen3-VL-32B-Instruct)	68.3
GA (Qwen3-VL-235B-A22B-Instruct)	81.5

throughout the entire editing session. Prompt template that used for GA on *Content Memory* and *Content Understanding* task are shown in Figure 15 and Figure 16, respectively.

Human agreement study of VLM-Based GA metric. To validate the reliability of our VLM-based GA metric, we conduct a human agreement study on 400 editing results sampled from four representative models (Nano-Banana-2, GPT-Image-1, BAGEL, and Edit-R2) covering all three turns of MICE-Bench. Human annotators were asked to judge whether each edited image satisfies the specified global constraint, without exposure to any automatic scores. As shown in Table 11, GA evaluated with Qwen3-VL-235B achieves 81.5% agreement with human judgments, substantially outperforming the 32B variant (68.3%). The gap is most pronounced on the Content Memory subset, where the 32B judger exhibits systematic false positives—assigning GA = 1.0 to edits that humans reject. These results confirm that the 235B judger provides a reliable proxy for human preference in evaluating global constraint compliance. Figures 17, and 18 present examples of the VLM-based GA judger's outputs on both tasks, covering both successful and failed editing results. These examples demonstrate that the GA metric aligns well with human judgments.

F Details for Edit-R2

Details for Q_{CoT}. Q_{CoT} guides the thinking expert to conduct two types of operations:

1. **Coreference resolution**: replace vague pronouns ("it," "them," "its," etc.) with the specific object names visible in the current image Y_t.
2. **Global constraint injection**: if any earlier turn specifies a persistent constraint (e.g., a color or material requirement for all subsequent edits), explicitly incorporate that constraint into the current instruction.

Examples of Q_{CoT} are shown in Figure 14. Note that these questions are tailored to MICE-Bench, which differ from BAGEL's original thinking system prompt (used in"BAGEL w/ think" in Table 1).

Unified optimization objective in Edit-R2. The policy is updated by minimizing a unified objective [14] that combines Flow-GRPO and Text-GRPO losses with a shared advantage:

$$\mathcal{L}_{\text{unified}} = \mathcal{L}_{\text{Flow-GRPO}}(\theta; \mathbf{z}_t, A_t) + \lambda\, \mathcal{L}_{\text{Text-GRPO}}(\theta; \mathbf{w}_t, A_t), \tag{6}$$

where

$$\mathcal{L}_{\text{Flow-GRPO}}(\theta) = -\frac{1}{G}\sum_{i=1}^{G}\frac{1}{K}\sum_{t=0}^{K-1}\min\Big(\rho_t^{(i)}(\theta)\,A_t^{(i)}, \text{clip}\big(\rho_t^{(i)}(\theta), 1{-}\epsilon, 1{+}\epsilon\big)\,A_t^{(i)}\Big), \tag{7}$$

and

$$\begin{aligned}\mathcal{L}_{\text{Text-GRPO}}(\theta) = -\frac{1}{G}\sum_{i=1}^{G}\frac{1}{K}\sum_{t=0}^{K-1}\frac{1}{|\mathbf{w}_t^{(i)}|}\sum_{j=1}^{|\mathbf{w}_t^{(i)}|}\min\Big(&\hat{\rho}_{t,j}^{(i)}(\theta)\,A_t^{(i)},\\ &\text{clip}\big(\hat{\rho}_{t,j}^{(i)}(\theta), 1{-}\epsilon, 1{+}\epsilon\big)\,A_t^{(i)}\Big).\end{aligned} \tag{8}$$

Here $\rho_t^{(i)}(\theta) = \pi_\theta(a_t^{(i)} \mid \tilde{s}_t^{(i)})/\pi_{\theta_{\text{old}}}(a_t^{(i)} \mid \tilde{s}_t^{(i)})$ and $\hat{\rho}_{t,j}^{(i)}(\theta) = \pi_\theta(w_{t,j}^{(i)} \mid w_{t,<j}^{(i)}, s_t^{(i)})/\pi_{\theta_{\text{old}}}(w_{t,j}^{(i)} \mid w_{t,<j}^{(i)}, s_t^{(i)})$ are the per-step and per-token importance sampling ratios, ϵ is the clipping threshold, and λ controls the balance between the two losses. A KL divergence penalty $\beta\, D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})$ can be optionally added to prevent excessive deviation from the reference policy.

