
JarvisArt: Liberating Human Artistic Creativity via an Intelligent Photo Retouching Agent

Yunlong Lin^{1*} Zixu Lin^{1*} Kunjie Lin^{1*} Jinbin Bai⁵ Panwang Pan⁴ Chenxin Li³
Haoyu Chen² Zhongdao Wang⁶ Xinghao Ding^{1†} Wenbo Li^{3♣} Shuicheng Yan^{5♠}

¹ Key Laboratory of Multimedia Trusted Perception and Efficient Computing,
Ministry of Education of China, Xiamen University, Xiamen, Fujian, China

² The Hong Kong University of Science and Technology (Guangzhou)

³ The Chinese University of Hong Kong ⁴ Bytedance

⁵ National University of Singapore ⁶ Tsinghua University

Project Page: <https://jarvisart.vercel.app/>

Abstract

Photo retouching has become integral to contemporary visual storytelling, enabling users to capture aesthetics and express creativity. While professional tools such as Adobe Lightroom offer powerful capabilities, they demand substantial expertise and manual effort. In contrast, existing AI-based solutions provide automation but often suffer from limited adjustability and poor generalization, failing to meet diverse and personalized editing needs. To bridge this gap, we introduce JarvisArt, a multi-modal large language model (MLLM)-driven agent that understands user intent, mimics the reasoning process of professional artists, and intelligently coordinates over 200 retouching tools within Lightroom. JarvisArt undergoes a two-stage training process: an initial Chain-of-Thought supervised fine-tuning to establish basic reasoning and tool-use skills, followed by Group Relative Policy Optimization for Retouching (GRPO-R) to further enhance its decision-making and tool proficiency. We also propose the Agent-to-Lightroom Protocol to facilitate seamless integration with Lightroom. To evaluate performance, we develop MMArt-Bench, a novel benchmark constructed from real-world user edits. JarvisArt demonstrates user-friendly interaction, superior generalization, and fine-grained control over both global and local adjustments, paving a new avenue for intelligent photo retouching. Notably, it outperforms GPT-4o with a **60%** improvement in average pixel-level metrics on MMArt-Bench for content fidelity, while maintaining comparable instruction-following capabilities.

1 Introduction

Photo retouching is fundamental to modern photography, enabling users to manipulate exposure, color, contrast, and tone for expressive, high-quality images. Commercial tools such as Adobe Lightroom and PicsArt offer extensive manual controls but demand specialized expertise and significant time investment, creating barriers for non-experts. Existing automated methods—including zero- and first-order optimization [15, 44, 54, 65], reinforcement learning [55, 29, 28], and diffusion-based editing [68, 2, 57]—improve automation yet remain limited in stylistic diversity, fine-grained adjustment, and scene generalization. More recently, instruction-guided multimodal models such as GPT-4o [24] and Gemini-2-Flash [52] have enabled natural-language-driven editing but frequently compromise content fidelity, intricate attribute control, and high-resolution support.

* Equal Contributions. ♣ Project Leader † Corresponding Author. ♠ Supervised the research.

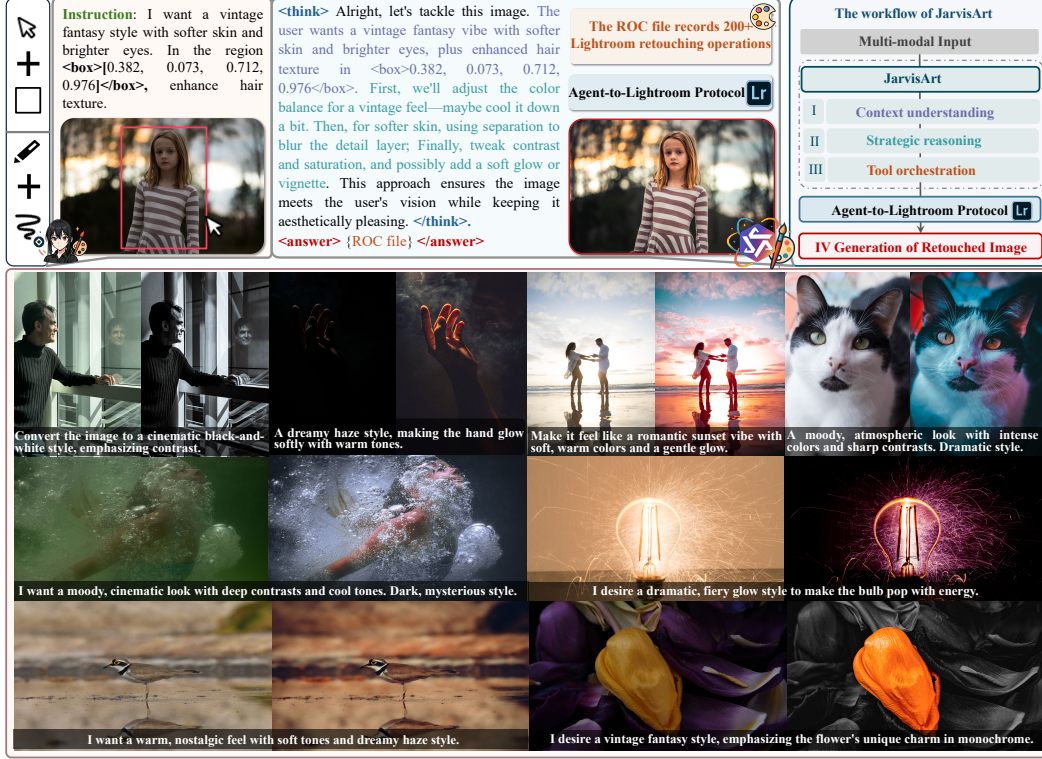


Figure 1: JarvisArt supports multi-granularity retouching goals, ranging from scene-level adjustments to region-specific refinements. Users can perform intuitive, free-form edits through natural inputs such as text prompts, bounding boxes, or brushstrokes. Furthermore, users can edit any-resolution images with JarvisArt. **Purple**: multi-modal context understanding. **Green**: retouching strategy reasoning. **Orange**: decision-making in tool orchestration.

LLM [13, 4, 60, 33]-powered agents have driven breakthroughs in autonomous task execution and problem solving, inspiring us to explore a novel photo-retouching paradigm: *an intelligent, user-friendly artist agent that interprets the user’s intent and delivers professional-level edits*. To this end, we introduce JarvisArt, which (1) accurately parses visual inputs and natural-language instructions, (2) embeds professional retouching expertise to emulate an artist’s reasoning, (3) efficiently manages over 200 Lightroom operations, and (4) supports both global and local adjustments through an intuitive interface. All planning and tool invocations are fully transparent, allowing users to interactively refine the retouching workflow to suit their individual preferences.

To translate this vision into practice, we must tackle three core challenges: the scarcity of high-quality data (source/target images, textual instructions, and editing configurations), the need for expert-level reasoning strategies, and the absence of a standardized Agent-to-Lightroom integration protocol. To overcome these, we first design a data-generation pipeline that yields the MMArt-55K dataset, comprising 5K standard and 50K Chain-of-Thought-enhanced multi-granularity samples. Next, we employ a two-stage post-training regime: (1) supervised fine-tuning (SFT) to instill a workflow of “understanding → reasoning → decision-making (recording Lightroom operations into a ROC file)”, as illustrated in Figure 1, and (2) Group Relative Policy Optimization for Retouching (GRPO-R) augmented with multi-dimensional tool-use rewards—namely, retouching-operation accuracy (evaluating both global and region-specific parameter prediction) and perceptual quality (assessing the visual fidelity of retouched outputs)—to refine decision-making and generalization. Finally, we introduce the Agent-to-Lightroom (A2L) protocol to enable seamless, automated Lightroom editing with bidirectional feedback. Consequently, JarvisArt deeply understands the intent of the user, generates diverse stylistic renditions, and seamlessly executes global and region-specific adjustments to produce visually compelling results (see Figure 1).

Our contributions can be summarized as follows:

- We introduce JarvisArt, an intelligent artist agent powered by an MLLM and linked to over 200 Lightroom operations, capable of producing diverse, user-driven stylistic edits that surpass current automated methods and rival professional human retouchers.
- We design a scalable data-synthesis pipeline to construct the MMArt dataset, comprising 5K standard instruction-based and 50K Chain-of-Thought-enhanced multi-granularity samples for detailed retouching tasks.
- We develop a two-stage post-training regime: SFT followed by GRPO-R with tailored tool-use rewards to enhance the agent’s reasoning, tool proficiency, and generalization.
- We establish an Agent-to-Lightroom communication protocol that enables seamless collaboration between JarvisArt and Lightroom, facilitating fully automated editing workflows.

2 Related Work

Photo Retouching. Existing automated pipelines have been proposed to streamline manual retouching. Zeroth- and First-order optimizations [15, 44, 43, 6, 54, 65, 53] were early attempts, but they are constrained by limited parameter prediction and reliance on pre-trained proxies. RL-based methods [55, 29, 28, 19, 45] attempt to mimic human workflows and offer some transparency but fail to capture artistic vision and lack deeper user interaction. Diffusion models [68, 2, 57, 75, 7, 59, 10] dominate high-fidelity image synthesis but rely on static prompts and lack multi-turn reasoning or flexible language alignment, limiting open-ended editing. Additionally, recent unified image editing models have achieved dual breakthroughs in comprehension and generation. Notable examples include closed-source models like GPT-4o [24] and Gemini-2-Flash [52], as well as open-source models such as Bagel [9], Janus-Pro [8], UniTok [41], QLIP [73], and VARGPT-v1.1 [76]. Despite these breakthroughs, three key limitations remain: (1) destructive editing by regenerating all pixels, compromising content preservation; (2) lack of interactive and interpretable local attribute control (e.g., softening or brightening skin); and (3) the absence of arbitrary-resolution editing due to generative model architectural constraints. Conversely, our study presents an interactive and interpretable retouching paradigm that integrates multimodal understanding with expert-level editing tools for non-destructive photo retouching. JarvisArt empowers users through a human-agent collaboration loop, enabling scene-level edits alongside precise region-specific tweaks-blending creative flexibility with the rigor of a professional workflow.

Reinforcement Fine-Tuning. Rule-based reinforcement fine-tuning, as demonstrated by OpenAI’s o1 [25] and Deepseek-R1 [13], has shown impressive performance in tasks such as mathematical reasoning [4, 25, 61, 64], and code generation [22, 26, 67, 70]. Subsequent research has extended this approach to multimodal models, designing task-specific reward functions for visual perception tasks. These include correct class prediction in image classification [46, 5, 42], Intersection-over-Union (IoU) metrics in image localization and detection [39, 21, 62, 49], accurate click position prediction in GUI grounding tasks [40, 56], and effective interaction with search engines to leverage up-to-date external information [27]. However, unlike these tasks with a single correct answer, our task involves tool-integrated retouching, which requires predicting multiple tools and their parameters. Designing effective reward signals to support learning in this setting remains an open and underexplored challenge. In this paper, we propose customized tool-use rewards, enabling JarvisArt to equip advanced artistic reasoning and tool invocation capabilities.

