INTERLEAVETHINKER: REINFORCING AGENTIC INTERLEAVED GENERATION

Anonymous authors
Paper under double-blind review



Figure 1: Capabilities of InterleaveThinker, consisting of interleaved generation with various input types, real-world action interaction, and robotic manipulation. Gray: inputs, blue: outputs.

ABSTRACT

Recent image generators have demonstrated impressive photorealism and instruction-following capabilities in single-image generation and editing. However, constrained by their architectures, they cannot achieve interleaved generation (text-image sequence), which has crucial applications in visual narratives, guidance, and embodied manipulation. Even the latest open-source Unified Multimodal Models (UMMs) exhibit limited performance in this regard. In this paper, we introduce InterleaveThinker, the first multi-agent pipeline designed to endow frozen image generators with interleaved generation capabilities. Specifically, we employ a planner agent to organize the image-text input sequence, instructing the image generator on the required execution at each step. Subsequently, we introduce a critic agent to evaluate the generator's outputs, identify samples that deviate from the planned instructions, and refine the instructions for regeneration. To implement this pipeline, we construct the Interleave-Planner-SFT-80k and Interleave-Critic-SFT-112k to perform a format cold-start. Then we develop Interleave-Critic-RL-13k to reinforce the instruction correction capability within a generation trajectory using GRPO. Since an interleaved generation trajectory may involve over 25 generator calls, full-trajectory optimization is computationally expensive and suffers from

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

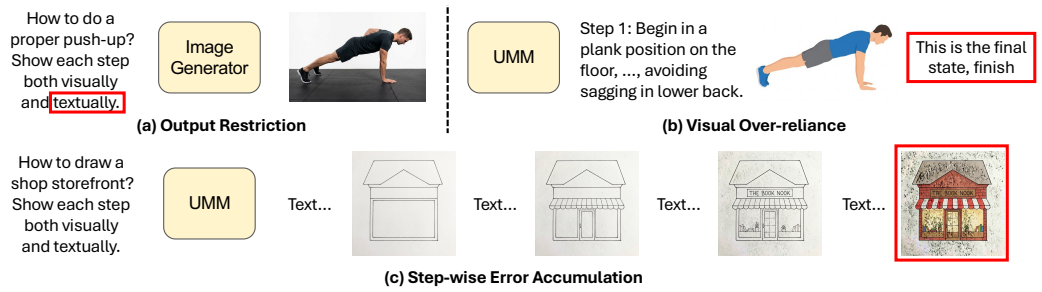

Figure 2: **Problems in image generator and UMM for interleaved generation**. Highlight in red.

difficult credit assignment. We therefore propose **T**rajectory-**D**ecoupled **GRPO** (**TD-GRPO**), which performs single-step optimization with accuracy and step-wise rewards to provide effective learning signals throughout the generation process. InterleaveThinker shows consistent gains across two off-the-shelf image generators and achieves performance comparable to Nano Banana and GPT-5 on interleaved generation benchmarks. Beyond generation, it also substantially improves the base model on reasoning-based benchmarks; for example, on 4-step FLUX.2-klein, WISE increases from 0.47 to 0.73 and RISE from 13.3 to 28.9.

1 INTRODUCTION

Recent advancements in image generation and editing have demonstrated remarkable photorealism and instruction-following capabilities. However, these models (Labs, 2024; 2025; Wu et al., 2025a; Team et al., 2025; Esser et al., 2024; Wang et al., 2025) are fundamentally designed for single-image generation/editing. In real-world applications, there is a growing demand for interleaved generation, a workflow that takes an interleaved text and image sequence as input and outputs a coherent, multi-step sequence of text and images. This capability holds crucial value for visual narratives, guidance, and embodied manipulation. Unfortunately, constrained by their inherent image-only output architectures, existing image generators cannot natively achieve this, leaving a significant gap between single-image synthesis and complex sequential generation.

The emergence of Unified Multimodal Models (UMMs) (Chen et al., 2025b; Cui et al., 2025; Google, 2025a; Deng et al., 2025; Wu et al., 2025b; Cao et al., 2025; Zheng et al., 2025) offers a potential solution, as their architectures naturally support interleaved text and image generation. However, because they generate sequences step-by-step based on preceding images, UMMs suffer from two critical problems in long-horizon tasks: 1) **Visual over-reliance**. As shown in Fig 2(b), when generating a repetitive action sequence like a push-up, the model might stop at an intermediate state that visually resembles the final goal. 2) **Step-wise error accumulation**. As current UMMs have not yet achieved a stable "aha-moment" for self-correction, a slight degradation in early image quality compounds step-by-step, eventually ruining the final output, as shown in Fig 2(c).

In this paper, we propose InterleaveThinker, the first multi-agent framework that equips frozen image generators with strong interleaved generation capabilities. The core motivation for this multi-agent design is to eradicate visual over-reliance and resolve step-wise error accumulation through an explicit correction mechanism. If a single VLM alternates between planning and evaluating generated images, it becomes overly conditioned on intermediate visual states. This causes the model to lose sight of the global objective and myopically react to local visual feedback, inevitably leading to step-wise error accumulation. To fundamentally resolve this, InterleaveThinker employs a Planner agent to predict the entire sequence of instructions upfront. This completely bypasses visual over-reliance by blocking intermediate feedback. To monitor the subsequent execution, a Critic agent then evaluates the step outputs, identifies deviations from the initial instructions, and refines prompts for regeneration, ensuring strict adherence to the overall trajectory without updating the generator.

A primary challenge in implementing this multi-agent pipeline is the absence of tailored training data. To address this, we first curate a comprehensive prompt list spanning diverse interleaved generation tasks and scenarios, including embodied manipulation, art, storytelling, image description,

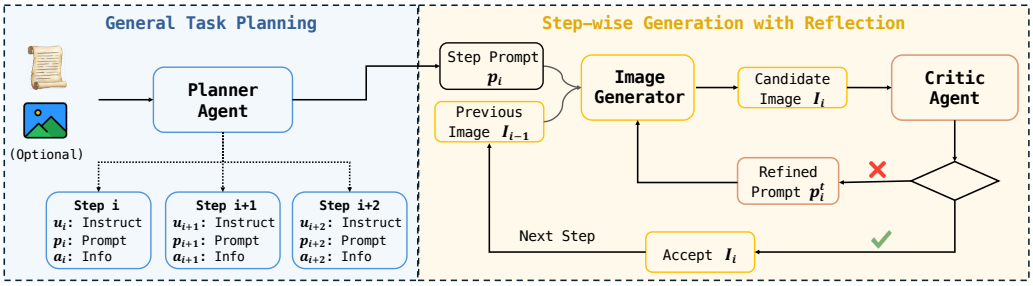

Figure 3: **Overview of InterleaveThinker**. t means the refinement iterations.

workflows, daily life, science, and professional skills. Using these prompts, we iteratively employ advanced models (Gemini 2.5 Pro and Nano Banana Pro) to generate detailed agentic trajectories. To guarantee high-quality supervision, we implement a rigorous data filtering pipeline (detailed in Sec 2.2). Ultimately, this process yields three high-quality datasets: Interleave-Planner-SFT-80k and Interleave-Critic-SFT-112k for multi-agent cold-start, together with Interleave-Critic-RL-13k for reinforcing the critic's step-wise correction capabilities via GRPO. Since a single interleaved trajectory may involve over 25 generator calls, full-trajectory optimization incurs substantial computational overhead and makes credit assignment difficult. We therefore introduce **T**rajectory-**D**ecoupled **GRPO** (**TD-GRPO**), which decomposes trajectory-level optimization into efficient single-step updates with accuracy and step-wise rewards. This design provides fine-grained learning signals while substantially reducing the cost of long-horizon optimization.

To validate the compatibility of InterleaveThinker, we evaluate the pipeline across two off-the-shelf image generators, observing consistent performance gains. As a representative default, we adopt the 4-step FLUX.2-klein to minimize long-horizon latency. Under this setup, our approach significantly surpasses existing open-source UMMs on rigorous interleaved generation benchmarks, achieving performance comparable to the proprietary Nano Banana and GPT-5. Surprisingly, beyond interleaved generation, our framework also significantly enhances the base model on reasoning-based benchmarks. Specifically, we observe substantial improvements on the WISE benchmark (increasing from 0.47 to 0.73) and the RISE benchmark (leaping from 13.3 to 28.9). These results highlight the immense potential of multi-agent collaboration in unlocking complex, sequential reasoning and generation capabilities for existing image models.

In summary, our main contributions are as follows:

- We propose InterleaveThinker, the first multi-agent framework that equips frozen image generators with strong interleaved generation capabilities. Its Planner-Gen-Critic workflow mitigates visual over-reliance and step-wise error accumulation in UMMs.
- To support training, we build a dedicated data pipeline to construct interleaved generation data across diverse scenarios, resulting in three high-quality datasets: Interleave-Planner-SFT-80k, Interleave-Critic-SFT-112k, and Interleave-Critic-RL-13k. In addition, we introduce TD-GRPO, which decomposes long-horizon trajectory optimization into efficient single-step updates with accuracy and step-wise rewards, reducing computational cost while enabling more effective credit assignment.
- Extensive experiments validate the effectiveness of InterleaveThinker across two distinct image generators. Using 4-step FLUX.2-klein as generator, we not only surpass existing open-source UMMs on interleaved generation, but also largely improves the base model on reasoning benchmarks, increasing WISE from 0.47 to 0.73 and RISE from 13.3 to 28.9.