G Broader Impacts

Edit-R2 advances multi-turn in-context image editing by enabling models to follow evolving user instructions while maintaining session-level consistency. On the positive side, this technology can substantially lower the barrier to expressive visual content creation, benefiting designers, educators, and artists who rely on iterative, natural-language-driven workflows.

At the same time, improvements in controllable image generation carry inherent risks. More capable editing models can be misused to produce manipulated imagery for disinformation campaigns, non-consensual image alteration, or other forms of visual deception. We acknowledge these risks and intend to release model weights and benchmark data alongside usage guidelines and, where appropriate, content filters consistent with those of the base model BAGEL. We also encourage the community to develop robust detection tools and regulatory frameworks that keep pace with advances in generative image editing.

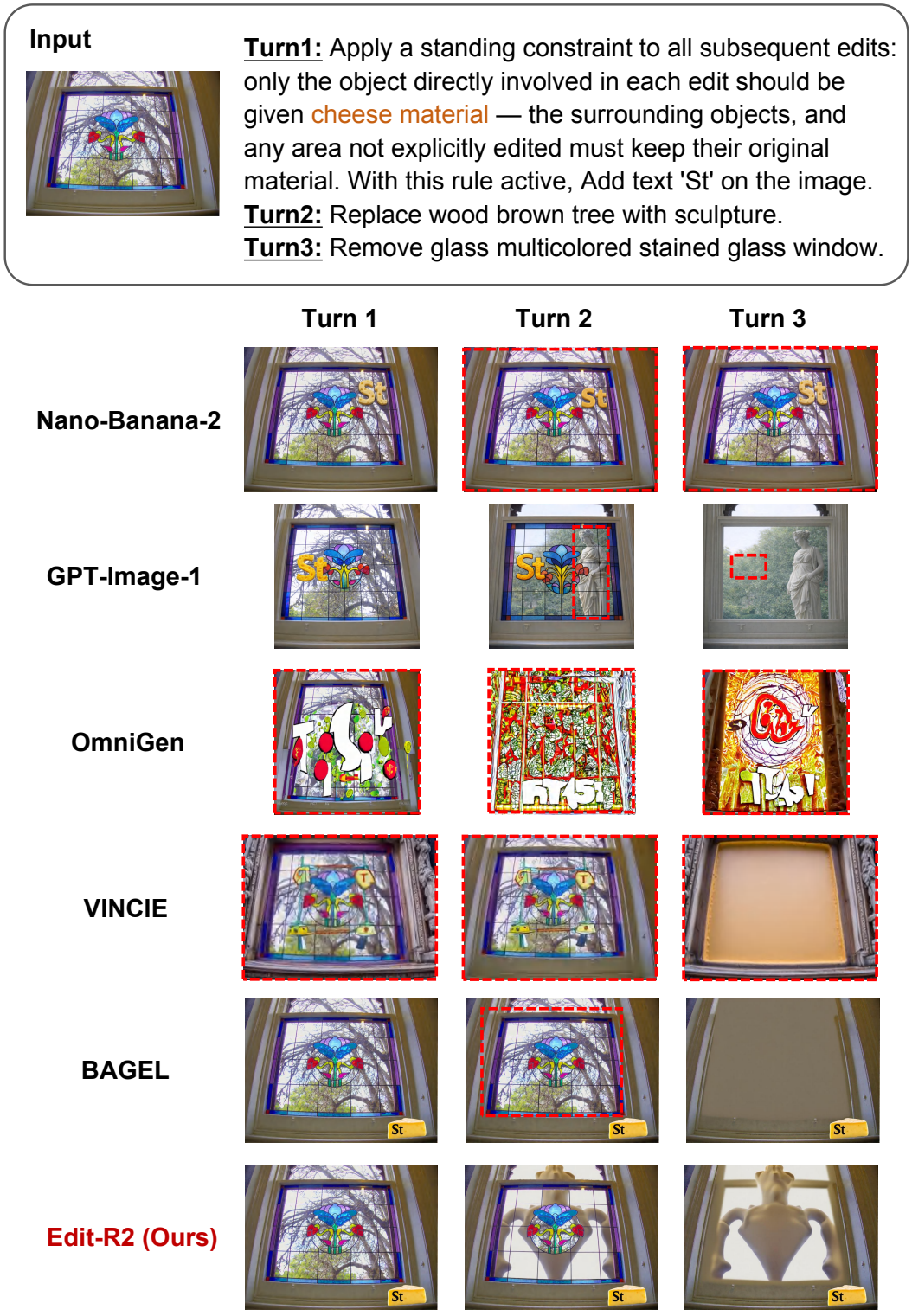


Figure 12: Additional qualitative comparison on the *Content Memory* task. GPT-Image-1 and BAGEL fail to add sculpture with cheese material at turn 2; Nano-Banana-2 ignore instructions at turn 2 and turn 3.