LLM-Empowered Agent. LLM-powered agents have revolutionized AI systems due to three key developments: 1) unprecedented reasoning capabilities of LLMs [13, 4, 60, 16, 17]; 2) advancements in tool manipulation and environmental interaction [31, 51, 18, 20, 38, 34, 35, 11] and 3) sophisticated memory architectures that support longitudinal experience accumulation [14, 71, 58, 63]. Despite these advancements, three fundamental limitations persist when applying LLM agents to professional photo retouching: 1) the absence of a domain-specific retouching knowledge base, which hinders accurate interpretation of user intent, 2) limited decision-making abilities in selecting suitable tools and determining precise parameter values, and 3) absence of standardized protocols to ensure compatibility with professional retouching software integrations. To address these limitations, we propose JarvisArt, a powerful artistic agent that integrates three core capabilities: (1) professional retouching expertise for precise understanding of user instructions, (2) proficiency with commercial retouching tools in Lightroom, and (3) standardized communication protocols for seamless Lightroom integration.

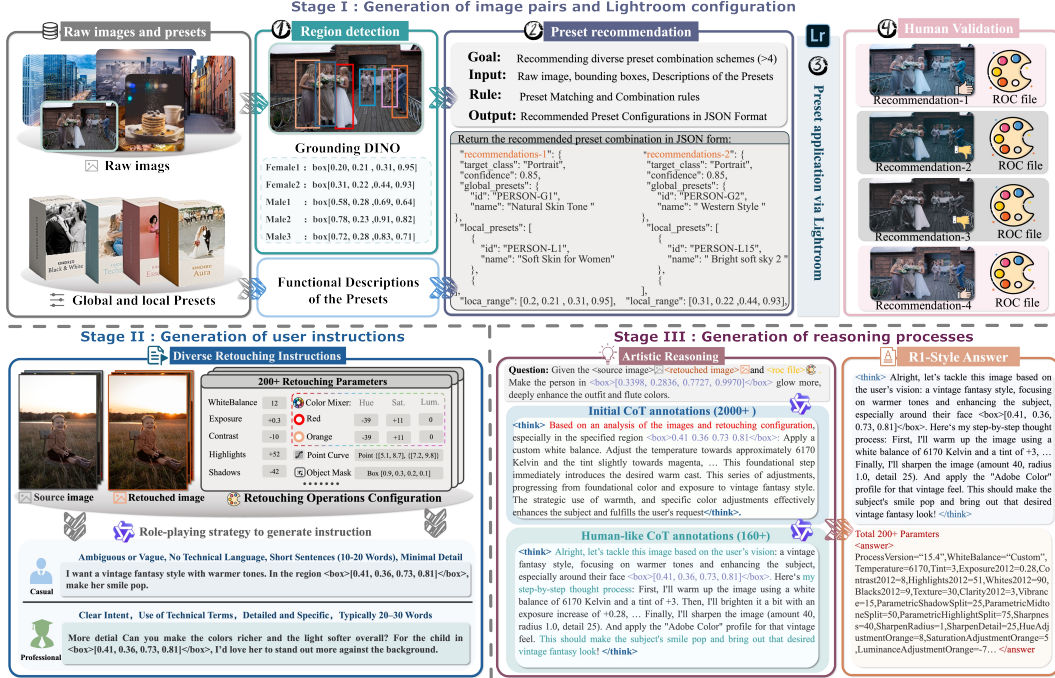


Figure 2: The data generation pipeline comprises three main stages: 1) Curation of diverse source–target examples covering varied scenes and styles with corresponding Lightroom configurations; 2) Generation of diverse user instructions that reflects different creative intents; 3) Production of Chain-of-Thought traces that simulate a human artist reasoning process.

3 Method

We begin by outlining the overall workflow of JarvisArt (Sec. 3.1). Next, we introduce a comprehensive data generation pipeline that constructs MMArt, a high-quality dataset comprising instruction and reasoning samples for agentic photo retouching tasks (Sec. 3.2). Finally, we investigate the core components of JarvisArt (Sec. 3.3), including a two-stage post-training pipeline and the Agent-to-Lightroom (A2L) protocol, which allows seamless collaboration between JarvisArt and Lightroom.

3.1 Overview

JarvisArt is an interactive, MLLM-based photo-retouching system that supports both scene-level and region-level edits. In addition to textual instructions, users can specify local areas via free-form brushstrokes or draggable bounding boxes. In Figure 1, JarvisArt’s pipeline comprises three stages: 1) Multi-modal context understanding to parse user directives, image content, and regions of interest; 2) Strategic reasoning grounded in photographic principles to formulate a retouching plan; and 3) Tool orchestration to select appropriate Lightroom operations and parameters. These operations are executed automatically through the A2L protocol. Formally, JarvisArt implements a function:

$$f(Q, I_{src}) \rightarrow \mathcal{T} = \{t_1, t_2, \dots, t_n\},$$

where Q is the user query, I_{src} the source image, and each t_i denotes a specific Lightroom edit (e.g., exposure+0.03). The final output is obtained by $I_{edit} = g(I_{src}, \mathcal{T})$, with $g(\cdot)$ representing Lightroom’s execution environment.

3.2 Data Generation Pipeline

We design a three-stage data-generation pipeline (Figure 2) to construct MMArt with explicit Chain-of-Thought (CoT) annotations. Each sample is a five-tuple $\langle I_{src}, I_{tgt}, Q, \mathcal{C}, O \rangle$, where I_{src} and I_{tgt} are the before-/after-retouch images, Q the user’s instruction, $\mathcal{C}$ the CoT reasoning wrapped in <think> tags, and O the retouching operation configuration (ROC) file of tool invocations and parameters

within `<answer>` tags. The pipeline proceeds as follows: 1) Curation of diverse source–target examples covering varied scenes and styles, and the corresponding Lightroom configurations; 2) Generation of natural-language instructions that reflect user intents; 3) Production of step-by-step reasoning traces. Further statistics and examples of MMart can be found in Appendix A.

Stage I: Generation of image pairs and Lightroom configuration. We source raw images from PPR10K [32], the Adobe Lightroom community, and licensed open-source collections, then curate a diverse library of global and local artistic presets². Leveraging Qwen2.5-VL-72B [60] for multimodal role-playing and Grounding DINO [37] for precise region localization, we simulate expert-level edits in four steps: 1) *Region detection*, in which Grounding DINO [37] identifies regions of interest (confidence > 0.8); 2) *Preset recommendation*, where Qwen2.5-VL-72B [60] proposes global and local presets based on image aesthetics; 3) *Preset application*, applying each recommendation in Lightroom to generate five candidate retouched images; and 4) *Human-in-the-loop validation*, selecting the most artistically pleasing outputs. Each finalized sample comprises $\langle I_{\text{src}}, I_{\text{tgt}}, O \rangle$, denoting the source image, the retouched image, and the detailed record of Lightroom operations. The role-playing prompts are detailed in Appendix A.4.

Stage II: Generation of user instructions. To simulate diverse editing intents, we employ Qwen2.5-VL-72B [60] with a role-playing prompt (Appendix A.4) to translate each $\langle I_{\text{src}}, I_{\text{tgt}}, O \rangle$ triplet into both scene-level and region-level instructions Q . We generate descriptions for two user types—casual users and professional editors with advanced aesthetic sensibilities, ensuring coverage of simple global edits as well as precise, localized adjustments.

Stage III: Generation of reasoning processes. For each sample quadruple $\langle I_{\text{src}}, I_{\text{tgt}}, Q, O \rangle$, we first apply QVQ-max’s [60] advanced visual reasoning to generate initial CoT annotations. To remove redundancy and enforce human-like coherence, we subsequently refine these traces using Qwen2.5-VL-72B [60] through iterative multimodal prompts, producing concise, context-rich reasoning processes $\mathcal{C}$. Full prompt templates are provided in Appendix A.4.

3.3 JarvisArt Framework

3.3.1 CoT Supervised Fine-tuning

Drawing on Deepseek-R1 [13], we initialize JarvisArt via supervised fine-tuning on CoT annotations to bootstrap its subsequent reinforcement learning. This phase 1) enforces a consistent, structured output format, 2) instills foundational reasoning skills spanning user-intent interpretation and aesthetic judgment, and 3) establishes preliminary proficiency in selecting Lightroom tools and configuring their parameters.

3.3.2 Reasoning-oriented Reinforcement Learning

Building on the SFT-initialized model, as shown in Figure 3, we apply group relative policy optimization for retouching (GRPO-R) [48] (Appendix B.1) to further refine JarvisArt’s artistic reasoning and tool-use proficiency. GRPO-R trains the agent with three interpretable, task-specific rewards: a format reward R_f that enforces structured output, a retouching operation accuracy reward R_{roa} that measures the correctness of selected tools and their parameter settings, and a perceptual quality reward R_{pq} that assesses the visual fidelity of the retouched image. The overall objective is thus $R = R_f + R_{\text{roa}} + R_{\text{pq}} \in [0, 3]$.

Format reward. Following prior work [13, 62, 49, 56], we include a format reward $R_f \in [0, 1]$ to enforce structured outputs: reasoning must appear within `<think>` tags and tool invocations within `<answer>` tags, ensuring consistent and reliable parsing.

Retouching operation accuracy reward. Inspired by existing explorations of reward designs [47, 27, 40, 56] in the fields of GUI and web searching. We consider over 200 retouching tools in Lightroom, containing both global adjustments—such as exposure, highlights, and tone curve—and local refinements using six types of masks: 1) linear masks for directional gradients, 2) radial masks for circular or elliptical regions, 3) object masks for isolating subjects (*e.g.*, people or objects), 4) color masks for hue-specific adjustments, 5) luminance masks for brightness-based selections, and 6)

²In Lightroom, a preset refers to a pre-saved retouching operation configuration (ROC) file, typically created by photographers or artists. It encapsulates stylistic parameters tailored for specific visual aesthetics.

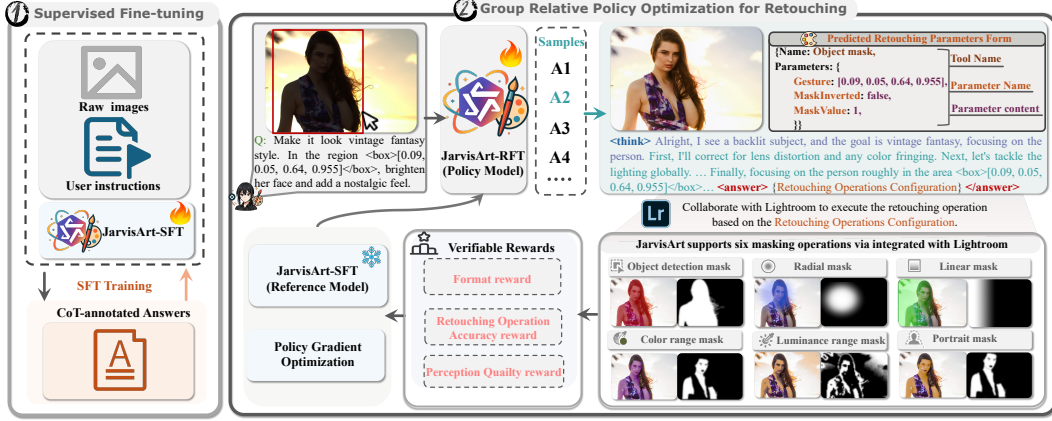


Figure 3: Overview of the two-stage post-training framework. Initially, JarvisArt undergoes supervised fine-tuning (SFT) on CoT-annotated data to develop foundational artistic reasoning and tool-use skills. Following this, we apply the Group Relative Policy Optimization for Retouching (GRPO-R) algorithm to further enhance the JarvisArt’s reasoning, tool proficiency, and generalization.

portrait masks for fine-tuning facial features such as skin and eyes. Further details are provided in Appendix E. To assess the accuracy of predicted tools and their parameters, $T^{pre} = \{T_1^{pre}, \dots, T_M^{pre}\}$, against the ground truth $T^{tgt} = \{T_1^{tgt}, \dots, T_N^{tgt}\}$, we define the ROA reward based on three evaluation criteria:

❶ *Tool name matching:*

$$r_{\text{name}} = \frac{|N_{T^{pre}} \cap N_{T^{tgt}}|}{|N_{T^{pre}} \cup N_{T^{tgt}}|} \in [0, 1], \quad (1)$$

where $N_{T^{pre}}$ and $N_{T^{tgt}}$ are the sets of tool names in the predicted and target sequences, respectively.