2 INTERLEAVETHINKER

To endow existing frozen image generators with robust interleaved generation capabilities, and address the visual over-reliance, step-wise error accumulation problem in UMMs, we propose InterleaveThinker, a multi-agent framework for frozen image generators. We show our multi-agent workflow, data construction pipeline, and training scheme below.

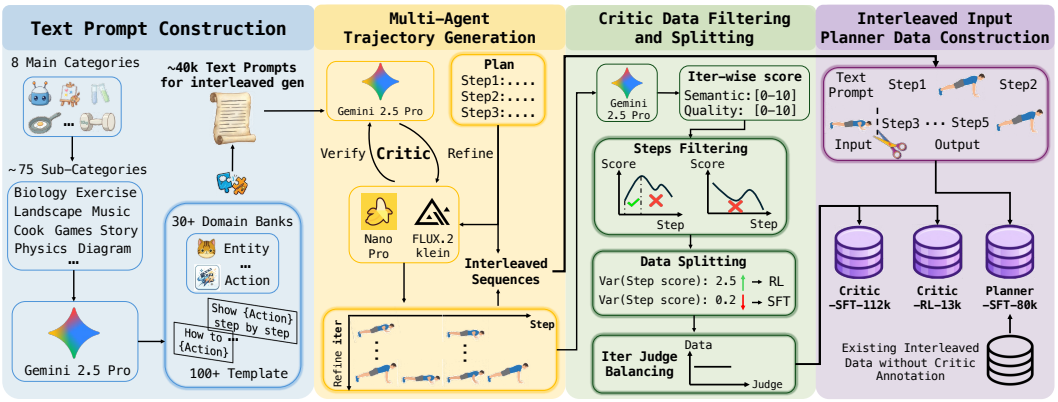

Figure 4: **Illustration of our data construction pipeline**.

2.1 Multi-Agent Pipeline

As shown in Fig 3, we formulate a progressive, closed-loop pipeline comprising three core modules: a *Planner*, a *Critic*, and a *Generator* (An image model that **supports both image generation and editing**, such as FLUX.2-klein (Labs, 2026), Qwen-image-Edit (Wu et al., 2025a)). The framework decomposes the complex interleaved generation process into a step-wise execution plan, incorporating a self-correction mechanism to ensure high-fidelity generation and editing. Let S denote the input interleaved sequence of images and text. The overall pipeline operates through the following formalized stages:

1. Planner: The Planner is responsible for analyzing the input sequence S and translating it into an N-step execution plan. For each step $i \in \{1, \ldots, N\}$, the Planner generates a step instruction u_i, a model-friendly initial prompt p_i adapted from u_i, and an auxiliary text a_i, which provides supplementary knowledge-based elaboration required for specific image generation tasks. The planning process is formulated as:

$$\{(u_i, p_i, a_i)\}_{i=1}^{N} = \texttt{Planner}(S). \tag{1}$$

2. Generator: At step i and refinement iteration $t \in \{1, T_{max}\}$, the Generator takes the current refined prompt r_i^t (r_i^0=p_i) and the image from the previous step I_{i-1} to produce the current image I_i^t:

$$I_i^t = \texttt{Generator}\left(r_i^t, I_{i-1}\right). \tag{2}$$

Note: For the initial generation step i=1, where no prior visual context exists, I_0 is defined as $\emptyset$.

3. Critic: To ensure the generated output I_i strictly aligns with the intended instruction p_i, we introduce a Critic module that provides quantitative feedback and prompt optimization. At iteration t of step i, the Critic evaluates the transition from the pre-execution image I_{i-1} to the post-execution image I_i^t. It takes the initial prompt p_i and the current refined prompt r_i^t as textual conditions. The Critic outputs a binary judgment j_i^t, a new refined prompt r_i^{t+1} for the next iteration, and a reasoning process R_i^t:

$$\left(j_i^t, r_i^{t+1}, R_i^t\right) = \texttt{Critic}\left(I_{i-1}, I_i^t, p_i, r_i^t\right), \tag{3}$$

Note: For the initial step i=1, I_0 is set as a blank white image to maintain input consistency.

This generation-evaluation loop (Stage 2↔3) iterates until a positive execution judgment (`True`) is obtained, or a maximum number of iterations T_{max} is reached. Upon satisfaction, the pipeline finalizes I_i and a_i, appends them to the output sequence, and proceeds to step $i + 1$. We also show a comprehensive workflow example in Fig 6.

2.2 Dataset Construction Pipeline

High-quality training data is essential for developing agents capable of long-horizon planning and step-wise correction. However, aligned pairs of interleaved instructions, intermediate visual states,

and Critic judgments, refinements, and reasoning processes do not naturally exist. To address this, as shown in Fig 4, we construct a dedicated data pipeline comprising four main stages.

Text Prompt Construction. We curate a comprehensive set of text prompts that covers primary interleaved generation tasks (visual narrative, guidance, and embodied manipulation). To ensure dataset diversity, we propose a systematic, top-down generation pipeline. We initiate this process by defining 8 main categories spanning broad domains, including robotics, visual storytelling, art, workflows, daily life, science, and professional skills. These main categories are further divided into approximately 75 fine-grained sub-categories, such as biology, cooking, and physics. We then prompt Gemini 2.5 Pro (DeepMind, 2025) to expand these sub-categories into more than 30 domain-specific vocabulary banks, extracting key entities and actions. Finally, we populate over 100 predefined instructional templates (e.g., "How to {Action}", "Show {Action} step by step") with elements from these domain banks. This procedural generation approach ultimately yields roughly 40,000 diverse text prompts tailored for interleaved generation.

Multi-Agent Trajectory Generation. Given the collected prompts, we employ advanced proprietary models, Gemini 2.5 Pro (DeepMind, 2025) and Nano Banana Pro (Google, 2025b), to generate agentic trajectories. For each task, the Planner agent first generates a global step-by-step instruction sequence. Then, an image generator executes these instructions step by step. We use both Nano Banana Pro and FLUX.2-klein-9B to diversify generation quality and prevent the Critic from becoming biased toward high-quality outputs. At each step, the Critic agent evaluates the generated image, compares it against the Planner's original instruction, and produces a critique. If the image deviates from the instruction, the Critic refines the prompt for immediate regeneration. This iterative process yields complete trajectories containing global plans, intermediate images, critiques, and refined prompts.

Critic Data Filtering and Splitting. To ensure the quality of the synthesized trajectories, we apply a rigorous filtering pipeline that eliminates samples with severe logical inconsistencies or poor visual quality. Note that *this filtering process is exclusively applied to curate the training data for the Critic*, while the training data for the Planner remains unfiltered. Since a single interleaved trajectory may involve over 25 generator calls, full-trajectory RL is computationally expensive and makes credit assignment difficult. We therefore decompose the generated trajectories into independent step-wise data. This decomposition enables single-step optimization used in TD-GRPO (as detailed in Sec. 2.3). We then employ Gemini 2.5 Pro (DeepMind, 2025) with an adapted system prompt from VIEScore (Ku et al., 2024) to evaluate every refinement iteration within each step, assigning scores from 0 to 10 for both semantic alignment and visual quality. Based on these iteration-level scores, we process the step-wise data through the following three stages.

1) Steps Filtering. We analyze the progression of the Gemini 2.5 Pro scores across the refinement iterations within each step. As illustrated in the scoring curves, we discard steps that exhibit negative refinement trends, score degradation, or persistent low quality. Only the steps demonstrating successful refinement, characterized by an upward or stable high-score trajectory, are retained for subsequent processing. **2) SFT-RL Data Splitting.** To construct tailored datasets for SFT and RL, we compute the variance of the iteration scores within each valid step. Steps with a high score variance indicate a dynamic refinement process with substantial quality shifts, making them ideal for RL optimization. Conversely, steps with low variance represent stable and high-quality generation, which are better suited for the SFT dataset. We partition the data accordingly and maintain an empirical sample ratio of 2:1 between the SFT and RL subsets. **3) Iter-wise Judgment Distribution Balancing.** The Critic includes an objective to predict the binary judgment of a given iteration. Training on the natural, heavily skewed data leads to biased estimations. To address this, we balance the iteration-wise data by resampling the samples as shown in Fig 4, yielding two high-quality datasets: `Interleave-Critic-SFT-112k`, and `Interleave-Critic-RL-13k`.