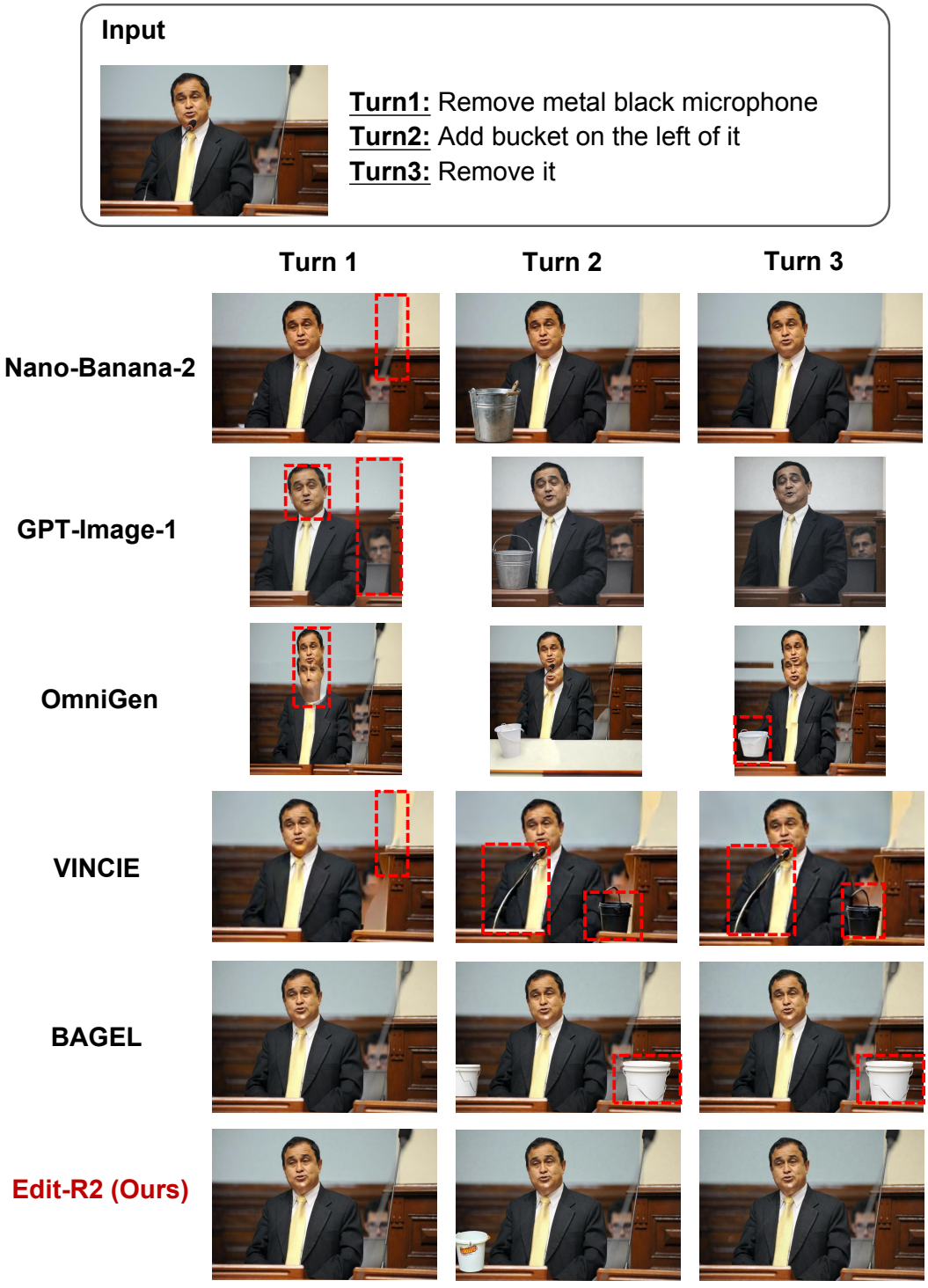


Figure 13: Additional qualitative comparison on the *Content Understanding* task. All models except Nano-Banana-2 and Edit-R2 fail in this case. The two successful models correctly identify the position of the microphone removed at turn 1, place the bucket to its left at turn 2, and successfully remove it at turn 3. However, Nano-Banana-2 unintentionally removes an unrelated object at turn 1, which degrades the CC.

Editing history:{history_text}
Current instruction to enhance: {current_prompt}
Please enhance the current instruction by doing both of the following if necessary:
1. Replace any vague pronouns (it, them, they, its, the one, etc.)
with the specific object names visible in the image.
2. If any earlier round specifies a persistent global constraint
(e.g., 'Ensure all subsequent edits are yellow', 'Use leather material for all edits'),
explicitly incorporate that constraint into the current instruction.
Think briefly, under 100 words.
Output ONLY the enhanced instruction. No explanation, no extra text.

Figure 14: Illustration of Q_{CoT}.

You are an expert evaluator specializing in multi-turn image editing tasks. You will be provided with a sequence of N images and N-1 editing instructions.

Task Definition:
The task consists of N-1 consecutive editing turns.
- Turn i (i ranges from 1 to N-1):
  - Input: Image i
  - Instruction: Instruction i
  - Output: Image i+1

Your objective is to evaluate each editing turn sequentially, beginning from Turn 1.

Evaluation Criteria Per Turn:
For each Turn i, compare Image i (input) with Image i+1 (output) with respect to Instruction i. A turn is considered successful ("yes") only when **all** of the following criteria are simultaneously satisfied:

1. **Instruction Adherence:** Image i+1 successfully reflects the modifications specified by Instruction i.

2. **Global Constraint Compliance:** If the first instruction establishes a global constraint that governs the entire editing session (e.g., "objects added or modified in subsequent edits must be yellow"), then Image i+1 must adhere to this constraint. The specific rules are as follows:
   - **Scope of Application:** Global constraint compliance is required only with respect to the objects directly involved in the current editing turn. For example, if the instruction is "add a cup on the table" and the global constraint specifies "all subsequent edits should use glass material", then in a successful Image i+1, only the newly added cup should be rendered in glass — the table and other pre-existing objects must not be altered to glass.
   - **Removal Instruction Exemption:** If the current turn's instruction involves removing or deleting an object (e.g., "remove the cup", "delete the tree in the background"), global constraint compliance is waived for that turn — the edit is considered compliant with the global constraint regardless of outcome. Since removal operations do not involve adding or modifying objects, the global constraint is inherently inapplicable.
   - **Prohibition of Global Tonal Substitution:** The global constraint requires that the specified attribute be applied to the specific objects involved in the edit, not as a wholesale tonal transformation of the entire image. If the post-edit image merely applies a color filter over the entire image to approximate the globally constrained color (e.g., the global constraint specifies "yellow", but the entire image is overlaid with a yellow tint), without the targeted objects genuinely reflecting the constraint, the turn must be judged as non-compliant.