❷ *Parameter name matching:*

$$r_{\text{param}} = \sum_{T_j^{tgt} \in T^{tgt}} \sum_{T_i^{pre} \in T^{pre}} \frac{|\text{keys}(T_i^{pre}) \cap \text{keys}(T_j^{tgt})|}{|\text{keys}(T_i^{pre}) \cup \text{keys}(T_j^{tgt})|} \in [0, |T^{tgt}|], \quad (2)$$

where $\text{keys}(\cdot)$ denotes the set of parameter names associated with a predicted or ground-truth tool. It is noted that an overlap in parameter names occurs only when the predicted and ground-truth tool names match.

❸ *Parameter value matching:*

$$r_{\text{value}} = \sum_{T_j^{tgt} \in T^{tgt}} \sum_{T_i^{pre} \in T^{pre}} \sum_{k \in \text{keys}(T_j^{tgt})} S_k(T_i^{pre}[k], T_j^{tgt}[k]) \in [0, \sum_{T_j^{tgt} \in T^{tgt}} |\text{keys}(T_j^{tgt})|], \quad (3)$$

where $S_k(\cdot) \in [0, 1]$ quantifies the correspondence between predicted and ground-truth parameter values, with a value of 1 indicating an exact match. Specifically, if the key k is absent in T_i^{pre} , then $T_i^{pre}[k]$ is undefined and $S_k = 0$. The computation of S_k depends on the parameter type: scalar differences for standard numerical values, intersection-over-union (IoU) for object masks, endpoint distance for linear masks, geometric similarity for radial masks, color distance between sampled points for color masks, luminance range differences for luminance masks, and category-specific criteria for portrait masks. Refer to Appendix B.2 for further details. Finally, the retouching operation accuracy reward is computed by measuring the matching degree between T^{pre} and T^{tgt} :

$$R_{\text{roa}} = \frac{1}{3} \left(r_{\text{name}} + \frac{r_{\text{param}}}{|T^{tgt}|} + \frac{r_{\text{value}}}{\sum_{T_j^{tgt} \in T^{tgt}} |\text{keys}(T_j^{tgt})|} \right) \in [0, 1]. \quad (4)$$

Perception quality reward. While parameter-based rewards offer critical guidance, they may not fully capture the perceptual quality of the final image, as different parameter settings can produce visually similar results. To address this limitation, we introduce the PQ reward, which evaluates two

key aspects: 1) global tone consistency via color distribution matching, and 2) pixel-wise fidelity. The reward is defined as:

$$R_{pq} = \gamma \cdot \text{CD}(I_{\text{edit}}, I_{\text{tgt}}) + (1 - \gamma) \cdot \text{L}(I_{\text{edit}}, I_{\text{tgt}}) \in [0, 1], \quad (5)$$

where I_{edit} is the retouched image and I_{tgt} is the target image. $\text{CD}(\cdot)$ measures color distribution similarity in CIELAB space [69] and $\text{L}(\cdot)$ denotes the pixel-wise distance. Both metrics are normalized to the range $[0, 1]$, with higher values indicating better similarity. The weighting factor is empirically set to $\gamma = 0.4$ to balance both terms.

3.3.3 Agent-to-Lightroom Protocol

Figure 4 presents the Agent-to-Lightroom (A2L) protocol, a standardized client-server interface that integrates JarvisArt with Lightroom. The workflow comprises five stages: 1) handshake, 2) file verification, 3) sandboxed execution, 4) async processing, and 5) result return. A2L features dual-transport communication, a structured message format, and resource management. Messages use bar-delimited commands for processing, status, and error handling, enhancing clarity and efficiency. It manages source images and retouching operation configuration (ROC) files, supporting ROC-to-Lua translation, and integrity checks. The source image can be directly retouched by Lua file in Lightroom. The Lua file can be directly applied in Lightroom to retouch the source image. Additional details are provided in the supplementary materials.

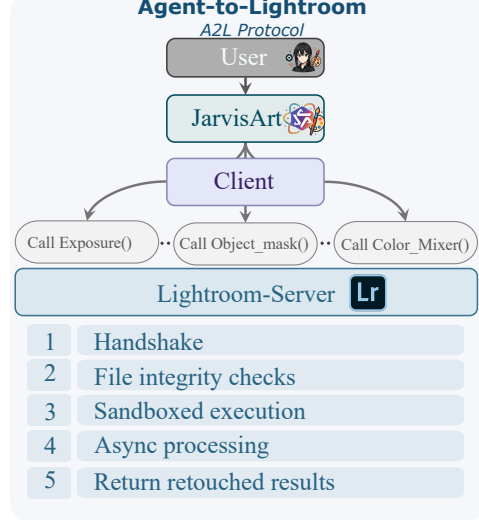


Figure 4: Agent-to-Lightroom protocol.

4 Experiment

4.1 Experimental Setup

Implementation details. We adopt Qwen2.5-VL-7B-Instruct [1] as the base model for JarvisArt. The CoT supervised fine-tuning phase is performed on 50K CoT-annotated instances from MMart, with a batch size of 2, a learning rate of $1e-5$, and training for 2 epochs using the Llama-Factory framework [74] on 8 A100 (80G) GPUs. The reinforcement learning phase, employing the GRPO-R algorithm, is conducted on 5K standard instruction samples from MMart, using the veRL framework [50]. For each training step, we sample a batch of 2, a learning rate of $1e-6$, and generate 4 responses per query, training for 2 epochs on 16 A100 (80G) GPUs.

MMart-Bench. To provide a comprehensive evaluation of JarvisArt’s performance, we introduce the MMart-Bench, which is sampled from the MMart dataset. It includes four main scenarios: portrait, landscape, street scenes, and still life, with 50 instances per category, totaling 200 instances. Each primary category contains multiple subcategories (Appendix A.1). For region-level evaluation, we utilize a portrait subset comprising 50 human-centered images with mask annotations.

Evaluation metrics. Following previous works [68, 30], six assessment metrics are used for evaluation: L1, L2, SC, PQ, and O. L1 and L2 measure the average pixel-level absolute difference between the retouched image and reference image. SC evaluates the alignment between the instruction text and the image (0–10 scale). PQ evaluates contextual coherence and artifact presence (0–10 scale). The overall score O is calculated as $O = \sqrt{SC \times PQ}$. For region-specific evaluation, we apply these six metrics to a specified mask region. Further details are provided in Appendix C.1.

Baselines. For a fair comparison, we evaluate JarvisArt against leading open-source photo retouching methods, including 3DLUT [66] and RSFNet [45], as well as instruction-driven editing models such as InstructPix2Pix [2], MagicBrush [68], OmniGen [57], VARGPT-v1.1 [76] and Step1X-Edit [36]. Proprietary solutions such as GPT-4o³ [24] and Gemini-2-Flash [52]⁴ are also included

³The results are obtained based on ChatGPT APP in May 2025.

⁴The results are obtained based on Gemini API in May 2025.

Table 1: Quantitative evaluation on MArt-Bench. We highlight the best and second-best instruction-based results. SC, PQ, and O refer to the metrics evaluated by Gemini-2-Flash. The RC means the metric calculated on specific mask region.

Method	Instruction	Scene-level					Region-level				
		$L1_{\times 10^2} \downarrow$	$L2_{\times 10^3} \downarrow$	SC $\uparrow$	PQ $\uparrow$	O $\uparrow$	$L1_{\times 10^2}^{RC} \downarrow$	$L2_{\times 10^3}^{RC} \downarrow$	SC $^{RC} \uparrow$	PQ $^{RC} \uparrow$	O $^{RC} \uparrow$
RSFNet [45]	✗	11.61	26.38	-	-	-	8.80	13.69	-	-	-
3DLUT [66]	✗	11.50	25.99	-	-	-	8.33	12.39	-	-	-
InstructPix2Pix [2]	✓	15.67	47.51	6.54	7.79	7.10	12.62	33.39	4.70	5.36	4.91
MagicBrush [68]	✓	18.39	65.25	3.93	4.09	3.85	12.37	32.81	3.04	3.41	3.13
OmniGen [57]	✓	28.49	133.45	4.25	4.42	4.13	25.16	109.10	6.17	7.56	6.72
VARGPT-v1.1 [76]	✓	27.05	126.47	1.83	1.38	1.48	23.71	107.32	1.38	1.15	1.08
Step1X-Edit [36]	✓	24.28	105.91	7.52	8.67	8.01	15.43	45.85	8.32	9.04	8.66
Gemini-2-Flash [52]	✓	23.07	90.99	7.62	8.78	8.08	16.52	52.88	8.04	9.25	8.61
GPT-4o [24]	✓	22.84	92.23	8.73	9.66	9.18	15.71	47.87	8.59	9.48	9.03
JarvisArt	✓	12.44	30.56	7.53	9.82	8.52	7.63	12.14	8.08	9.39	8.69

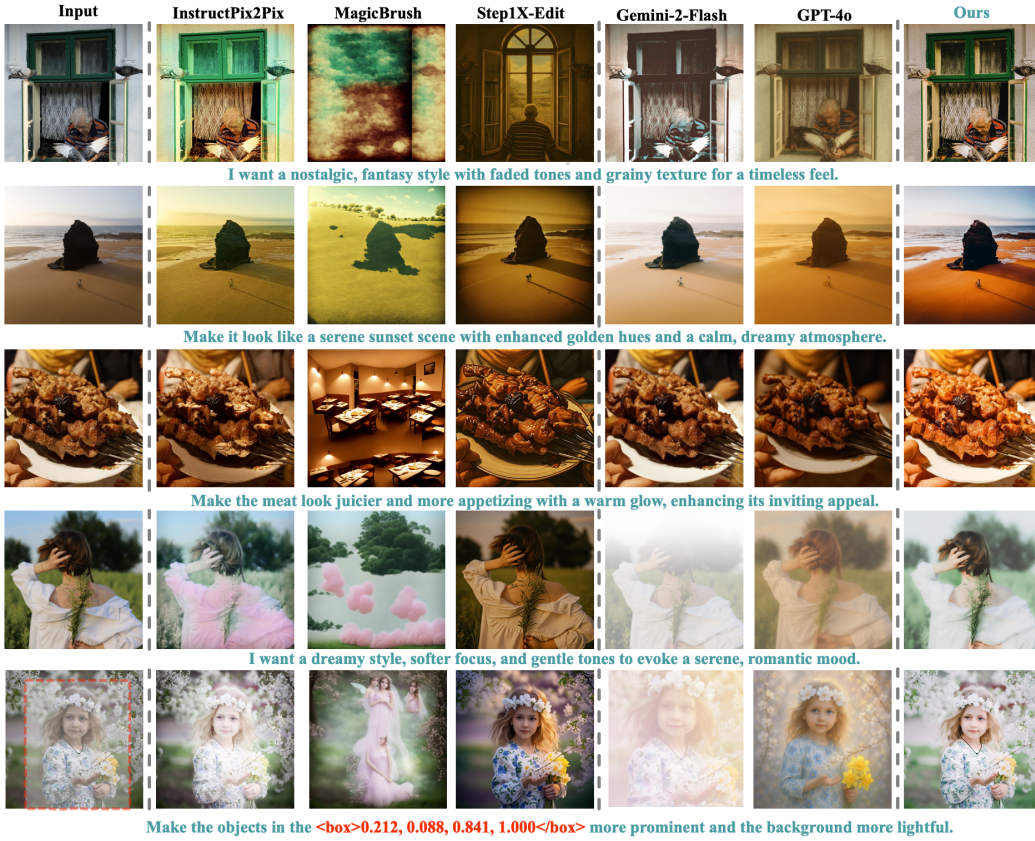


Figure 5: Visual comparison of different methods on MArt-Bench.

for comparison. Notably, all test images are cropped to a 512×512 resolution, as some baselines are incapable of processing high-resolution or arbitrarily sized inputs.