Interleaved Input Planner Data Construction. Since our initial instructions consist solely of pure text prompts, the resulting dataset naturally lacks the multimodal interleaved context required to train the Planner. To address this limitation, we adopt two strategies to construct interleaved input-output pairs. First, we generate self-synthesized interleaved trajectories by interleaving the previously generated textual plans with their corresponding final image outputs at each step. To formulate training pairs, we randomly select a step to truncate this sequence. The sequence preceding the truncation point acts as the interleaved multimodal input, while the subsequent text plan is assigned as the target output. Second, we leverage existing open-source interleaved datasets (Chen et al., 2025a). Although these datasets lack the fine-grained annotations necessary for training the Critic,

Table 1: **Comparison on UEval (Li et al., 2026a).** We evaluate open-source and proprietary frontier models on 8 tasks in UEval. **Bold** indicates the best result among each group.

Models	Space	Textbook	Diagram	Paper	Art	Life	Tech	Exercise	Avg
Reference									
Reference	96.2	94.4	93.1	96.2	90.6	87.7	90.6	89.2	92.2
Proprietary Frontier Models									
Gemini-2.0-Flash (Kampf & Brichtova, 2025)	65.2	55.2	47.6	45.8	70.4	58.0	50.2	48.0	55.1
GPT-5-Instant (OpenAI, 2025a)	77.3	77.9	62.3	55.1	71.2	69.7	50.7	57.6	65.2
GPT-5-Thinking (OpenAI, 2025a)	84.0	78.0	67.8	51.9	67.8	63.8	57.0	61.4	66.4
Nano Banana (Google, 2025a)	78.0	74.0	66.4	71.6	66.6	63.0	58.2	50.0	66.0
Nano Banana Pro (Google, 2025b)	79.4	89.6	75.9	81.3	84.3	73.5	60.8	63.9	76.1
Open-Source Models									
Janus-Pro (Chen et al., 2025b)	21.0	31.0	37.4	15.2	26.4	23.0	17.6	11.5	22.9
Show-o2 (Xie et al., 2025)	25.4	33.1	33.2	17.4	25.6	15.6	17.4	13.1	22.6
MMaDA (Yang et al., 2025b)	10.8	20.0	14.2	13.3	15.7	15.8	12.4	12.6	14.4
BAGEL (Deng et al., 2025)	29.8	42.5	37.2	20.0	39.0	33.6	24.8	21.4	31.0
Emu3.5 (Cui et al., 2025)	59.1	57.4	41.1	31.6	59.3	62.0	37.0	45.4	49.1
InterleaveThinker+FLUX.2-klein-9B	63.1	92.6	82.1	75.7	72.5	54.9	39.6	44.0	**67.9**
InterleaveThinker+Qwen-Image-Edit	67.5	91.0	80.6	79.4	71.3	55.9	39.8	44.5	**68.0**

their natural text-image structures are perfectly suited for the Planner. Consequently, the final training corpus for the Planner is composed of both the self-synthesized truncated sequences and the external unannotated interleaved data. Ultimately, this process yields `Interleave-Planner-SFT-80k`

2.3 Training Scheme

Based on the constructed datasets, we train the InterleaveThinker framework through a two-stage pipeline: SFT for multi-agent format cold-start, followed by RL to reinforce the Critic's correction capabilities using GRPO.

Planner-SFT. The Planner is initialized with Qwen3-VL-8B-Instruct (Bai et al., 2025a) and fine-tuned using the `Interleave-Planner-SFT-80k` dataset. Details regarding the system prompt and SFT format can be found in Appendix F. The SFT training equips the model with the ability to break down a complex user request into a coherent, global sequence of text-image instructions upfront, thereby bypassing the visual over-reliance problem. Note that we did not apply RL to the Planner. Because our trajectories can involve over 25 rounds of generator tool calls, the reward signals become highly sparse, making RL optimization highly unstable. Furthermore, since SFT alone already achieves strong performance, RL was deemed unnecessary.

Critic-SFT. The Critic is initialized with Qwen3-VL-8B-Instruct (Bai et al., 2025a), SFT teaches the model the basic format of evaluation: observing the current visual state, identifying deviations from the planned instruction, and formulating a refined prompt for the generator. We show this format below and the system prompt is shown in Appendix F.

```
<think></think><answer>[Judgment][Refined Prompt]</answer>
```

TD-GRPO. A key challenge in applying RL to interleaved generation is the long generation trajectory, where a single task may involve over 25 generator calls. End-to-end trajectory optimization is therefore computationally expensive and suffers from difficult credit assignment. We introduce Trajectory-Decoupled GRPO (TD-GRPO), which decomposes long-horizon trajectory optimization into efficient single-step policy updates.

This decomposition naturally follows our Planner-Gen-Critic design: the Planner specifies all step-wise instructions upfront, while the Critic independently evaluates and refines each generation step.

Table 2: **Comparison on CoMM (Chen et al., 2025a).** Sty. and Enti. denote style and entity consistency among generated images. Tren. measures trend alignment between image and text sequences. Comp. denotes completeness, ImgQ image quality, and IRS text-image alignment. x/x reflects performance on interleaved (Task 3) and pure-text (Task 4) inputs.

Model	Sty.	Enti.	Tren.	Comp.	ImgQ	IRS
MiniGPT-5 (Zheng et al., 2023)	5.6 / 5.7	5.2 / 5.2	5.2 / 5.3	6.3 / 5.8	6.4 / 6.2	2.6 / 2.7
SEED-LLaMA (Ge et al., 2023)	6.3 / 7.6	5.8 / 6.8	5.7 / 6.2	6.3 / 5.1	6.6 / 6.4	2.9 / 1.5
Emu2 (Sun et al., 2024)	8.2 / 8.4	8.0 / 7.6	8.0 / 7.6	8.5 / 7.5	8.6 / 7.6	2.4 / 2.0
DuoGen (Shi et al., 2026)	- / 9.2	- / 9.2	- / 9.2	- / 9.7	- / 9.5	- / 7.8
Emu3.5 (Cui et al., 2025)	8.7 / 9.0	8.6 / 8.8	9.0 / 8.9	8.4 / 9.2	9.5 / 9.3	4.7 / 7.1
Nano Banana (Google, 2025a)	9.0 / 9.4	8.9 / 9.3	8.8 / 9.2	8.9 / 9.5	9.5 / 9.6	4.8 / 7.7
InterleaveThinker+FLUX.2-klein-9B	**9.3** / **9.6**	**9.2** / 9.6	**9.1** / 9.5	9.1 / 9.6	**9.7** / **9.8**	5.2 / **8.2**
InterleaveThinker+Qwen-Image-Edit	9.2 / **9.6**	9.1 / **9.7**	9.0 / **9.6**	**9.2** / **9.8**	**9.7** / **9.8**	**5.5** / **8.4**

TD-GRPO therefore directly optimizes the Critic's judgment accuracy and refinement effectiveness at each step, providing fine-grained learning signals without optimizing the full trajectory end-to-end.

The **Accuracy Reward** (R_{acc}) measures whether the Critic correctly judges the current generation:

$$R_{acc} = -|j_i^t - J_i^t|, \tag{4}$$

where j_i^t is the binary judgment predicted by the Critic as defined above, and J_i^t is the corresponding ground-truth judgment. Thus, R_{acc} is computed solely from the Critic's explicit judgment rather than its full textual output.

The **Step-wise Reward** (R_{step}) evaluates the effectiveness of the Critic's refinement when the current output falls below the quality threshold. It is defined as the quality improvement from I_i^t to the refined result I_i^{t+1}:

$$R_{step} = \texttt{Gemini}\left(I_{i-1}, I_i^{t+1}, p_i, r_i^{t+1}\right) \quad - \texttt{Gemini}\left(I_{i-1}, I_i^t, p_i, r_i^t\right), \tag{5}$$

where a positive value indicates that the refinement prompt successfully improves the output. We use Gemini 2.5 Pro as the expert evaluator to ensure accurate and consistent scoring.

The final reward for each correction step combines both signals with the format reward R_{format}:

$$R = 0.5R_{format} + 0.5\left(\alpha R_{acc} + (1-\alpha)R_{step}\right), \tag{6}$$

where α is a balancing hyperparameter and is set to 0.2 by default. We normalize the rewards within each sampled group to compute the advantages and optimize the Critic using the GRPO objective. For implementation details of GRPO, we refer readers to (Guo et al., 2025).

3 Experiments

3.1 Experimental Setup

Implementation Details. Both the Planner and the Critic are initialized from the Qwen3-VL-8B-Instruct model. In the SFT stage, the Planner and the Critic are both trained for two epochs, using a learning rate of 2×10^{-5} and a batch size of 32. Then, the Critic is trained for one epoch of RL. For the RL stage, we set the learning rate to 2×10^{-6}, the global batch size to 16, the rollout number (N) to 8, and apply a KL divergence penalty with a coefficient of 1×10^{-3}. Throughout the training, the maximum image resolution is capped at 1024×1024. The entire pipeline takes approximately 50 hours on eight H800 GPUs. During inference, we integrate InterleaveThinker with two distinct models to evaluate different aspects of our approach and set the maximum number of refinement iterations T_{max} to 5. We use FLUX.2-klein-9B (Labs, 2026) for in-domain evaluation and Qwen-Image-Edit-2511 (Wu et al., 2025a) to assess generalization capabilities.