Execution and Output Logic:
Evaluate each turn sequentially (Turn 1, Turn 2, ...).

- **If Turn i succeeds ("yes"):**
  Output:
  <answer_turn_i> yes </answer_turn_i>
  <reason_turn_i> Briefly explain why the turn is considered successful. </reason_turn_i>
  Then proceed to evaluate Turn i+1 (if applicable).

- **If Turn i fails ("no"):**
  Output:
  <answer_turn_i> no </answer_turn_i>
  <reason_turn_i> Explain the reason for failure (e.g., "failed to add the specified object", or "object B was successfully added, but object A was inadvertently removed"). </reason_turn_i>
  Halt immediately. Do not evaluate any subsequent turns.
  Output: <answer_final> no </answer_final>

- **If all turns (1 through N-1) succeed:**
  After evaluating the final turn, output: <answer_final> yes </answer_final>

Input Data:
Image Sequence:
- Image 1: The initial image
- Image 2: The result of Turn 1
- ...
- Image N: The result of Turn N-1

Instructions:
{instructions_formatted}

Response Format:
Provide your evaluation strictly using the XML-style tags described above. Do not include any conversational text outside of the tags. Replace the placeholder 'i' in the tags with the actual turn index (e.g., <answer_turn_1>, <answer_turn_2>).

Figure 15: Prompt template used for VLM-Based GA metric on *Content Memory* task.

You are an expert evaluator specializing in assessing **Content Understanding** capabilities in multi-turn image editing tasks. You will be provided with a sequence of N images and N-1 editing instructions.

Task Background:
In real-world multi-turn image editing scenarios, users naturally transition from using full object names to pronouns (e.g., "it", "them", "there", "its") when referring to objects that were manipulated in previous turns, once the object context has been established. This requires the editing model to perform accurate **pronoun resolution** — i.e., correctly identifying the specific object a pronoun refers to and executing the intended editing operation on that object.

Task Definition:
The task consists of N-1 consecutive editing turns.
- Turn i (i ranges from 1 to N-1):
  - Input: Image i
  - Instruction received by the model: Instruction i (may contain pronouns such as "it", "them", "there", "its")
  - Explicit Reference Instruction: Formatted Instruction i (the specific object referred to by each pronoun is annotated in square brackets, serving as the objective reference for evaluation)
  - Output: Image i+1

Your objective is to evaluate each editing turn sequentially, beginning from Turn 1.

Evaluation Criteria Per Turn:
For each Turn i, compare Image i (input) with Image i+1 (output) with respect to Instruction i. A turn is considered successful ("yes") only when **all** of the following criteria are simultaneously satisfied:

1. **Pronoun Resolution (Content Understanding):** If Instruction i contains pronouns (e.g., "it", "them", "there", "its"), the object edited in Image i+1 must correspond to the object annotated in square brackets in Formatted Instruction i. That is, the model must correctly identify which specific object the pronoun refers to and perform the edit on that object — not on any other. The specific rules are as follows:
   - **"it" / "them" reference:** The pronoun refers to the same object manipulated in the preceding turn. For example, if the instruction is "Remove it" and the formatted instruction is "Remove [red car]", then the object removed in Image i+1 must be the red car, not any other object.
   - **"its" reference:** The possessive pronoun refers to an attribute of the object manipulated in the preceding turn. For example, if the instruction is "Change its color to blue" and the formatted instruction is "Change the color of [wooden brown door] to [blue]", then it is the wooden brown door whose color should become blue in Image i+1.
   - **"there" reference:** The spatial pronoun refers to the location previously occupied by an object that was removed in an earlier turn. For example, if the instruction is "Add a bench there" and the formatted instruction is "Add [bench] on the [left] of [flower bed]", then Image i+1 should contain a bench placed at the location where the flower bed previously was.
   - **Turn 1 typically contains no pronouns** (as there is no prior context), in which case only the **Instruction Following** criterion below is evaluated.