4.2 Experimental Results

4.2.1 Evaluation on MArt-Bench

As shown in Table 1, JarvisArt outperforms most open-source instruction-based baselines, achieving state-of-the-art performance across all 10 evaluation metrics. Compared to closed-source models such as GPT-4o [25] and Gemini-2-Flash [52], JarvisArt achieves superior content preservation—for

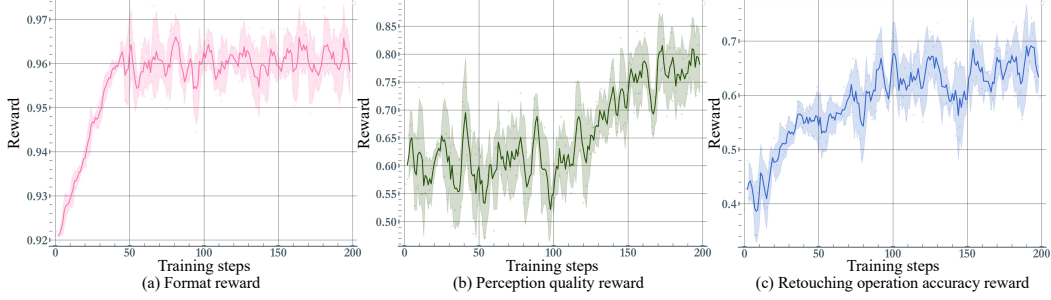


Figure 7: Visualization of the reward trends across training steps of for JarvisArt.

instance, an $L1 \times 10^2$ score of 12.44, which is 45.6% lower (and thus better) than GPT-4o’s score of 22.84. JarvisArt also demonstrates competitive instruction-following capability ($O = 8.52$), closely matching GPT-4o ($O = 9.18$) and outperforming Gemini-2 Flash ($O = 8.08$). Notably, in the local editing setting—where content fidelity is especially critical—the advantage of our method over GPT-4o and Gemini-2-Flash is significantly amplified. As illustrated in Figure 5, especially in portrait scenarios, competing methods often exhibit noticeable uncanny valley effects, producing significant visual artifacts that diverge from users’ creative intent. In contrast, JarvisArt mitigates these issues through its Lightroom-integrated workflow, enabling high-quality, non-destructive editing. More results in Appendix D.

4.2.2 User Preference Study

Evaluating instruction-driven photo retouching remains inherently subjective, as even expert evaluators often disagree on the "optimal" outcome. To quantify preferences, we conducted a user preference study on the MMArt-Bench, recruiting 80 participants to evaluate four advanced algorithms: Step1X-Edit [36], Gemini-2-Flash [52], GPT-4o [24], and JarvisArt. Evaluations focus on two criteria: (1) image consistency (preservation of source image content) and (2) aesthetic quality (visual appeal of retouched results). A five-point ordinal scale (worst = 2, poor = 4, fair = 6, good = 8, and excellent = 10) for quantitative metrics. Results in Figure 6 show JarvisArt achieves best subjective quality, producing edits favored by users.

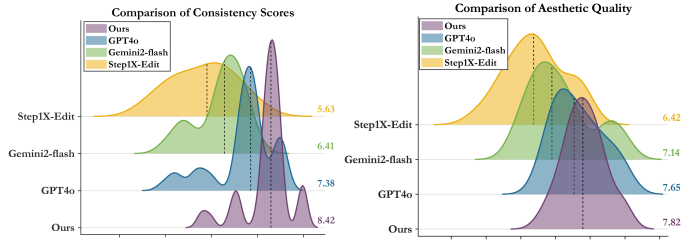


Figure 6: User preference study.

Results in Figure 6 show JarvisArt achieves best subjective quality, producing edits favored by users.

4.2.3 Visualization of Reward Trends for GRPO-R

Figure 7 shows additional visualizations of GRPO-R training. The format reward converges quickly early on. While the PQ reward initially fluctuates and grows gradually, the ROA reward rises more rapidly—likely because the model inherits "parameter preferences" from the SFT phase. As a result, it first focuses on the more easily optimized ROA, then gradually shifts attention to the PQ reward, which requires longer exploration due to the broader search space, where different edit operations may yield similar visual outcomes. Moreover, unlike Deepseek-R1 [13], JarvisArt does not display a clear "aha moment". This absence may stem from the lack of intermediate visual feedback during the artistic reasoning process. For example, when the model makes a hypothetical retouching adjustment like highlight+5, it cannot obtain the corresponding visual result, preventing the model from validating this step’s correctness within the decision-making chain. Unlike mathematical problem-solving, where each step can be validated immediately, our artistic reasoning involves numerous retouching parameters. If we perform step-wise validation for each parameter, it would require high concurrency in calling Lightroom. This is impractical due to the high computational cost and the slow training speed. Investigating step-wise visual rewards within proxy validation environments may offer a promising approach to eliciting the "aha moment". We intend to explore in future work.

Table 2: Ablation studies on different training strategies and reward design.

Configurations	L1 $\times 10^2$ ↓	L2 $\times 10^3$ ↓	SC ↑	PQ ↑	O ↑
Training strategy					
only SFT	14.42	44.38	7.32	8.67	7.94
only RL	17.55	58.19	6.88	8.13	7.38
SFT + RL (Ours)	12.44	30.56	7.53	9.82	8.52
Reward design					
Format + ROA	14.09	40.36	7.45	8.77	8.04
Format + PQ	13.78	35.41	7.48	8.92	8.15
Format + ROA + PQ (Ours)	12.44	30.56	7.53	9.82	8.52

5 Ablation Study

Training strategy. We assess the impact of different post-training strategies by comparing model performance under three settings: 1) SFT on 50K CoT-enhanced samples, 2) GRPO-R training on 5K standard samples from scratch; and 3) GRPO-R fine-tuning based on SFT-initial model. Rows 2–4 in Table 2 show that SFT yields better results than GRPO-R trained from scratch. This is likely because, without SFT to instill the basic reasoning and tool-use abilities, the GRPO-R training process must explore a significantly larger search space, thereby hindering optimization. Our combined SFT+GRPO-R strategy achieves the best results, suggesting that GRPO-R can effectively enhance the SFT-initialized model’s reasoning, tool proficiency, and generalization by expanding its exploration capacity.

Reward design. As shown in Rows 6–8 of Table 2, individual reward combinations (Format+ROA or Format+PQ) result in suboptimal performance, with Format+PQ performing slightly better—possibly because PQ aligns more closely with the ultimate objective of enhancing visual quality and offers a broader optimization space to escape local optima. The full combination (Format+PQ+ROA) achieves the highest performance. This result aligns with our intuition that parameter-oriented (ROA) and perception-driven (PQ) rewards are complementary: ROA ensures parameter accuracy, while PQ maintains visual fidelity. The multi-dimensional reward system provides a balanced optimization signal, guiding the model to predict accurate edit operations while preserving high visual quality.

6 Conclusion

This report introduces JarvisArt, an interactive and interpretable MLLM-guided agent that integrates with 200+ Lightroom editing tools, enabling non-destructive editing on images of any-resolution. To develop this artist agent, we propose a new data generation pipeline that curates the MMart-55K dataset, comprising 5K standard and 50K CoT-enhanced samples. Based on this dataset, we train JarvisArt using a two-stage post-training regimen: 1) CoT SFT to instill basic reasoning and tool-use abilities, and 2) GRPO-R to improve the agent’s reasoning, tool proficiency, and generalization through customized tool-use rewards: retouching operation accuracy reward for assessing the predicted editing operations, and the perceptual quality reward to evaluate the visual fidelity of the edited outputs. Furthermore, to enable seamless, automated Lightroom editing, we introduce the Agent-to-Lightroom protocol. Evaluation results from our MMart-Bench demonstrate that our proposed algorithm significantly outperforms existing advanced image editing algorithms.

7 Acknowledgments

This work was supported in part by the Xiamen science and technology major project under Grant 3502Z20241029; in part by the Dreams Foundation of Jianghuai Advance Technology Center project under Grant 2023-ZM01D002; in part by the National Natural Science Foundation of China under Grant 82172073, Grant 82272071, and Grant 62271430; and in part by the Open Fund of the National Key Laboratory of Infrared Detection Technologies.

References

- [1] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.
- [2] T. Brooks, A. Holynski, and A. 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] V. Bychkovsky, S. Paris, E. Chan, and F. Durand. Learning photographic global tonal adjustment with a database of input / output image pairs. In *The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
- [4] Z. Cai, M. Cao, H. Chen, K. Chen, K. Chen, X. Chen, X. Chen, Z. Chen, Z. Chen, P. Chu, et al. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*, 2024.
- [5] L. Chen, L. Li, H. Zhao, Y. Song, and Vinci. R1-v: Reinforcing super generalization ability in vision-language models with less than \$3. *GitHub repository: <https://github.com/Deep-Agent/R1-V>*, 2025.
- [6] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 15–26, 2017.
- [7] S. Chen, J. Bai, Z. Zhao, T. Ye, Q. Shi, D. Zhou, W. Chai, X. Lin, J. Wu, C. Tang, et al. An empirical study of gpt-4o image generation capabilities. *arXiv preprint arXiv:2504.05979*, 2025.
- [8] X. Chen, Z. Wu, X. Liu, Z. Pan, W. Liu, Z. Xie, X. Yu, and C. Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.
- [9] C. Deng, D. Zhu, K. Li, C. Gou, F. Li, Z. Wang, S. Zhong, W. Yu, X. Nie, Z. Song, G. Shi, and H. Fan. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.
- [10] K. Feng, Y. Ma, B. Wang, C. Qi, H. Chen, Q. Chen, and Z. Wang. Dit4edit: Diffusion transformer for image editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 2969–2977, 2025.
- [11] K. Feng, Y. Ma, X. Zhang, B. Liu, Y. Yuluo, Y. Zhang, R. Liu, H. Liu, Z. Qin, S. Mo, et al. Follow-your-instruction: A comprehensive mllm agent for world data synthesis. *arXiv preprint arXiv:2508.05580*, 2025.
- [12] T.-J. Fu, W. Hu, X. Du, W. Y. Wang, Y. Yang, and Z. Gan. Guiding instruction-based image editing via multimodal large language models. In *International Conference on Learning Representations (ICLR)*, 2024.
- [13] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- [14] Z. Guo, L. Xia, Y. Yu, T. Ao, and C. Huang. Lightrag: Simple and fast retrieval-augmented generation. 2024.
- [15] N. Hansen. The cma evolution strategy: a comparing review. *Towards a new evolutionary computation: Advances in the estimation of distribution algorithms*, pages 75–102, 2006.
- [16] J.-Y. He, Z.-Q. Cheng, C. Li, J. Sun, Q. He, W. Xiang, H. Chen, J.-P. Lan, X. Lin, K. Zhu, et al. Metadesigner: Advancing artistic typography through ai-driven, user-centric, and multilingual wordart synthesis. *arXiv preprint arXiv:2406.19859*, 2024.
- [17] J.-Y. He, Z.-Q. Cheng, C. Li, J. Sun, W. Xiang, X. Lin, X. Kang, Z. Jin, Y. Hu, B. Luo, et al. Wordart designer: User-driven artistic typography synthesis using large language models. *arXiv preprint arXiv:2310.18332*, 2023.
- [18] S. Hong, X. Zheng, J. Chen, Y. Cheng, J. Wang, C. Zhang, Z. Wang, S. K. S. Yau, Z. Lin, L. Zhou, et al. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 3(4):6, 2023.
- [19] Y. Hu, H. He, C. Xu, B. Wang, and S. Lin. Exposure: A white-box photo post-processing framework. *ACM Transactions on Graphics (TOG)*, 37(2):1–17, 2018.