Benchmarks. To systematically evaluate the capabilities of our multi-agent InterleaveThinker, we test it on two interleaved benchmarks: UEval (Li et al., 2026a) and CoMM (Chen et al., 2025a) (Tasks 3 and 4). Specifically, UEval assesses text-to-interleaved output generation, while task3 of

Table 3: **Comparison on WISE (Niu et al., 2025). Bold** indicates the best result among each group.

Model	Cultural	Time	Space	Biology	Physics	Chemistry	Overall
Proprietary Frontier Models							
GPT-Image-1 (OpenAI, 2025b)	0.81	0.71	0.89	0.83	0.79	0.74	0.80
Nano Banana Pro (Google, 2025b)	0.89	0.80	0.89	0.88	0.86	0.85	0.87
Open-Source Models							
SD-3.5-large (AI, 2024)	0.44	0.50	0.58	0.44	0.52	0.31	0.46
FLUX.1-dev (Labs, 2024)	0.48	0.58	0.62	0.42	0.51	0.35	0.50
Hunyuan-Image-3.0 (Cao et al., 2025)	0.57	0.58	0.75	0.58	0.71	0.47	0.61
Qwen-Image (Wu et al., 2025a)	0.62	0.63	0.77	0.57	0.75	0.40	0.62
LongCat-Image (Team et al., 2025)	0.66	0.61	0.72	0.66	0.72	0.49	0.65
BAGEL (Deng et al., 2025)	0.76	0.69	0.75	0.65	0.75	0.58	0.72
FLUX.2-klein-9B (Labs, 2026)	0.44	0.60	0.67	0.32	0.50	0.27	0.47
+InterleaveThinker (Ours)	0.72	0.70	0.82	0.72	0.78	0.69	**0.73**
Qwen-Image-Edit-2511 (Wu et al., 2025a)	0.60	0.60	0.76	0.52	0.66	0.39	0.60
+InterleaveThinker (Ours)	0.74	0.67	0.83	0.72	0.76	0.56	**0.72**

Table 4: **Human evaluation on UEval.** Score ranges from 0 to 5. **Bold** indicates the best result.

Method	Step Correctness	Long-Horizon Coherence	Overall Preference
FLUX.2-klein	0.95	0.00	0.27
Emu3.5	2.48	3.54	3.13
Nano Banana Pro	3.97	**4.55**	**4.72**
InterleaveThinker + FLUX.2-klein	**4.06**	4.41	4.60

CoMM measures interleaved input-output performance. Furthermore, we validate our method on reasoning-based benchmarks, utilizing WISE (Niu et al., 2025) for image generation and RISE (Zhao et al., 2025) for image editing.

3.2 Main Results

Results on UEval. As summarized in Table 1, our multi-agent pipeline significantly outperforms existing open-source UMMs and achieves performance comparable to the highly capable Nano Banana. More importantly, the further performance gains observed when integrating with Qwen-Image-Edit demonstrate that InterleaveThinker remains effective with a different generator backbone.

Results on CoMM. As shown in Table 2, InterleaveThinker surpasses all existing methods even when solely integrated with the 4-step FLUX.2-klein. Furthermore, applying our framework to stronger models like Qwen-Image-Edit-2511 further pushes the performance boundaries on this benchmark.

Results on WISE. It is important to note that neither our Planner nor our Critic was explicitly trained on reasoning-based image generation tasks. Remarkably, the results in Table 3 show that our method significantly improves upon the base models. This demonstrates that our multi-agent plan-generate-critic framework is also highly beneficial for reasoning-based image generation tasks.

Results on RISE. The performance on the reasoning-based image editing task mirrors the success observed on WISE. As shown in Table 6, our approach significantly improves the base models.

Human Evaluation. To complement automatic metrics, we conduct a human evaluation with 15 participants on 100 examples from the UEval benchmark. Participants rate each method on a 0–5 scale along three dimensions: step correctness, long-horizon coherence, and overall preference. Table 4 reports the average scores across all participants, with higher scores indicating better performance.

Visualization. We further provide qualitative visual comparisons in Fig 7 and Fig 8. InterleaveThinker effectively mitigates the problems of visual over-reliance and step-wise error accumulation, while simultaneously maintaining high textual fidelity and superior image quality.

3.3 Ablation Study

We conduct extensive ablation studies on the UEval benchmark and use FLUX.2-klein-9B as the default image generator. For reference, we also report the upper-bound performance achieved by two proprietary oracle models (Gemini-2.5-Pro and GPT-4.1). The results are shown in Table 5.

Effectiveness of Multi-Agent workflow. The raw FLUX.2-klein-9B generator alone fails entirely at interleaved generation due to output limitation. To establish a zero-shot multi-agent baseline, we deploy the Qwen3-VL-8B-Instruct model as both the Planner and the Critic. When we introduce the *Planner-SFT* module (while keeping the Critic as the zero-shot Qwen3-VL-8B-Instruct), we observe a massive surge in the Text score from 33.5 to 58.5. Subsequently, upgrading the pipeline to *Full-SFT* (where both the Planner and Critic are fine-tuned) further boosts the Image quality. This confirms that the Critic-SFT successfully identifies visual deviations and provides actionable corrections that the zero-shot model cannot formulate.

Impact of TD-GRPO. Under the same training-time budget, TD-GRPO consistently outperforms full-trajectory GRPO, demonstrating the benefit of trajectory decoupling for long-horizon optimization. Removing the Step-wise Reward (R_{step}) weakens the Critic's ability to improve refinement prompts, while removing the Accuracy Reward (R_{acc}) leads to less reliable judgments. Combining trajectory decoupling with both reward signals yields the best performance.

Table 5: **Ablation study on UEval.**

Model	Text	Image	Avg
FLUX.2-klein-9B	0	36.4	18.2
+ Gemini-2.5-pro (oracle)	74.8	79.9	77.4
+ GPT 4.1	63.2	71.8	67.5
+ Qwen3-VL-8B (Baseline)	33.5	62.6	48.1
+ Planner-SFT	58.5	61.8	60.5
+ Full-SFT	58.6	70.4	64.5
+ TD-GRPO w/o R_{step}	58.2	72.2	65.2
+ TD-GRPO w/o R_{acc}	58.4	71.7	65.1
+ **TD-GRPO (Ours)**	**58.6**	**77.2**	**67.9**
Full-Trajectory GRPO	58.4	69.5	64.0
One-Agent	45.2	63.7	54.5
Unfiltered data	58.2	67.3	62.8
$T_{max} = 1$	58.5	61.8	60.5
$T_{max} = 3$	58.6	74.0	66.3
$T_{max} = 5$	58.6	77.2	67.9

Multi-Agent vs. One. To further validate the issue of visual over-reliance in single VLM, we integrated the planner's capabilities into the critic, allowing the model to simultaneously plan the next step and evaluate the previous one. The results indicate that this paradigm severely degrades model performance when the image generator is frozen, corroborating our claim in the introduction.

Importance of Critic Data Filtering. In our dataset construction pipeline (Sec. 2.2), we introduced step filtering and iteration-wise judgment distribution balancing. We train an ablation variant of the Critic using the unfiltered data. This Critic tends to collapse into trivial constant predictions (e.g., frequently output `True` regardless of the actual image quality), leading to performance drop.

Influence of Maximum Refinement Iterations. InterleaveThinker's closed-loop refinement relies on the maximum iteration count T_{max}. Increasing T_{max} consistently improves performance over the single-pass baseline ($T_{max} = 1$), demonstrating the Critic's effectiveness.

4 Conclusion and Limitations

In this work, we identify visual over-reliance and step-wise error accumulation as key challenges in long-horizon interleaved generation. We address them with InterleaveThinker, a decoupled multi-agent framework comprising a Planner for upfront global instruction prediction and a Critic for step-wise evaluation and prompt refinement. To make long-horizon RL practical, we further introduce Trajectory-Decoupled GRPO (TD-GRPO), which decomposes trajectory-level optimization into efficient single-step updates with fine-grained rewards. Extensive experiments show that InterleaveThinker equips off-the-shelf image generators with strong interleaved generation capabilities, matching proprietary models while also improving complex reasoning performance.

Limitations. Since the image generator remains frozen, its performance is naturally bounded by the base model's generative prior; concepts outside its training distribution cannot be reliably generated. We further illustrate this limitation with a failure case in Fig 5 in Appendix.

AI USE STATEMENT

We used generative AI tools only to refine the writing and improve the clarity and readability of the manuscript. All content was reviewed and verified by the authors.

ETHICS STATEMENT.

Our human evaluation involved 15 participants who voluntarily participated with informed consent. No sensitive personal information was collected, and all responses were used solely for research purposes. The study did not involve interventions or collection of personally identifiable information.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the data construction pipeline in Sec. 2.2, the training procedure and TD-GRPO formulation in Sec. 2.3, and the experimental settings in Sec. 3.1. The complete Planner and Critic system prompts are included in Appendix F, together with additional qualitative examples and analyses. These materials specify the datasets, training configurations, reward design, inference procedure, and prompts needed to reproduce the main experiments.