2. **Instruction Following:** Image i+1 successfully reflects the modifications specified by Instruction i — including the correct edit type (add / remove / replace / recolor, etc.) and the correct attribute value (e.g., the color is indeed changed to the specified color).

Execution and Output Logic:
Evaluate each turn sequentially (Turn 1, Turn 2, ...).

- **If Turn i succeeds ("yes"):**
  Output:
  <answer_turn_i> yes </answer_turn_i>
  <reason_turn_i> Briefly explain why the turn is considered successful. </reason_turn_i>
  Then proceed to evaluate Turn i+1 (if applicable).

- **If Turn i fails ("no"):**
  Output:
  <answer_turn_i> no </answer_turn_i>
  <reason_turn_i> Provide a detailed explanation of the failure. Clearly indicate whether the failure is due to a pronoun resolution error, an instruction following error, or both. For example:
    - Pronoun resolution error: "The pronoun 'it' should refer to [red car], but the model incorrectly removed the blue truck."
    - Instruction following error: "The target object [wooden door] was correctly identified, but its color was not successfully changed to blue."
    - Both errors: "The pronoun 'its' should refer to an attribute of [silver helmet], but the model modified the color of a different object, and the resulting color is also incorrect." </reason_turn_i>
  Halt immediately. Do not evaluate any subsequent turns.
  Output: <answer_final> no </answer_final>

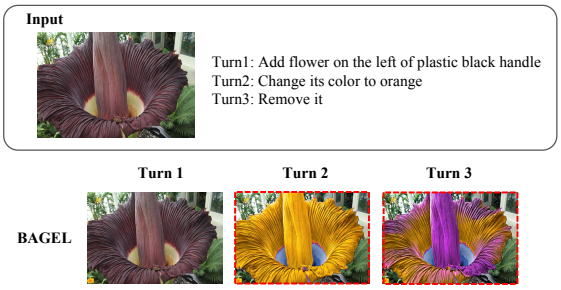
- **If all turns (1 through N-1) succeed:**
  After evaluating the final turn, output: <answer_final> yes </answer_final>

Input Data:
Image Sequence:
- Image 1: The initial image
- Image 2: The result of Turn 1
- ...
- Image N: The result of Turn N-1

Instructions:
{instructions_formatted}

Response Format:
Provide your evaluation strictly using the XML-style tags described above. Do not include any conversational text outside of the tags. Replace the placeholder 'i' in the tags with the actual turn index (e.g., <answer_turn_1>, <answer_turn_2>).

Figure 16: Prompt template used for VLM-Based GA metric on *Content Understanding* task.



Evaluation Process:

Turn 1:
- **Input Image:** Image 1
  - Description: A dark purple flower with a black plastic handle in the background.
- **Instruction Received by Model:** Add flower on the left of plastic black handle
  - Explicit Reference Instruction: Add [flower] on the [left] of [plastic black handle]
- **Output Image:** Image 2
  - Description: An orange flower added to the left of the black plastic handle.

Evaluation:
1. **Pronoun Resolution (Content Understanding):**
   - The instruction in this turn contains no pronouns, so pronoun resolution is not required.
2. **Instruction Following:**
   - The instruction requires adding a flower to the left of the black plastic handle. In Image 2, an orange flower has indeed been added to the left of the black plastic handle, meeting the instruction.

Conclusion:
<answer_turn_1> yes </answer_turn_1>
<reason_turn_1> The instruction requires adding a flower to the left of the black plastic handle. Image 2 successfully achieves this change, with both the position and the added object being correct. </reason_turn_1>

Turn 2:
- **Input Image:** Image 2
  - Description: An orange flower added to the left of the black plastic handle.
- **Instruction Received by Model:** Change its color to orange
  - Explicit Reference Instruction: Change the color of [flower] to [orange]
- **Output Image:** Image 3
  - Description: The flower has been changed to yellow.