- [20] R. Huang, M. Li, D. Yang, J. Shi, X. Chang, Z. Ye, Y. Wu, Z. Hong, J. Huang, J. Liu, et al. Audiogpt: Understanding and generating speech, music, sound, and talking head. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 23802–23804, 2024.
- [21] W. Huang, B. Jia, Z. Zhai, S. Cao, Z. Ye, F. Zhao, Y. Hu, and S. Lin. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*, 2025.
- [22] B. Hui, J. Yang, Z. Cui, J. Yang, D. Liu, L. Zhang, T. Liu, J. Zhang, B. Yu, K. Lu, et al. Qwen2.5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.
- [23] M. Hui, S. Yang, B. Zhao, Y. Shi, H. Wang, P. Wang, Y. Zhou, and C. Xie. Hq-edit: A high-quality dataset for instruction-based image editing. *arXiv preprint arXiv:2404.09990*, 2024.
- [24] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.
- [25] A. Jaech, A. Kalai, A. Lerer, A. Richardson, A. El-Kishky, A. Low, A. Helyar, A. Madry, A. Beutel, A. Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.
- [26] F. Jiao, G. Guo, X. Zhang, N. F. Chen, S. Joty, and F. Wei. Preference optimization for reasoning with pseudo feedback. *arXiv preprint arXiv:2411.16345*, 2024.
- [27] B. Jin, H. Zeng, Z. Yue, D. Wang, H. Zamani, and J. Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.
- [28] Z. Ke, C. Sun, L. Zhu, K. Xu, and R. W. Lau. Harmonizer: Learning to perform white-box image and video harmonization. In *European conference on computer vision*, pages 690–706. Springer, 2022.
- [29] S. Kosugi and T. Yamasaki. Unpaired image enhancement featuring reinforcement-learning-controlled image editing software. In *Proceedings of the AAAI conference on artificial intelligence*, pages 11296–11303, 2020.
- [30] M. Ku, D. Jiang, C. Wei, X. Yue, and W. Chen. Viescore: Towards explainable metrics for conditional image synthesis evaluation. *arXiv preprint arXiv:2312.14867*, 2023.
- [31] LangChain. Langchain: Build context-aware reasoning applications. <https://github.com/langchain-ai/langchain>, 2023.
- [32] J. Liang, H. Zeng, M. Cui, X. Xie, and L. Zhang. Ppr10k: A large-scale portrait photo retouching dataset with human-region mask and group-level consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 653–661, 2021.
- [33] Y. Lin, Z. Lin, H. Chen, P. Pan, C. Li, S. Chen, W. Kairun, Y. Jin, W. Li, and X. Ding. Jarvisir: Elevating autonomous driving perception with intelligent image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.
- [34] Y. Lin, Z. Lin, H. Chen, P. Pan, C. Li, S. Chen, K. Wen, Y. Jin, W. Li, and X. Ding. Jarvisir: Elevating autonomous driving perception with intelligent image restoration. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22369–22380, 2025.
- [35] P. Liu, C. Li, Z. Li, Y. Wu, W. Li, Z. Yang, Z. Zhang, Y. Lin, S. Han, and B. Y. Feng. Ir3d-bench: Evaluating vision-language model scene understanding as agentic inverse rendering. *arXiv preprint arXiv:2506.23329*, 2025.
- [36] S. Liu, Y. Han, P. Xing, F. Yin, R. Wang, W. Cheng, J. Liao, Y. Wang, H. Fu, C. Han, et al. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025.
- [37] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. 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.
- [38] Z. Liu, T. Hoang, J. Zhang, M. Zhu, T. Lan, J. Tan, W. Yao, Z. Liu, Y. Feng, R. RN, et al. Apigen: Automated pipeline for generating verifiable and diverse function-calling datasets. *Advances in Neural Information Processing Systems*, 37:54463–54482, 2024.
- [39] Z. Liu, Z. Sun, Y. Zang, X. Dong, Y. Cao, H. Duan, D. Lin, and J. Wang. Visual-rft: Visual reinforcement fine-tuning. *arXiv preprint arXiv:2503.01785*, 2025.
- [40] Z. Lu, Y. Chai, Y. Guo, X. Yin, L. Liu, H. Wang, G. Xiong, and H. Li. Ui-r1: Enhancing action prediction of gui agents by reinforcement learning. *arXiv preprint arXiv:2503.21620*, 2025.

- [41] C. Ma, Y. Jiang, J. Wu, J. Yang, X. Yu, Z. Yuan, B. Peng, and X. Qi. Unitok: A unified tokenizer for visual generation and understanding. *arXiv preprint arXiv:2502.20321*, 2025.
- [42] F. Meng, L. Du, Z. Liu, Z. Zhou, Q. Lu, D. Fu, B. Shi, W. Wang, J. He, K. Zhang, et al. Mm-eureka: Exploring visual aha moment with rule-based large-scale reinforcement learning. *arXiv preprint arXiv:2503.07365*, 2025.
- [43] A. Mosleh, A. Sharma, E. Onzon, F. Mannan, N. Robidoux, and F. Heide. Hardware-in-the-loop end-to-end optimization of camera image processing pipelines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7529–7538, 2020.
- [44] J. Nishimura, T. Gerasimow, R. Sushma, A. Sutic, C.-T. Wu, and G. Michael. Automatic isp image quality tuning using nonlinear optimization. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2471–2475. IEEE, 2018.
- [45] W. Ouyang, Y. Dong, X. Kang, P. Ren, X. Xu, and X. Xie. Rsfnet: A white-box image retouching approach using region-specific color filters. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12160–12169, 2023.
- [46] Z. Pan and H. Liu. Metaspatial: Reinforcing 3d spatial reasoning in vlms for the metaverse. *arXiv preprint arXiv:2503.18470*, 2025.
- [47] C. Qian, E. C. Acikgoz, Q. He, H. Wang, X. Chen, D. Hakkani-Tür, G. Tur, and H. Ji. Toolrl: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*, 2025.
- [48] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- [49] H. Shen, P. Liu, J. Li, C. Fang, Y. Ma, J. Liao, Q. Shen, Z. Zhang, K. Zhao, Q. Zhang, et al. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025.
- [50] G. Sheng, C. Zhang, Z. Ye, X. Wu, W. Zhang, R. Zhang, Y. Peng, H. Lin, and C. Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.
- [51] Significant-Gravitas. Autogpt. <https://github.com/Significant-Gravitas/AutoGPT>, 2023.
- [52] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.
- [53] E. Tseng, F. Yu, Y. Yang, F. Mannan, K. S. Arnaud, D. Nowrouzezahrai, J.-F. Lalonde, and F. Heide. Hyperparameter optimization in black-box image processing using differentiable proxies. *ACM Trans. Graph.*, 38(4):27–1, 2019.
- [54] E. Tseng, Y. Zhang, L. Jebe, X. Zhang, Z. Xia, Y. Fan, F. Heide, and J. Chen. Neural photo-finishing. *ACM Trans. Graph.*, 41(6):238–1, 2022.
- [55] J. Wu, Y. Wang, L. Li, F. Zhang, and T. Xue. Goal conditioned reinforcement learning for photo finishing tuning. *Advances in Neural Information Processing Systems*, 37:46294–46318, 2024.
- [56] X. Xia and R. Luo. Gui-r1: A generalist r1-style vision-language action model for gui agents. *arXiv preprint arXiv:2504.10458*, 2025.
- [57] S. Xiao, Y. Wang, J. Zhou, H. Yuan, X. Xing, R. Yan, S. Wang, T. Huang, and Z. Liu. Omnigen: Unified image generation. *arXiv preprint arXiv:2409.11340*, 2024.
- [58] T. Xie, F. Zhou, Z. Cheng, P. Shi, L. Weng, Y. Liu, T. J. Hua, J. Zhao, Q. Liu, C. Liu, et al. Openagents: An open platform for language agents in the wild. *arXiv preprint arXiv:2310.10634*, 2023.
- [59] Z. Yan, Y. Ma, C. Zou, W. Chen, Q. Chen, and L. Zhang. Eedit: Rethinking the spatial and temporal redundancy for efficient image editing. *arXiv preprint arXiv:2503.10270*, 2025.
- [60] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.
- [61] A. Yang, B. Zhang, B. Hui, B. Gao, B. Yu, C. Li, D. Liu, J. Tu, J. Zhou, J. Lin, et al. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