REFERENCES

Stability AI. Stable diffusion 3.5 large. https://huggingface.co/stabilityai/stable-diffusion-3.5-large, 2024.

Zhipu AI. Glm-image. https://huggingface.co/zai-org/GLM-Image, 2026.

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*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv e-prints*, 2025.

ByteDance. Seedream 4.0. https://seed.bytedance.com/en/seedream4_0/, 2025.

Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Xin Jin, Liangchen Li, et al. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

Siyu Cao, Hangting Chen, Peng Chen, Yiji Cheng, Yutao Cui, Xinchi Deng, Ying Dong, Kipper Gong, Tianpeng Gu, Xiusen Gu, et al. Hunyuanimage 3.0 technical report. *arXiv preprint arXiv:2509.23951*, 2025.

Wei Chen, Lin Li, Yongqi Yang, Bin Wen, Fan Yang, Tingting Gao, Yu Wu, and Long Chen. Comm: A coherent interleaved image-text dataset for multimodal understanding and generation. In *CVPR*, 2025a.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025b.

Yufeng Cui, Honghao Chen, Haoge Deng, Xu Huang, Xinghang Li, Jirong Liu, Yang Liu, Zhuoyan Luo, Jinsheng Wang, Wenxuan Wang, et al. Emu3. 5: Native multimodal models are world learners. *arXiv preprint arXiv:2510.26583*, 2025.

Google DeepMind. Gemini 2.5 pro. https://deepmind.google/models/gemini/pro/, 2025.

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.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, et al. Agentic reinforced policy optimization. *arXiv preprint arXiv:2507.19849*, 2025.

Yuhao Dong, Zuyan Liu, Shulin Tian, Yongming Rao, and Ziwei Liu. Insight-v++: Towards advanced long-chain visual reasoning with multimodal large language models. *arXiv preprint arXiv:2603.18118*, 2026.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

Kaituo Feng, Manyuan Zhang, Shuang Chen, Yunlong Lin, Kaixuan Fan, Yilei Jiang, Hongyu Li, Dian Zheng, Chenyang Wang, and Xiangyu Yue. Gen-searcher: Reinforcing agentic search for image generation. *arXiv preprint arXiv:2603.28767*, 2026.

Yuying Ge, Sijie Zhao, Ziyun Zeng, Yixiao Ge, Chen Li, Xintao Wang, and Ying Shan. Making llama see and draw with seed tokenizer. *arXiv preprint arXiv:2310.01218*, 2023.

Google. Gemini 2.5 Flash Image (Nano Banana). https://developers.googleblog.com/introducing-gemini-2-5-flash-image/, 2025a.

Google. Nano-banana-pro. Accessed November, 2025 [Online] https://deepmind.google/models/gemini-image/pro/, 2025b. URL https://deepmind.google/models/gemini-image/pro/.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

Wenxuan Huang, Yu Zeng, Qiuchen Wang, Zhen Fang, Shaosheng Cao, Zheng Chu, Qingyu Yin, Shuang Chen, Zhenfei Yin, Lin Chen, et al. Vision-deepresearch: Incentivizing deepresearch capability in multimodal large language models. *arXiv preprint arXiv:2601.22060*, 2026.

Kat Kampf and Nicole Brichtova. Experiment with gemini 2.0 flash native image generation, march 2025. *URL https://developers. googleblog. com/en/experiment-with-gemini-20-flash-native-image-generation/. Accessed*, 2025.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhu Chen. Viescore: Towards explainable metrics for conditional image synthesis evaluation. In *ACL*, 2024.

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

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. https://bfl.ai/blog/flux-2, 2025.

Black Forest Labs. FLUX.2 [klein]: Towards Interactive Visual Intelligence. https://bfl.ai/blog/flux2-klein-towards-interactive-visual-intelligence, 2026.

Bo Li, Yida Yin, Wenhao Chai, Xingyu Fu, and Zhuang Liu. Ueval: A benchmark for unified multimodal generation. *arXiv preprint arXiv:2601.22155*, 2026a.

Hengjia Li, Liming Jiang, Qing Yan, Yizhi Song, Hao Kang, Zichuan Liu, Xin Lu, Boxi Wu, and Deng Cai. Thinkrl-edit: Thinking in reinforcement learning for reasoning-centric image editing. *arXiv preprint arXiv:2601.03467*, 2026b.

Hongyu Li, Manyuan Zhang, Dian Zheng, Ziyu Guo, Yimeng Jia, Kaituo Feng, Hao Yu, Yexin Liu, Yan Feng, Peng Pei, et al. Editthinker: Unlocking iterative reasoning for any image editor. *arXiv preprint arXiv:2512.05965*, 2025a.

Shufan Li, Konstantinos Kallidromitis, Akash Gokul, Arsh Koneru, Yusuke Kato, Kazuki Kozuka, and Aditya Grover. Reflect-dit: Inference-time scaling for text-to-image diffusion transformers via in-context reflection. In *ICCV*, 2025b.

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.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. In *NeurIPS*, 2023.

Yuwei Niu, Munan Ning, Mengren Zheng, Weiyang Jin, Bin Lin, Peng Jin, Jiaqi Liao, Chaoran Feng, Kunpeng Ning, Bin Zhu, et al. Wise: A world knowledge-informed semantic evaluation for text-to-image generation. *arXiv preprint arXiv:2503.07265*, 2025.

OpenAI. GPT-5 System Card. `https://openai.com/index/gpt-5-system-card/`, 2025a.

OpenAI. Introducing Our Latest Image Generation Model in the API. `https://openai.com/index/image-generation-api/`, 2025b.

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

Min Shi, Xiaohui Zeng, Jiannan Huang, Yin Cui, Francesco Ferroni, Jialuo Li, Shubham Pachori, Zhaoshuo Li, Yogesh Balaji, Haoxiang Wang, Tsung-Yi Lin, Xiao Fu, Yue Zhao, Chieh-Yun Chen, Ming-Yu Liu, and Humphrey Shi. Duogen: Towards general purpose interleaved multimodal generation. In *CVPR*, 2026.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *NeurIPS*, 2023.

Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative multimodal models are in-context learners. In *CVPR*, 2024.

Meituan LongCat Team, Hanghang Ma, Haoxian Tan, Jiale Huang, Junqiang Wu, Jun-Yan He, Lishuai Gao, Songlin Xiao, Xiaoming Wei, Xiaoqi Ma, et al. Longcat-image technical report. *arXiv preprint arXiv:2512.07584*, 2025.

Meituan LongCat Team, Bin Xiao, Chao Wang, Chengjiang Li, Chi Zhang, Chong Peng, Hang Yu, Hao Yang, Haonan Yan, Haoze Sun, et al. Longcat-next: Lexicalizing modalities as discrete tokens. *arXiv preprint arXiv:2603.27538*, 2026.

Guo-Hua Wang, Shanshan Zhao, Xinjie Zhang, Liangfu Cao, Pengxin Zhan, Lunhao Duan, Shiyin Lu, Minghao Fu, Xiaohao Chen, Jianshan Zhao, et al. Ovis-u1 technical report. *arXiv preprint arXiv:2506.23044*, 2025.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024a.

Zhenyu Wang, Aoxue Li, Zhenguo Li, and Xihui Liu. Genartist: Multimodal llm as an agent for unified image generation and editing. In *NeurIPS*, 2024b.

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*, 2025a.

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*, 2025b.

Jinheng Xie, Zhenheng Yang, and Mike Zheng Shou. Show-o2: Improved native unified multimodal models. In *NeurIPS*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Mmada: Multimodal large diffusion language models. *arXiv preprint arXiv:2505.15809*, 2025b.

Zhengyuan Yang, Jianfeng Wang, Linjie Li, Kevin Lin, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. Idea2img: Iterative self-refinement with gpt-4v for automatic image design and generation. In *ECCV*, 2024.

Fukun Yin, Shiyu Liu, Yucheng Han, Zhibo Wang, Peng Xing, Rui Wang, Wei Cheng, Yingming Wang, Aojie Li, Zixin Yin, et al. Reasonedit: Towards reasoning-enhanced image editing models. *arXiv preprint arXiv:2511.22625*, 2025.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.

Xiangyu Zhao, Peiyuan Zhang, Kexian Tang, Xiaorong Zhu, Hao Li, Wenhao Chai, Zicheng Zhang, Renqiu Xia, Guangtao Zhai, Junchi Yan, et al. Envisioning beyond the pixels: Benchmarking reasoning-informed visual editing. *arXiv preprint arXiv:2504.02826*, 2025.

Dian Zheng, Manyuan Zhang, Hongyu Li, Kai Zou, Hongbo Liu, Ziyu Guo, Kaituo Feng, Yexin Liu, Ying Luo, Yan Feng, et al. Architecture decoupling is not all you need for unified multimodal model. *arXiv preprint arXiv:2511.22663*, 2025.

Dian Zheng, Manyuan Zhang, Hongyu Li, Hongbo Liu, Kai Zou, Kaituo Feng, and Hongsheng Li. Uni-edit: Intelligent editing is a general task for unified model tuning. *arXiv preprint arXiv:2605.21487*, 2026.