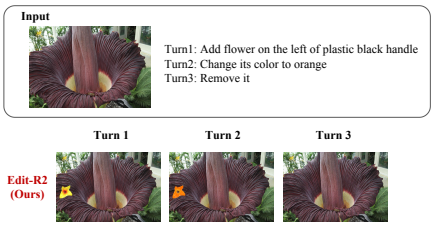
Evaluation:
1. **Pronoun Resolution (Content Understanding):**
   - The pronoun "its" in the instruction refers to the object added in the previous round, i.e., the flower. According to the explicit reference instruction, the model needs to change the color of the flower. However, the model did not correctly understand that "its" refers to the newly added flower in the first round; instead, it mistakenly changed the original flower in the image to orange.
2. **Instruction Following:**
   - Although the model changed a flower to orange, it operated on the wrong object (the original flower rather than the newly added one). Therefore, the instruction is not correctly followed.

Conclusion:
<answer_turn_2> **no** </answer_turn_2>
<reason_turn_2> **The model failed to correctly understand that "its" refers to the flower newly added in the first round; instead, it mistakenly changed the original flower in the image to orange. This is a pronoun resolution error.** </reason_turn_2>

Final Conclusion:
Since Turn 2 fails to complete the edit successfully, no subsequent rounds are evaluated.

<answer_final> **no** </answer_final>

Figure 17: Example of GA output for failed editing results on content understanding task.



Evaluation Process:

Turn 1:
- **Input Image:** Image 1
  - Description: A dark purple flower with a black plastic handle in the background.
- **Instruction Received by Model:** Add flower on the left of plastic black handle
  - Explicit Reference Instruction: Add [flower] on the [left] of [plastic black handle]
- **Output Image:** Image 2
  - Description: A yellow flower added to the left of the black plastic handle.

Evaluation:
1. **Pronoun Resolution (Content Understanding):**
   - The instruction in this turn contains no pronouns, so pronoun resolution is not required.
2. **Instruction Following:**
   - The instruction requires adding a flower to the left of the plastic black handle. Observing Image 2, a yellow flower has indeed been added to the left of the plastic black handle.

Conclusion:
<answer_turn_1> yes </answer_turn_1>
<reason_turn_1> The instruction requires adding a flower to the left of the plastic black handle. Image 2 successfully adds a yellow flower, meeting the instruction. </reason_turn_1>

Turn 2:
- **Input Image:** Image 2
  - Description: A yellow flower added to the left of the black plastic handle.
- **Instruction Received by Model:** Change its color to orange
  - Explicit Reference Instruction: Change the color of [flower] to [orange]
- **Output Image:** Image 3
  - Description: The flower's color has been changed to orange.

Evaluation:
1. **Pronoun Resolution (Content Understanding):**
   - The pronoun "its" in this turn refers to the flower added in the previous round. The explicit reference instruction clearly refers to "[flower]", i.e., the yellow flower added in the previous round. The model indeed changes the color of the flower to orange in Image 3, so pronoun resolution is correct.
2. **Instruction Following:**
   - The instruction requires changing the color of the flower to orange. Observing Image 3, the flower's color has been successfully changed to orange.

Conclusion:
<answer_turn_2> yes </answer_turn_2>
<reason_turn_2> The instruction requires changing the color of the flower to orange. Image 3 successfully changes the flower's color to orange, meeting the instruction. </reason_turn_2>

Turn 3:
- **Input Image:** Image 3
  - Description: An orange flower added to the left of the black plastic handle.
- **Instruction Received by Model:** Remove it
  - Explicit Reference Instruction: Remove [orange flower]
- **Output Image:** Image 4
  - Description: The orange flower has been removed.