- [62] Y. Yang, X. He, H. Pan, X. Jiang, Y. Deng, X. Yang, H. Lu, D. Yin, F. Rao, M. Zhu, et al. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization. *arXiv preprint arXiv:2503.10615*, 2025.
- [63] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.
- [64] H. Ying, S. Zhang, L. Li, Z. Zhou, Y. Shao, Z. Fei, Y. Ma, J. Hong, K. Liu, Z. Wang, et al. Internlm-math: Open math large language models toward verifiable reasoning. *arXiv preprint arXiv:2402.06332*, 2024.
- [65] K. Yu, Z. Li, Y. Peng, C. C. Loy, and J. Gu. Reconfigisp: Reconfigurable camera image processing pipeline. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4248–4257, 2021.
- [66] H. Zeng, J. Cai, L. Li, Z. Cao, and L. Zhang. Learning image-adaptive 3d lookup tables for high performance photo enhancement in real-time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4):2058–2073, 2020.
- [67] K. Zhang, G. Li, Y. Dong, J. Xu, J. Zhang, J. Su, Y. Liu, and Z. Jin. Codedpo: Aligning code models with self generated and verified source code. *arXiv preprint arXiv:2410.05605*, 2024.
- [68] K. Zhang, L. Mo, W. Chen, H. Sun, and Y. Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36:31428–31449, 2023.
- [69] X. Zhang, B. A. Wandell, et al. A spatial extension of cielab for digital color image reproduction. In *SID international symposium digest of technical papers*, volume 27, pages 731–734. Citeseer, 1996.
- [70] Y. Zhang, S. Wu, Y. Yang, J. Shu, J. Xiao, C. Kong, and J. Sang. o1-coder: an o1 replication for coding. *arXiv preprint arXiv:2412.00154*, 2024.
- [71] Z. Zhang, X. Bo, C. Ma, R. Li, X. Chen, Q. Dai, J. Zhu, Z. Dong, and J.-R. Wen. A survey on the memory mechanism of large language model based agents. *arXiv preprint arXiv:2404.13501*, 2024.
- [72] H. Zhao, X. S. Ma, L. Chen, S. Si, R. Wu, K. An, P. Yu, M. Zhang, Q. Li, and B. Chang. Ultraedit: Instruction-based fine-grained image editing at scale. *Advances in Neural Information Processing Systems*, 37:3058–3093, 2024.
- [73] Y. Zhao, F. Xue, S. Reed, L. Fan, Y. Zhu, J. Kautz, Z. Yu, P. Krähenbühl, and D.-A. Huang. Qlip: Text-aligned visual tokenization unifies auto-regressive multimodal understanding and generation. *arXiv preprint arXiv:2502.05178*, 2025.
- [74] Y. Zheng, R. Zhang, J. Zhang, Y. Ye, Z. Luo, Z. Feng, and Y. Ma. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand, 2024. Association for Computational Linguistics.
- [75] D. Zhou, J. Huang, J. Bai, J. Wang, H. Chen, G. Chen, X. Hu, and P.-A. Heng. Magictailor: Component-controllable personalization in text-to-image diffusion models. *arXiv preprint arXiv:2410.13370*, 2024.
- [76] X. Zhuang, Y. Xie, Y. Deng, D. Yang, L. Liang, J. Ru, Y. Yin, and Y. Zou. Vargpt-v1. 1: Improve visual autoregressive large unified model via iterative instruction tuning and reinforcement learning. *arXiv preprint arXiv:2504.02949*, 2025.

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 sections offer a comprehensive discussion of the manuscript's context, intuition, and ambitions, as well as its contributions.

Guidelines:

- The answer NA 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 NA 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: The limitations of the work are discussed by authors at the end of the paper.

Guidelines:

- The answer NA 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: [\[Yes\]](#)

Justification: For each theoretical result, the paper provides the full set of assumptions and a complete (and correct) proof.

Guidelines:

- The answer NA 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: The pipeline of the methods and the details of experiments are presented with corresponding reproducible credentials.

Guidelines:

- The answer NA 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: Code will be released upon acceptance of the paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.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://nips.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, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: The pipeline of the methods and the details of experiments are presented with corresponding reproducible credentials.

Guidelines:

- The answer NA 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: The results contain the standard deviation of the results over several random runs.

Guidelines:

- The answer NA 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: The details of experiments are presented with corresponding reproducible credentials.

Guidelines:

- The answer NA 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 conducted in the paper conforms with the NeurIPS Code of Ethics

Guidelines:

- The answer NA 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: Despite JarvisArt's strong artistic reasoning and tool-use capabilities, limitations remain. First, it lacks support for advanced Lightroom AI features like background removal, which we plan to address through paired artist-edited data generation. Second, GRPO-R training on only 5K samples may introduce bias; expanding this dataset could improve generalization.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: 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, by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

Guidelines:

- The answer NA 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: The original owners of assets, including data and models, used in the paper, are properly credited and are the license and terms of use explicitly mentioned and properly respected.

Guidelines:

- The answer NA 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: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA 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: [NA]

Justification: The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA 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: [NA]

Justification: The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: As the starting point for model training.

Guidelines:

- The answer NA 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 (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

Appendices

Our Appendices includes the following sections:

- Sec.A Details of the MMArt Dataset.
 - Statistics of the MMArt Dataset.
 - Comparison of Existing Datasets.
 - Data Samples of MMArt.
 - Prompt Templates.
- Sec.B Additional Method Details.
 - Group Relative Policy Optimization.
 - Details of Reward Calculation.
- Sec.C Additional Experimental Details.
 - Calculation of Local Metrics.
 - Prompts for MLLM-based Metrics.
- Sec.D Additional Experimental Results.
 - Additional Quantitative Evaluation by Qwen-2.5-VL-72B.
 - Examples of Intricate Retouching Tasks with JarvisArt.
 - More Visual Comparisons.
 - Comparison on MIT-FiveK.
- Sec.E Details of Retouching Tools in Lightroom.

A Details of the MMArt dataset.

A.1 Statistics of the MMArt dataset

Figure 8(a) illustrates the composition and distribution of scenarios in our MMArt dataset. The dataset is structured into four major scene categories that reflect common real-world photo retouching contexts: portrait (40.8%, including shooting purposes, shooting time/lights, subjects, and indoor/outdoor scenes), landscape (33.3%, comprising nature, city, aerial photography, travel, underwater, night scene and architecture), street scenes (5.71%, including sports, life, event and documentary), and still life (20.2%, encompassing food, close-up scenes, black/white photography, art and animals). Each major category contains a diverse set of subcategories, ensuring comprehensive coverage and representativeness. Furthermore, Figure 8(b) displays a word cloud of user instructions, highlighting the linguistic diversity of the instructions.

A.2 Comparison of Existing Datasets

Table 3 presents a comparison between our MMArt dataset and existing image editing datasets. MMArt is designed with the following key properties to facilitate advanced research in image retouching:

- **Real Images:** All samples in MMArt are real photographs, ensuring the dataset’s authenticity and practical value for real-world applications.
- **Diverse User Instructions:** Each image is paired with detailed user instructions, capturing a wide variety of editing intentions and reflecting the diversity of natural language expressions.
- **Flexible Resolution:** MMArt supports images of any resolution, including high-resolution samples, making it suitable for both research and practical deployment scenarios.
- **Chain-of-Thought (CoT) Annotations:** The dataset provides CoT reasoning annotations, which help to reveal the underlying logic and step-by-step process of user intent understanding and image editing.
- **Lightroom Retouching Configuration:** For every sample, MMArt includes comprehensive Lightroom parameter configurations, allowing for non-destructive, reproducible, and transparent image editing.

These properties make MMArt a high-quality, flexible, and richly annotated resource for the development and evaluation of advanced image retouching techniques.

A.3 Data Samples of MMArt

The diversity of collected photos is shown in Figure 9. Moreover, Figure 10 demonstrates MMArt samples with Chain-of-Thought (CoT) reasoning, while Figure 11 shows standard examples without CoT annotations.

Table 3: Comparison of MMArt and existing retouching datasets in terms of data properties.

Property	InstructP2P [2]	MagicBrush [68]	UltraEdit [72]	MGIE [12]	HQEdit [23]	FiveK [3]	MMArt
Real Image?	✗	✓	✓	✓	✗	✓	✓
User Instructions?	✓	✓	✓	✓	✓	✗	✓
Any Resolution?	✗	✗	✓	✗	✗	✓	✓
High Resolution?	✗	✗	✗	✗	✓	✓	✓
CoT Annotations?	✗	✗	✗	✗	✗	✗	✓
Lightroom Configuration?	✗	✗	✗	✗	✗	✓	✓

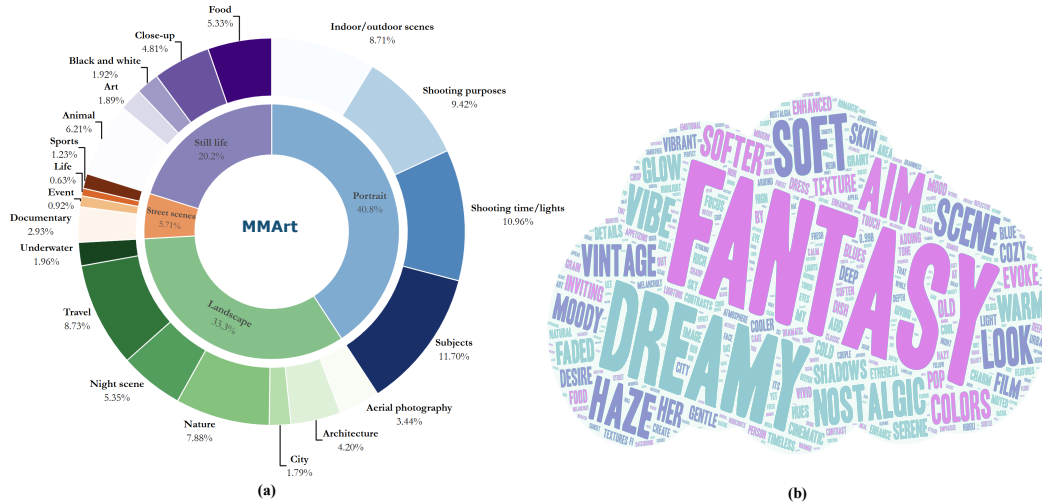


Figure 8: Statistics of the MMArt dataset. (a) The dataset is divided into four primary scenarios: portrait, landscape, street scenes, and still life, each containing a variety of subcategories. (b) A word cloud illustrates the rich linguistic diversity found in user instructions.

A.4 Prompt Templates

The prompt templates utilized throughout the various stages of MMart are summarized here—Aesthetic Preset Recommendation (Figure 21), User Instruction Simulation (Figures 22 and 23), and Chain-of-Thought Data Construction (Figure 24 and 25).

B Additional Method Details

B.1 Group Relative Policy Optimization

In GRPO, given a task question, the model generates a set of N potential responses $\{O_1, O_2, \dots, O_N\}$. Each response is evaluated by taking the corresponding actions and computing its reward $\{R_1, R_2, \dots, R_N\}$. Unlike PPO, which relies on a single reward signal and a critic to estimate the value function, GRPO normalizes these rewards to calculate the relative advantage of each response. The relative quality A_i of the i -th response is computed as

$$A_i = \frac{r_i - \text{Mean}(\{r_1, r_2, \dots, r_N\})}{\text{Std}(\{r_1, r_2, \dots, r_N\})},$$

where Mean and Std represent the mean and standard deviation of the rewards, respectively. This normalization step ensures that responses are compared within the context of the group, allowing GRPO to better capture nuanced differences between candidates. Policy updates are further constrained by minimizing the KL divergence between the updated and reference models, ensuring stable RL learning. Refer to [13, 48] for more details.