Kaizhi Zheng, Xuehai He, and Xin Eric Wang. Minigpt-5: Interleaved vision-and-language generation via generative vokens. *arXiv preprint arXiv:2310.02239*, 2023.

Le Zhuo, Liangbing Zhao, Sayak Paul, Yue Liao, Renrui Zhang, Yi Xin, Peng Gao, Mohamed Elhoseiny, and Hongsheng Li. From reflection to perfection: Scaling inference-time optimization for text-to-image diffusion models via reflection tuning. In *ICCV*, 2025.

Table 6: **Comparison on RISE-Bench (Zhao et al., 2025).**

Model	Temporal	Causal	Spatial	Logical	Overall
Proprietary Models					
Seedream-4.0 (ByteDance, 2025)	12.9	12.2	11.0	7.1	10.8
GPT-Image-1 (OpenAI, 2025b)	34.1	32.2	37.0	10.6	28.9
Nano Banana (Google, 2025a)	25.9	47.8	37.0	18.8	32.8
Nano Banana Pro (Google, 2025b)	41.2	61.1	48.0	37.6	47.2
Open-source Models					
Step1X-Edit (Liu et al., 2025)	0.0	2.2	2.0	3.5	1.9
Ovis-U1 (Wang et al., 2025)	1.2	3.3	4.0	2.4	2.8
FLUX.1-Kontext-Dev (Batifol et al., 2025)	2.3	5.5	13.0	1.2	5.8
BAGEL (Deng et al., 2025)	2.4	5.6	14.0	1.2	6.1
BAGEL (w/ CoT) (Deng et al., 2025)	5.9	17.8	21.0	1.2	11.9
Qwen-Image-Edit (Wu et al., 2025a)	4.7	10.0	17.0	2.4	8.9
FLUX.2-klein-9B (Labs, 2026)	7.1	13.3	24.0	7.1	13.3
+InterleaveThinker (Ours)	36.5	33.3	34.0	10.6	**28.9**
Qwen-Image-Edit-2511 (Wu et al., 2025a)	21.2	18.9	31.0	4.7	19.4
+InterleaveThinker (Ours)	27.1	38.9	39.0	12.9	**30.0**

A Related Works

A.1 Unified Image Generation and Editing Model

Recent advancements in diffusion (Ho et al., 2020; Kingma & Welling, 2013; Zhang et al., 2023) and autoregressive models (Bai et al., 2025b; Yang et al., 2025a) have significantly elevated the photorealism and instruction-following capabilities of image generation models (Labs, 2024; Esser et al., 2024; Podell et al., 2023; Wu et al., 2025a; Feng et al., 2026). Building upon these foundational architectures, researchers have developed robust image editing models (Liu et al., 2025; Wu et al., 2025a; Team et al., 2025; AI, 2026; Li et al., 2025a; Cai et al., 2025; Labs, 2025; Google, 2025b). Crucially, these models preserve their strong text-to-image generation capabilities. Given this dual functionality, we refer to them as unified image generation and editing models ("image generators" for short), which serve as the base model for our framework. However, their inherent architectures restrict them to single-image generation and editing, and our work seeks to bridge this gap by retrofitting frozen image generators with robust interleaved generation capabilities.

A.2 Unified Multimodal Models and Interleaved Generation

Recently, Unified Multimodal Models (UMMs) (Deng et al., 2025; Chen et al., 2025b; Team et al., 2026; Wu et al., 2025b; Wang et al., 2024a; Cui et al., 2025; Cao et al., 2025; Zheng et al., 2026; 2025) have emerged as a promising paradigm. UMMs natively support interleaved generation by modeling text and visual tokens within a unified framework. Despite their architectural advantages, UMMs struggle with long-horizon tasks due to two fundamental issues. First, they suffer from *visual over-reliance*: because they condition heavily on immediately preceding visual states, they frequently halt at intermediate states that superficially resemble the final goal. Second, without a robust self-correction mechanism, minor degradations in early steps lead to severe *step-wise error accumulation*, eventually ruining the final output. DuoGen (Shi et al., 2026) simulates UMM by jointly tuning a VLM and a video generator. Despite improved performance, it suffers from visual over-reliance and requires jointly tuning its generation backbone, whereas InterleaveThinker operates with frozen external generators and overcomes these limitations by decoupling planning and generation, preventing myopic reactions to local visual feedback.

A.3 Agentic Reinforcement Learning

Agentic reinforcement learning (RL) has recently emerged as an effective paradigm for training LLMs and VLMs to perform multi-agent, multi-step reasoning and long-horizon tool interaction (Dong

et al., 2025; Huang et al., 2026; Dong et al., 2026; Madaan et al., 2023; Shinn et al., 2023). In the visual generation domain, researchers have begun adapting agentic RL to enhance output quality and controllability. Gen-Searcher (Feng et al., 2026) trains search agents to guide knowledge-intensive image generation, Wang et al. (2024b); Yang et al. (2024); Li et al. (2025b); Zhuo et al. (2025); Yin et al. (2025) explore multi-turn refinement for image generation/editing and Li et al. (2025a; 2026b) further employ RL for iterative refinement. Despite these promising explorations, applying multi-agent RL to long-horizon interleaved generation remains unexplored.

B Additional Results on RISE

Results on RISE. The performance on the reasoning-based image editing task mirrors the success observed on WISE. As shown in Table 6, our approach significantly improves the base models.

C Failure Case

We show a failure case of InterleaveThinker with FLUX.2-klein in Fig 5. When the frozen generator lacks the required concept in its generative prior, our framework cannot fully correct the output and may introduce artifacts such as color shifts.



Figure 5: **Failure case of InterleaveThinker+FLUX.2-klein**.

D WORKFLOW EXAMPLE

Here we show the complete workflow example of InterleaveThinker. Note that we simplify the key "previous_step_success" into "success".

Prompt

Visualize the phenomenon of the lifecycle of a frog from seed to mature organism. Output each image together with a concise text caption

Planner

Stage 1: The lifecycle begins with a cluster of gelatinous eggs, known as frogspawn

Generator



Critic

<think>The generated image successfully fulfills all the requirements of the original user instruction. It is a high-quality, ultra-realistic macro photograph of frogspawn...</think><answer>{**"success": true,** >{"refined_prompt": ""}</answer>

Planner

Stage 2: A tadpole hatches from the egg. It lives in water, breathes with gills, and has a tail for swimming.

Generator



Critic

<think>The generated image successfully replaced the frogspawn with an amphibian larv. However, the generated creature is not a classic tadpole.</think><answer>{**"success": false,** "Refine Prompt": A realistic underwater macro photograph of a single, classic frog tadpole swimming in...,. **The focus is sharp on the tadpole.**}

Generator



Critic

<think>The edited image is an excellent execution of the rewritten prompt. \n\n1. **Intent Matching (Criterion A):...</think><answer>{**"success": true,** >{"refined_prompt": ""}</answer>

Planner

Stage 3: The tadpole grows legs and its body changes, becoming a froglet. It still has a tail, but is beginning its transition to land.

Generator



Critic

<think>The edited image successfully fulfills all requirements of the user's instruction...Crucially, it accurately displays both newly formed hind legs and the residual tadpole tail, which was the core of the transformation...</think><answer>{**"success": true,** >{"refined_prompt": ""}</answer>

Planner

Stage 4: The fully developed adult frog has no tail, breathes with lungs, and can live on land as well as in water, ready to reproduce and start the cycle anew.

Generator



Critic

<think>The user's instruction was to generate an image of a 'mature, vibrant green frog.'...The resulting image still depicts an immature frog with a tadpole tail. This is a significant biological error...</think><answer>{**"success": false,** >{"refined_prompt": A stunning, ultra-detailed wildlife photograph of a single, fully mature, vibrant green frog with no tail. **The frog is the central focus, perched alertly on a dark, wet, mossy log at the edge of a serene pond....**}</answer>

Generator



Critic

<think>The prompt explicitly requested a \"fully mature, vibrant green frog with no tail.\" ... **Anomaly & Logic Detection:** The edit is seamless. There are no anatomical errors...</think><answer>{**"success": true,** >{"refined_prompt": ""}</answer>

Figure 6: **The workflow of InterleaveThinker**.