Evaluation:
1. **Pronoun Resolution (Content Understanding):**
   - The pronoun "it" in this turn refers to the orange flower edited in the previous round. The explicit reference instruction clearly refers to "[orange flower]", i.e., the flower whose color was changed to orange in the previous round. The model indeed removes the orange flower in Image 4, so pronoun resolution is correct.
2. **Instruction Following:**
   - The instruction requires removing the orange flower. Observing Image 4, the orange flower has been removed, meeting the instruction.

Conclusion:
<answer_turn_3> **yes** </answer_turn_3>
<reason_turn_3> **The instruction requires removing the orange flower. Image 4 successfully removes the orange flower, meeting the instruction.** </reason_turn_3>

Final Conclusion:
All rounds successfully complete the editing task.

<answer_final> **yes** </answer_final>

Figure 18: Example of GA output for successful editing results on content understanding task.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly state all three contributions: (i) the MICE-Bench benchmark with automated GA metric, (ii) the Edit-R2 RL framework with IC-CoT and unified multi-turn policy optimization, and (iii) quantitative improvements over strong baselines. All claims are directly supported by experimental results in Section 5.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: A dedicated Limitations paragraph appears at the end of the Conclusion section, discussing that MICE-Bench currently covers two in-context editing scenarios and that extending to a broader taxonomy and longer multi-turn sessions are left for future work.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper does not include formal theoretical results or proofs. Contributions are algorithmic and empirical in nature; the MDP formulation and optimization objectives are presented as methods rather than theorems.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: Algorithm 1 provides full pseudocode for Edit-R2. Appendix A details all hyperparameters (LoRA rank, learning rate, group size, clipping threshold, KL coefficient, reward weights, denoising steps), optimizer (AdamW), hardware (4×6 NVIDIA H200 GPUs), and evaluation protocol. The benchmark construction pipeline is fully described in Appendix E.

Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The MICE-Bench construction pipeline is fully documented in Appendix E with all prompt templates and design rationale. Code and benchmark data will be made publicly available upon acceptance.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: All training hyperparameters are summarized in Table 5 in Appendix A, including LoRA configuration, optimizer, learning rate, group size, clipping threshold, KL coefficient, reward weights, and hardware. Evaluation details (VLM judger, denoising steps) are also specified.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Main results are reported as averages over 4 independent runs, providing a measure of result stability across different random seeds.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Appendix A specifies that training is distributed across 4×6 NVIDIA H200 GPUs with a total batch size of 48, with 2 GPUs per node reserved for reward model deployment.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics. The human annotation study involves low-risk judgment tasks. No sensitive personal data is collected or released.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: A Broader Impacts section is included in the appendix discussing positive applications (accessible and expressive image editing) and potential negative impacts (image manipulation and misinformation), along with mitigation strategies.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [N/A]

Justification: No public model or dataset release is made at submission time. Upon release, we will adopt standard content-safety measures (usage guidelines and content filters) consistent with the base model BAGEL's release policy.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: All models, tools, and datasets used in this work—including BAGEL, GroundingDINO, Qwen3-VL, EdiVal-Agent, and all baseline models—are properly cited throughout the paper. The base model BAGEL is used within its publicly specified terms.

Guidelines:

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

13. **New assets**

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

Answer: [Yes]

Justification: MICE-Bench is documented in Appendix E with full details of its construction pipeline, task categories (Table 10), evaluation metrics (Appendix E.5), prompt templates (Figure 15, 16), and human agreement validation.

Guidelines:

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

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

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

Answer: [Yes]

Justification: A human agreement study on 400 editing results is described in Appendix E.5. Annotators were asked to judge whether each edited image satisfies the specified global constraint without exposure to automatic scores. The annotation task and its scope are clearly described.

Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The human annotation study involves only low-risk image quality judgment tasks with no collection of personal data, and does not require IRB approval under standard institutional guidelines.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: LLMs are an integral part of the core methodology. Qwen3-VL is used as the reward model for IF and GA during training and evaluation (Section 4 and Appendix A). LLMs are also used in the MICE-Bench construction pipeline for object inventory extraction, instruction generation, and pronoun rewriting (Appendix E).

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.