B.2 Details of Reward Calculation

The parameter value matching function $S_k(\cdot)$ for each parameter k is determined based on its specific type. Let V_k^{pre} and V_k^{tgt} denote the predicted and ground truth values for the k -th parameter, respectively. For notational simplicity, we omit the subscript k in the following formulas. The calculation proceeds as follows:

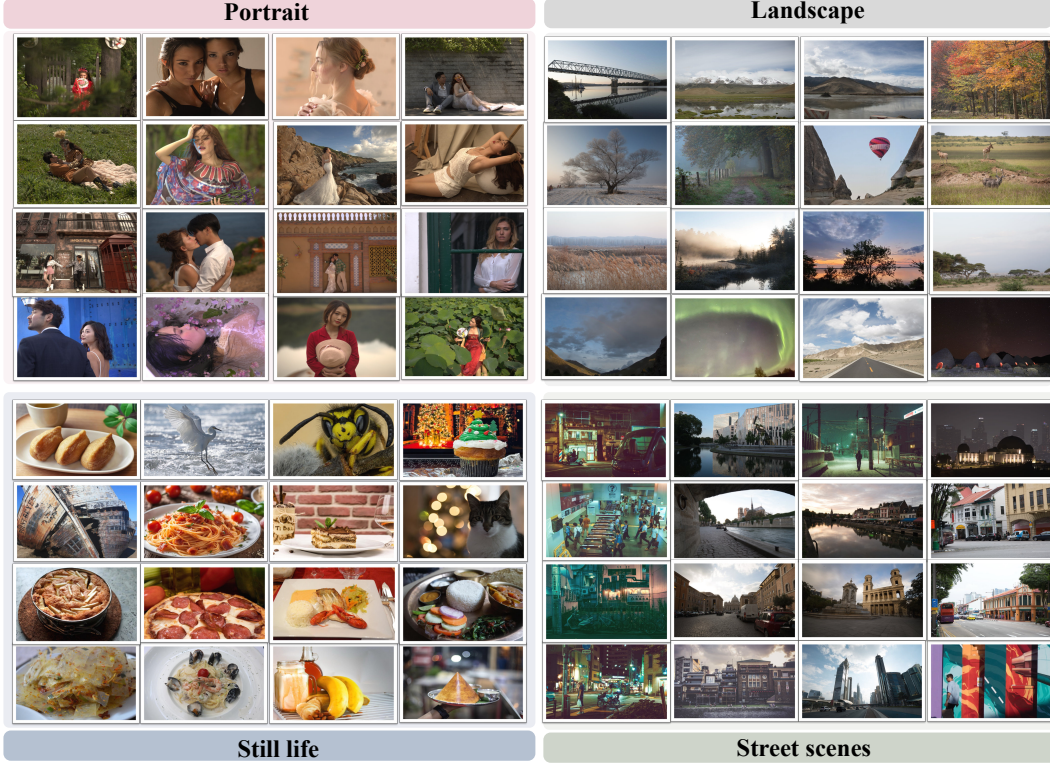


Figure 9: Visual examples to demonstrate the diversity of the proposed dataset.

- **Scalar Parameters.** For scalar parameters such as exposure or contrast, the matching function S is defined as:

$$S = \max\left(0, 1 - \frac{|V^{pre} - V^{tgt}|}{V_{max} - V_{min}}\right) \in [0, 1],$$

where $|\cdot|$ represents the absolute error between the predicted and ground truth values.

- **Linear Gradient Masks.** We assess the similarity between predicted and target linear gradient masks by measuring the distances between their start points $p_s = (x_s, y_s)$ and end points $p_e = (x_e, y_e)$, with coordinates normalized to $[0, 1]$ for resolution invariance. The similarity score is computed as:

$$S = \max\left(0, 1 - \|p_s^{pre} - p_s^{tgt}\| - \|p_e^{pre} - p_e^{tgt}\|\right) \in [0, 1],$$

where $\|\cdot\|$ denotes Euclidean distance.

- **Radial Gradient Masks.** We measure similarity between predicted and target radial gradient masks using three geometric parameters: center position $c = (x, y)$, scale factors (W, H) , and rotation angle θ . Center point similarity is given by:

$$S_{center} = \max(0, 1 - 2 \cdot \|c^{pre} - c^{tgt}\|) \in [0, 1],$$

where c^{pre} and c^{tgt} are normalized to $[0, 1]$. Further, scaling similarity compares width/height ratios:

$$S_{scale} = \max(0, 1 - |W^{pre}/W^{tgt} - 1| - |H^{pre}/H^{tgt} - 1|) \in [0, 1],$$

The angle numerical value similarity is defined by:

$$S_{angle} = \max\left(0, 1 - \frac{|\theta^{pre} - \theta^{tgt}|}{\theta_{max} - \theta_{min}}\right) \in [0, 1],$$

The final similarity score combines these components as follows:

$$S = 0.4 \cdot S_{center} + 0.4 \cdot S_{scale} + 0.2 \cdot S_{angle} \in [0, 1].$$

- **Object Masks.** For object masks, the similarity score S is defined as the Intersection-over-Union (IoU) between the predicted B^{pre} and ground truth B^{tgt} bounding boxes. Each box is parameterized as $[x_1, y_1, x_2, y_2]$. The similarity score is computed as:

$$S = \text{IoU}(B^{pre}, B^{tgt}) \in [0, 1],$$

Table 4: Hyper-parameter settings for SFT and GRPO-R

Hyper-parameter	SFT	GRPO-R
Batch size	2	2
Learning rate	1×10^{-5}	1×10^{-6}
Weight decay	0	0
Optimizer	AdamW	AdamW
Warmup ratio	0.1	0.1
LR scheduler	cosine	cosine
Training samples	50K	5K
Training epochs	2	2
Precision	bfloat16	bfloat16
KL coefficient	—	0.04
Reward coefficients	—	$R_f : 1, R_{roa} : 1, R_{pq} : 1$ ($\gamma = 0.4$)
Number of generations	—	4
GPU resources	$8 \times \text{A100}$ (≈ 384 GPU hours)	$16 \times \text{A100}$ (≈ 2076 GPU hours)

where higher values indicate better alignment, with $S = 1$ denoting perfect overlap and $S = 0$ indicating no intersection.

- **Portrait Masks.** In portrait masks, the model predicts different special category IDs to denote distinct regions, such as ID=0 for face, ID=1 for hair, ID=2 for eyes, ID=3 for skin, etc. The matching score S is defined as follows:

$$S = \begin{cases} 1, & \text{if the predicted and target category IDs coincide,} \\ 0, & \text{otherwise.} \end{cases}$$

- **Color Range Masks.** To evaluate color range mask similarity, we sample N representative points from both predicted and target color distributions and compute the mean CIEDE2000 color difference ΔE_{100} in LAB color space. The similarity score is given by:

$$S = \max \left(0, 1 - \frac{1}{N} \sum_{n=1}^N \Delta E_{100}(c_n^{pre}, c_n^{tgt}) \right) \in [0, 1],$$

where c_n^{pre} and c_n^{tgt} denote the n -th sampled colors from the predicted and ground-truth distributions, respectively.

- **Luminance Range Masks.** To evaluate luminance range mask similarity, we compare the predicted and target luminance extremes by computing their absolute differences. The similarity score is defined as:

$$S = \max \left(0, 1 - \frac{|l_{\min}^{pre} - l_{\min}^{tgt}| + |l_{\max}^{pre} - l_{\max}^{tgt}|}{2(l_{\max}^{tgt} - l_{\min}^{tgt})} \right) \in [0, 1],$$

where the denominator normalizes by the target luminance range to ensure scale invariance.

B.3 Hyper-parameter Setting

To ensure reproducibility, we provide the complete hyperparameter settings for both the SFT and GRPO-R phases in Table 4.

C Additional Experimental Details

C.1 Calculation of Local Metrics

To evaluate the model’s effectiveness in localized regions, we compute six metrics— $L1^{RC}$, $L2^{RC}$, SC^{RC} , PQ^{RC} , and O^{RC} —using human-centric masks from the portrait subset of MMart-Bench. For $L1^{RC}$ and $L2^{RC}$, inspired by PPR10K [32], given an image I with resolution $H \times W$, we define a weighting matrix $W_I = [w_{i,j}] \in \mathbb{R}^{H \times W}$, where $w_{i,j} = 1$ for human regions and $w_{i,j} = \alpha$ ($\alpha \leq 1$) for background regions, with α empirically set to 0.5. For instance, the human-centric L1 difference metric is expressed as:

$$L1^{RC} = |W_I \circ I^{pred} - W_I \circ I^{tgt}|,$$

where I^{pred} and I^{tgt} are the predicted and target images, respectively, and $\circ$ denotes element-wise multiplication. The $L2^{RC}$ metric is defined in a similar manner. For SC^{RC} , PQ^{RC} , and O^{RC} , with α empirically set to 0, we focus solely on the mask region of the edited image and prompt the MLLM to emphasize local adjustments.

Table 5: Quantitative evaluation on MMArt-Bench. We highlight the **best** and **second-best** instruction-based results. SC, PQ, and O refer to the metrics evaluated by Qwen2.5-VL-72B [1]. The RC means the metric calculated on specific mask region.

Method	Instruction	Scene-level					Region-level				
		$L1_{\times 10^2} \downarrow$	$L2_{\times 10^3} \downarrow$	SC $\uparrow$	PQ $\uparrow$	O $\uparrow$	$L1_{\times 10^2}^{RC} \downarrow$	$L2_{\times 10^3}^{RC} \downarrow$	$SC^{RC} \uparrow$	$PQ^{RC} \uparrow$	$O^{RC} \uparrow$
RSFNet [45]	✗	11.62	26.38	-	-	-	8.80	13.70	-	-	-
3DLUT [66]	✗	11.51	26.00	-	-	-	8.34	12.26	-	-	-
InstructPix2Pix [2]	✓	15.62	47.26	6.17	5.81	5.47	12.48	32.69	4.67	3.85	3.64
MagicBrush [68]	✓	18.31	64.76	3.93	2.25	2.44	12.44	32.93	2.80	2.15	2.01
OmniGen [57]	✓	28.40	132.82	4.14	2.16	2.70	24.85	106.81	3.80	3.85	3.67
VARGPT-v1.1 [76]	✓	27.04	126.26	1.27	0.17	0.29	23.59	105.86	0.09	0.02	0.03
Step1X-Edit [36]	✓	24.17	105.14	7.03	4.94	5.71	15.27	44.93	7.50	6.89	7.11
Gemini-2-Flash [52]	✓	23.06	90.96	7.65	6.77	7.00	16.74	53.76	7.33	7.17	7.19
GPT-4o [24]	✓	22.77	91.79	8.52	7.37	7.85	15.67	47.60	8.07	7.87	7.95
JarvisArt	✓	12.66	31.88	6.19	8.51	6.67	7.75	12.38	7.54	8.46	7.91

Table 6: Quantitative evaluation on MIT-FiveK [3]. We highlight the **best** and **second-best** instruction-based results. SC, PQ, and O refer to the metrics evaluated by Gemini-2-Flash.

Method	Instruction	$L1_{\times 10^2} \downarrow$	$L2_{\times 10^3} \downarrow$	SC $\uparrow$	PQ $\uparrow$	O $\uparrow$
InstructPix2Pix [2]	✓	16.23	49.54	6.36	8.34	7.15
MagicBrush [68]	✓	17.29	53.45	4.92	5.50	4.95
OmniGen [57]	✓	28.53	128.59	3.12	2.48	2.57
VARGPT-v1.1 [76]	✓	26.96	117.16	2.94	2.00	2.29
Step1X-Edit [36]	✓	22.08	91.72	7.20	8.48	7.69
Gemini-2-Flash [52]	✓	18.69	61.27	7.86	9.22	8.47
GPT-4o [24]	✓	21.49	78.11	8.72	9.76	9.22
JarvisArt	✓	12.98	30.05	7.36	9.82	8.48

C.2 Prompt for MLLM-based Metrics

As shown in Figure 20, we present the evaluation prompts utilized for both scene-level and region-level assessments of the Semantic Consistency (SC) and Perceptual Quality (PQ) metrics. Notably, the overall score is calculated as $O = \sqrt{SC \times PQ}$.