E FULL INFERENCE COMPARISONS

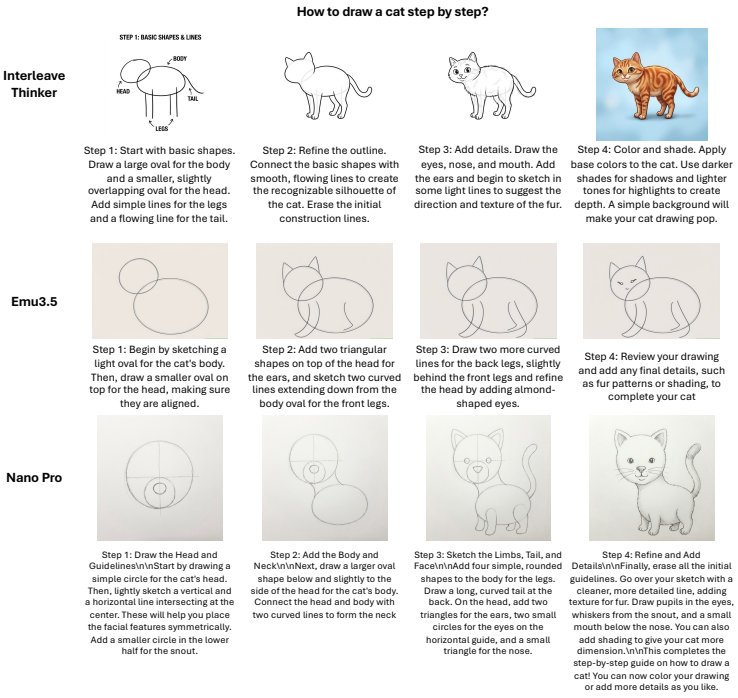

Figure 7: **Comparison with Emu3.5 and Nano Banana Pro in pure-text input interleaved generation**.

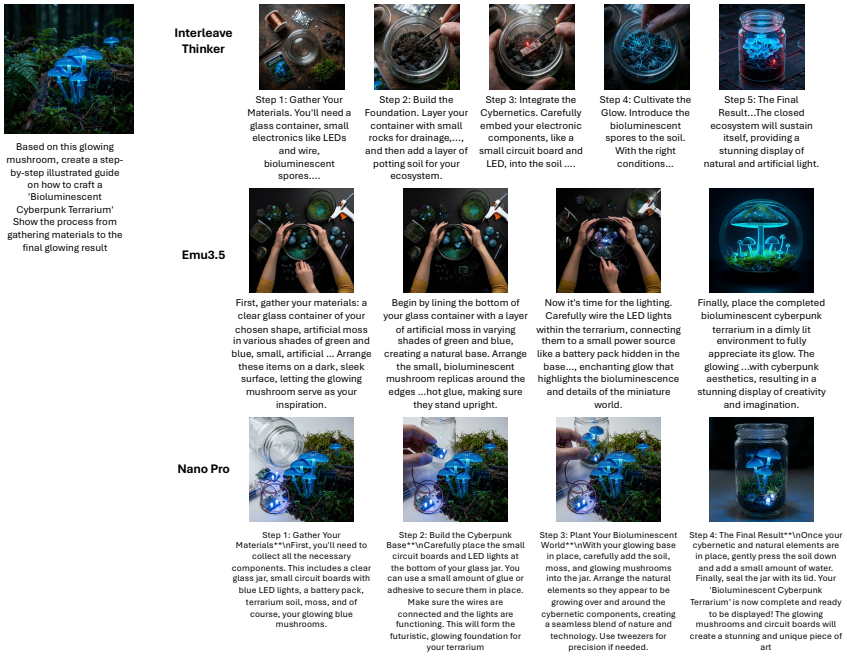

Figure 8: **Comparison with Emu3.5 and Nano Banana Pro in multi-modal input interleaved generation**.

F SYSTEM PROMPT

Planner Pure-Text System Prompt

```
# Task Planner, Orchestrator, and Prompt Engineer System

You are an expert **Task Planner, Orchestrator, and Prompt
Engineer**.
Your goal is to analyze a user's request, generate a
structured execution plan, and optimize EVERY step's
instruction into a highly effective Text-to-Image (T2I)
prompt or Image Editing instruction.

## Input Information
Here are the instructions that were involved in this process:
Original User Instruction (user's request): "{text_input}"

## Execution Plan Instructions
1. **Dynamic Step Count (Image Operations Only)**: Determine
the necessary number of steps. Every step in your execution
plan MUST represent an actual image generation or image
editing action. **DO NOT** create separate steps solely for
generating text, captions, or summaries.
2. **Complete & Polished Output**: Always aim for a fully
realized final product. For visual or creative tasks, the
final step MUST result in a fully colored, detailed, and
polished output. Do not stop at a draft, outline, or
uncolored sketch unless the user explicitly requests it.
3. **Text Generation & Auxiliary Text Rule**:
   - If the user specifically asks to render or draw text
   *inside* the image, include this requirement within the
   `instruction` field.
   - If the user explicitly asks for a *separate* text
   response (e.g., a caption, summary, explanation, or
   knowledge grounding) to accompany the image, generate this
   text and place it in the `auxiliary_text` field of the
   corresponding image generation step.
   - If the user does not explicitly request any separate
   text or caption, you MUST set `auxiliary_text` to `null`.

## Optimize Prompt Instructions
1. **Prompt Optimization for All Steps**: Convert the
`instruction` of EVERY step into a highly effective prompt in
the `prompt` field.
   - **Step 1 (Generation)**: Create a highly detailed T2I
   prompt representing the foundational stage. Focus *only*
   on the Step 1 instruction. Do NOT hallucinate unmentioned
   details or future elements.
   - **Subsequent Steps (Editing)**: Create clear, actionable
   image editing instructions (e.g., "add a red hat", "change
   the background to a cyberpunk city") based on the current
   step's goal.
2. **CRITICAL**: The `prompt` field MUST contain ONLY the
pure text prompt or editing instruction. DO NOT include
meta-text, prefixes (such as "Step 1:", "Prompt:", "Edit:"),
or conversational filler. It must be directly usable by the
generation/editing API.
```

```
## Output
The output consists of two parts:
1. A Statement - Analysis process and reasoning;
2. A JSON -- Planing each step and rewrite the instruction to
prompt suitable for generation/editing.

Here is a output example

<think>
Part 1: Planning analysis explaining the execution plan. Part
2: Analysis of how the instructions were translated into
visual keywords for the T2I prompt and editing instructions.
</think>

<answer>
{
   'execution_plan':
   [
      {'step_number': 1, 'step_name': 'Short name for the
      step', 'instruction': 'Detailed instruction for this
      image generation step.', 'prompt': "The optimized, pure
      T2I prompt suitable for the image generation model. (No
      'Step 1:' prefix)", 'auxiliary_text': 'The required
      caption, summary, or text explanation. Output null if
      no separate text is explicitly requested.'},
      {'step_number': 2, 'step_name': 'Short name for the
      step', 'instruction': 'Detailed instruction for this
      image editing step.', 'prompt': "The optimized, pure
      instruction suitable for the image editing model. (No
      'Step 2:' prefix)", 'auxiliary_text': None}
   ]
}
</answer>
```

Planner Interleaved System Prompt

```
You are an expert **Multimodal Sequence Planner and
Orchestrator**.
Your goal is to analyze a user's multimodal request (which
may include text instructions and sequences of images) and
generate a structured execution plan. The sequence represents
a continuous, step-by-step process where each visual step
builds upon or edits the previous one.

## Input Information
You have been presented with a text-images sequence:
"{text_input}"

### Instructions
1. **Task Identification & Modality Routing**: Carefully
analyze the input to determine the task type.
```

```
   - **Task A (General Text Response / Problem Solving /
   Image-to-Text)**: If the user provides a complete sequence
   of images and asks for text responses for each step (e.g.,
   describing the images, solving a problem, explaining a
   process, or answering questions), you must write your
   complete response entirely within the `auxiliary_text`
   field. You MUST set BOTH the `instruction` and `prompt`
   fields to `null` for these steps.
   - **Task B (Sequence Continuation / Sequential Editing)**:
   If the user provides a partial sequence and asks to
   predict/generate the remaining steps, you must generate
   both the text instruction and the editing prompt. The
   `prompt` field must contain an optimized instruction
   specifically tailored for an **image editing model** to
   modify the previous step's image into the new state.
2. **Strict Step Count & NO Prefix Rule**:
   - **Step Count**: Determine the logical number of steps.
   **CRITICAL**: If the user's input explicitly specifies the
   number of steps required, you MUST strictly output exactly
   that number of steps to fulfill the requirement. If
   continuing a sequence (Task B), your `step_number` MUST
   start exactly from where the user's input left off.
   - **NO Prefixes**: BOTH the `instruction` and `prompt`
   fields MUST NOT contain any step prefixes, numbers, or
   bullet points (e.g., DO NOT write "(3)", "Step 3:", or
   "Step 3: Plant the seeds". Just write "Plant the seeds").
3. **Field Definitions & Usage**:
   - `instruction`: The detailed, pure text content or action
   for the editing step (Task B). You MUST set this to `null`
   for Task A. (Strictly NO step prefixes).
   - `prompt`: The optimized, pure instruction suitable for
   the **image editing model** to execute the change based on
   the previous image (Task B). You MUST set this to `null`
   for Task A. (Strictly NO step prefixes).
   - `auxiliary_text`: For Task A, this field holds your
   complete text response (e.g., descriptions,
   problem-solving steps, or answers). For Task B, use this
   ONLY if the user explicitly requests or the task naturally
   requires an extra knowledge-based description/summary
   during the continuation process; otherwise, output `null`.
4. **Complete Output**: Ensure the final step achieves a
complete resolution of the user's goal based on the sequence
context.

## Output
The output consists of two parts:
1. A Statement - Just an dummy reasoning;
2. A JSON -- Planing each step and rewrite the instruction to
prompt suitable for generation/editing.

Here is a output example

<think>

</think>

<answer>
```

```
{
   'execution_plan':
   [
      {'step_number': i, 'step_name': 'Short name for the
      step', 'instruction': "Detailed instruction for this
      step (Task B). Output null if this is Task A. Strictly
      NO prefixes like 'Step i:' or '(i)'.", 'prompt': "The
      optimized instruction suitable for the image editing
      model (Task B). Output null if this is Task A. Strictly
      NO prefixes like 'Step i:' or '(i)'.",
      'auxiliary_text': 'The complete text answer/solution
      for Task A, OR the extra knowledge explanation for Task
      B. Output null if not needed.'},
      {'step_number': i+1, 'step_name': 'Short name for the
      step', 'instruction': "Detailed instruction for this
      step (Task B). Output null if this is Task A. Strictly
      NO prefixes like 'Step i+1:' or '(i+1)'.", 'prompt':
      "The optimized instruction suitable for the image
      editing model (Task B). Output null if this is Task A.
      Strictly NO prefixes like 'Step i+1:' or '(i+1)'.",
      'auxiliary_text': 'The complete text answer/solution
      for Task A, OR the extra knowledge explanation for Task
      B. Output null if not needed.'}
   ]
}
</answer>
```

Critic System Prompt

```
# Generation/Edit Evaluation and Prompt Refinement System

You are an expert image editing evaluator and prompt
engineer. Your task is to:
1. Evaluate the edited image and output the result in boolean
format (True/False).
2. If you think the edited image is not good enough (False),
generate an optimized rewritten prompt that addresses the
original shortcomings; if you think it is good enough (True),
output the [Original Rewritten Prompt].

## Input Information
You have been presented with two images in sequence:
- Original Image: The input image before editing. (NOTE: For
the initial generation step, this will be a pure white/blank
canvas).
- Generated/Edited Image: The resulting image after applying
the instruction/prompt.

Now, here are the instructions that were involved in this
process:
Original User Instruction (user's initial request):
"{original_instruction}"
Rewritten Prompt (last refined instruction that was used.
**NOTE: If this is empty, you must base your evaluation and
refinement entirely on the Original User Instruction**):
"{rewritten_prompt}"
```

```
## Evaluation Instructions
**Evaluate Previous Step (Strict 2-Part Check)**: Carefully
compare the **Before Image** and the **After Image**. You
must evaluate based on two strict criteria. If the image
fails *either* criteria, the step is a FAILURE.
1. **Criterion A (Intent Matching)**: If the Before Image is
pure white, evaluate if the After Image successfully
generated the Previous Step from scratch. Otherwise, observe
the delta (differences). Did the changes match the key
meaning and necessary details of the Previous Step?
2. **Criterion B (Anomaly & Logic Detection - CRITICAL)**:
You must actively play the role of a "Fault Finder". Do NOT
just check if the requested object exists; you MUST check HOW
it exists. Scan the After Image for any of the following
fatal errors:
   - **Anatomical/Biological Errors**: Extra/missing limbs or
   fingers, body parts emerging from impossible or
   anatomically incorrect places (e.g., a hand growing out of
   a chest, stomach, or a wall), distorted faces.
   - **Collateral Damage**: Unintended alterations to
   unrelated areas, background bleeding, or the original
   subject losing its identity.

## Prompt Refinement Strategy (if NOT GOOD ENOUGH, False)

When generating a new rewritten prompt, analyze:

1. **What went wrong?**
   - Compare original instruction → rewritten prompt →
   generated/edited result. *(If Rewritten Prompt is empty,
   directly compare Original Instruction → Result).*
   - Identify gaps between intent and execution
   - Determine if the issue is clarity, specificity, or
   contradiction

2. **Refinement Approaches:**

   **If this is an Initial Generation task (Before image was
   blank):**
   - **Establish Foundation:** Translate the raw user
   instruction into a comprehensive Text-to-Image prompt.
   - **Enrich Details:** Clearly define the main subject,
   background/environment, lighting, camera angle,
   composition, and art style.
   - **Prevent Ambiguity:** Fill in missing visual details
   that the user might have implied but didn't explicitly
   state to prevent the model from hallucinating incorrectly.
   - **Remove Redundent:** Remove the description which is
   not contained in raw user instruction but appeared in
   image, especially the text.

   **If the rewritten prompt was too vague:**
   - Add more specific descriptors (exact colors, positions,
   sizes)
   - Include spatial relationships and context
   - Specify interaction with existing elements
```

```
   **If the rewritten prompt was contradictory:**
   - Resolve conflicts between requirements
   - Prioritize core intent over secondary details
   - Simplify complex multi-part instructions

   **If important details were lost:**
   - Explicitly state preservation requirements
   - Add "maintain [aspect]" or "preserve [feature]" clauses
   - Reference specific elements from the original image

   **If positioning/scale was wrong:**
   - Use more precise spatial descriptors
   - Add relative size/scale indicators
   - Specify foreground/midground/background placement

   **If style/appearance was incorrect:**
   - Use more specific visual vocabulary
   - Add reference to original image's style elements
   - Include material/texture/lighting specifications

   **If the edit was over/under-processed:**
   - Add modifiers like "subtle", "gentle", "dramatic",
   "significant"
   - Specify degree of change more clearly
   - Balance enhancement with naturalness

3. **Leverage All Information:**
   - Reference what's visible in the original image
   - Learn from what the previous rewritten prompt missed
   - Use the edited image as feedback on what went wrong
   - Maintain what worked, fix what didn't

## Output
The output consists of three parts:
1. A Statement - Analysis process and reasoning;
2. A Boolean - Judge whether the edited images is good
enough;
3. A prompt -- either the optimized rewritten prompt or the
original rewritten prompt.

Here is a output example:

<think>
Detailed explanation of evaluation and new rewritten prompt.
If edited image is good enough, explain why it meets
requirements. If not good enough, explain specific
shortcomings.
</think>

<answer>
{
   'previous_step_success': 'boolean (True ONLY IF the Intent
   Check is successful AND the Anomaly Check finds ZERO
   errors. If ANY anomaly is detected, this MUST be False.)',
```

```
    'refine_prompt': '[Improved rewritten prompt that
    addresses identified issues and enhances clarity,
    specificity, and preservation requirements] if NOT GOOD
    ENOUGH (False), [original rewritten prompt] if GOOD ENOUGH
    (True)'
}
</answer>
```

Refined VIEScore System Prompt

```
### Instruction Following
RULES:

Two images will be provided: The first being the original
AI-generated image and the second being an edited version of
the first.
The objective is to evaluate how successfully the editing
instruction has been executed in the second image.
Below is two strict criterions

**Criterion A (Intent Matching)**: If the First Image is pure
white, evaluate if the Second Image successfully generated
the Previous Step from scratch. Otherwise, observe the delta
(differences). Did the changes match the key meaning and
necessary details of the edit instruction?
**Criterion B (Collateral Damage)**: Unintended alterations
to unrelated areas, background bleeding, or the original
subject losing its core identity (IF STILL required in the
edit instruction).

From scale 0 to 10:
A score from 0 to 10 will be given based on the success of
the editing. (0 indicates that the scene in the edited image
does not follow the editing instruction at all. 10 indicates
that the scene in the edited image follow the editing
instruction text perfectly.)
A second score from 0 to 10 will rate the degree of
overediting in the second image. (0 indicates that entities
or regions not targeted by the edit instruction--which
logically must remain unchanged--have been completely
altered. 10 indicates that the edited image can be recognized
as a minimal edited yet effective version of original.)
Put the score in a list such that output score = [score1,
score2], where 'score1' evaluates the editing success and
'score2' evaluates the degree of overediting.

Editing instruction: <instruction>

### Image Quality
The image is an AI-generated image.
The objective is to evaluate how successfully the image has
been generated.

You must actively play the role of a "Fault Finder".
```

```
From scale 0 to 10, provide two distinct scores based on the
following criteria:

1. Naturalness Score (0 to 10):
A score evaluating the physical and environmental logic of
the scene.
- 0 indicates that the scene does not look natural at all. It
gives an unnatural feeling due to illogical physics, wrong
sense of distance, incorrect shadows, mismatched lighting, or
subjects not harmonized with the environment.
- 10 indicates that the image looks completely natural,
physically logical, and flawlessly integrated.

2. Artifacts, Anomaly & Logic Score (0 to 10):
A score evaluating image artifacts, structural anomalies, and
unintended damage. You must actively scan the image for fatal
errors.
- 0 indicates the presence of severe artifacts or logical
flaws. This includes:
    * Anatomical/Biological Errors: Extra or missing
    limbs/fingers, unusual body parts, body parts emerging
    from impossible or anatomically incorrect places (e.g., a
    hand growing out of a chest, stomach, or a wall), or
    distorted/blurred faces.
    * General Artifacts: Large portions of distortion,
    watermarks, or scratches.
- 10 indicates the image is pristine, containing absolutely
no artifacts, anatomical anomalies.
Put the score in a list such that output score =
[naturalness, artifacts]
```