D Additional Experimental Results

D.1 Additional Quantitative Evaluation by Qwen2.5-VL-72B

As shown in Table 5, to further evaluate MLLM-based metrics, we conducted an additional quantitative analysis using Qwen2.5-VL-72B [1]. Our findings suggest that such metrics may be unreliable, struggle to effectively reflect a model’s instruction-following capability. Despite this, our model demonstrates instruction-following performance comparable to that of contemporary SOTA closed-source model GPT-4o, while achieving a significant improvement in content fidelity.

D.2 Examples of Intricate Retouching Tasks with JarvisArt

Figures 12-15 present the challenging retouching examples, which involve both global and local editing demands, as well as vague user instructions. JarvisArt excels in understanding these ambiguous intentions, applying modifications at both the scene and region levels, and delivering visually effective results in the final images.

D.3 More Visual Comparisons

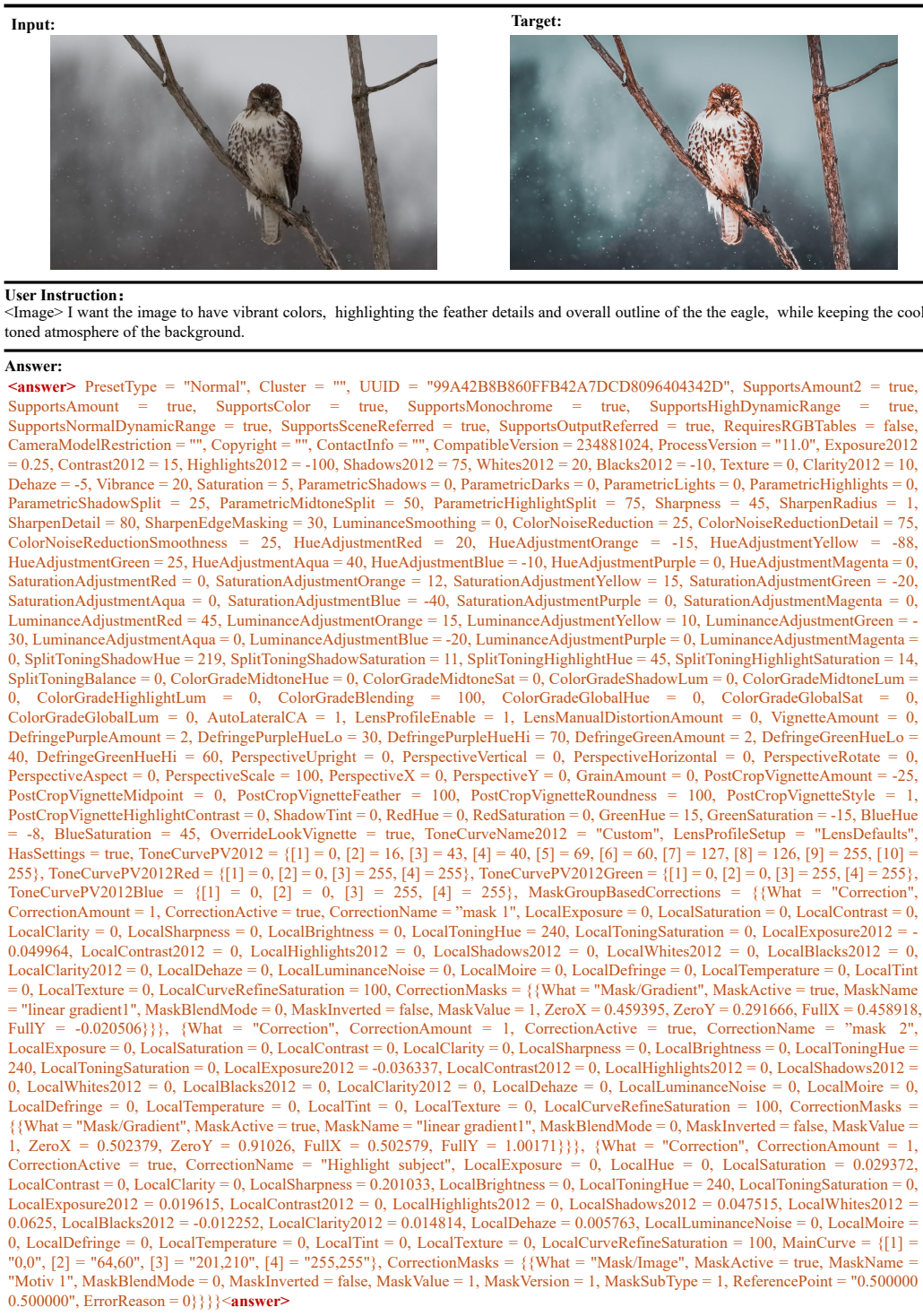
Figures 16-19 present additional photo retouching results from the MMArt-Bench and MIT-FiveK, highlighting the superiority of JarvisArt in terms of instruction adherence, content fidelity, and visual appeal, while also achieving the style most similar to the target image. Notably, we also include a visual comparison with two commercial editing tools: Adobe Lightroom and Google Photo’s auto-retouching modes.

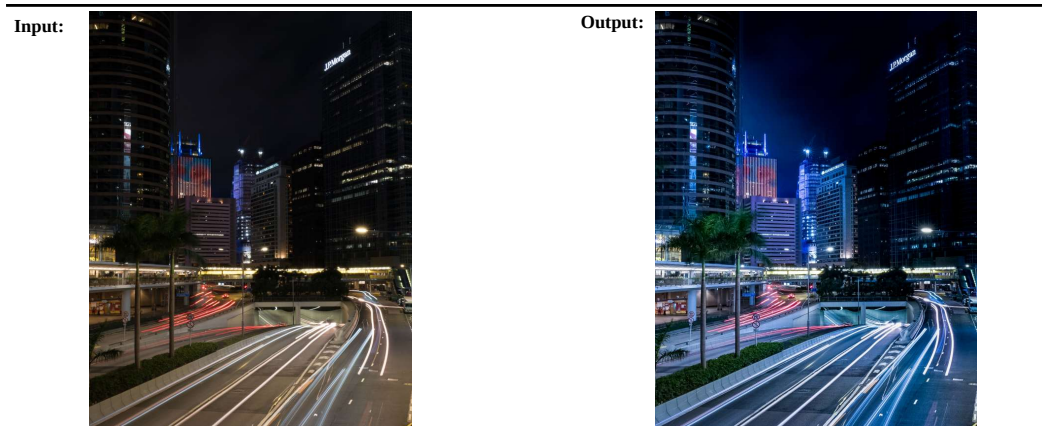
D.4 Comparison on MIT-FiveK

To assess the generalization ability of our system, we conduct comprehensive qualitative and visual comparisons on the MIT-FiveK [3] benchmark dataset. Specifically, we randomly select 50 data samples from MIT-FiveK [3] and generate corresponding user instructions based on the source images, expert C’s reference images, and the associated retouching parameter configurations (see Section 3.2 for implementation details). As shown in Table 6 and Figures 18 and 19, our system achieves state-of-the-art performance in both instruction-following and content preservation metrics. These results confirm the robustness and effectiveness of our method in faithfully executing user instructions while maintaining original image content. Furthermore, our approach consistently outperforms existing baselines on multiple real-world benchmarks, highlighting its strong generalization ability and practical applicability in interactive photo retouching.

Input:	Output:
	
User Instruction: <Image> I wanted the scene to pop more, making the cathedral stand out with enhanced details. The water needed a cooler tone, while the bridge and path should appear darker for depth. Overall, I aimed for a balanced look that emphasizes the architecture and adds a cinematic feel.	
Answer with CoT annotation: <think>Alright, I've got this image showing a bridge, cathedral, and river. The goal? Enhance the cathedral's details, cool the water, darken the bridge/path, and achieve a cinematic feel. The image appears muted, with vibrant water needed. For the cathedral, I'll boost clarity to 10 and use sharpening. To cool the water, I'll adjust blue saturation to -40 and apply split toning. Darkening the bridge/path involves local exposure reductions in masks. A cinematic look comes from increasing vibrance to 20, adjusting HSL, and using a vignette. Highlights at -100 may be too much; I'll tweak that. Yellow hue shift (-88) might alter the cathedral unnaturally. I'll check these and refine. Composition-wise, the vignette and local darkening will focus on the cathedral. Final tweaks ensure the image meets the vision.<think> <answer>CompatibleVersion = 251658240, ProcessVersion = "11.0", WhiteBalance = "Custom", IncrementalTemperature = -9, IncrementalTint = -3, Exposure2012 = 1.56, Contrast2012 = 54, Highlights2012 = -100, Shadows2012 = 100, Whites2012 = -96, Blacks2012 = -30, Texture = 0, Clarity2012 = 31, Dehaze = 0, Vibrance = -21, Saturation = 100, ParametricShadows = 0, ParametricDarks = 0, ParametricLights = 0, ParametricHighlights = 0, ParametricShadowSplit = 25, ParametricMidtoneSplit = 50, ParametricHighlightSplit = 75, Sharpness = 40, SharpenRadius = 0.5, SharpenDetail = 100, SharpenEdgeMasking = 60, LuminanceSmoothing = 27, LuminanceNoiseReductionDetail = 100, LuminanceNoiseReductionContrast = 0, ColorNoiseReduction = 0, HueAdjustmentRed = 0, HueAdjustmentOrange = 0, HueAdjustmentYellow = -100, HueAdjustmentGreen = -100, HueAdjustmentAqua = 0, HueAdjustmentBlue = -11, HueAdjustmentPurple = 0, HueAdjustmentMagenta = 0, SaturationAdjustmentRed = 100, SaturationAdjustmentOrange = 79, SaturationAdjustmentYellow = 12, SaturationAdjustmentGreen = -100, SaturationAdjustmentAqua = 34, SaturationAdjustmentBlue = -14, SaturationAdjustmentPurple = -100, SaturationAdjustmentMagenta = -100, LuminanceAdjustmentRed = 0, LuminanceAdjustmentOrange = 0, LuminanceAdjustmentYellow = 0, LuminanceAdjustmentGreen = 0, LuminanceAdjustmentAqua = 0, LuminanceAdjustmentBlue = -28, LuminanceAdjustmentPurple = 0, LuminanceAdjustmentMagenta = 0, SplitToningShadowHue = 217, SplitToningShadowSaturation = 6, SplitToningHighlightHue = 0, SplitToningHighlightSaturation = 0, SplitToningBalance = 0, ColorGradeMidtoneHue = 0, ColorGradeMidtoneSat = 0, ColorGradeShadowLum = 0, ColorGradeMidtoneLum = 0, ColorGradeHighlightLum = 0, ColorGradeBlending = 50, ColorGradeGlobalHue = 0, ColorGradeGlobalSat = 0, ColorGradeGlobalLum = 0, AutoLateralCA = 0, LensProfileEnable = 0, LensManualDistortionAmount = 0, VignetteAmount = 0, DefringePurpleAmount = 0, DefringePurpleHueLo = 30, DefringePurpleHueHi = 70, DefringeGreenAmount = 0, DefringeGreenHueLo = 40, DefringeGreenHueHi = 60, PerspectiveUpright = 0, PerspectiveVertical = 0, PerspectiveHorizontal = 0, PerspectiveRotate = 0, PerspectiveAspect = 0, PerspectiveScale = 100, PerspectiveX = 0, PerspectiveY = 0, GrainAmount = 0, PostCropVignetteAmount = -17, PostCropVignetteMidpoint = 50, PostCropVignetteFeather = 50, PostCropVignetteRoundness = 0, PostCropVignetteStyle = 1, PostCropVignetteHighlightContrast = 0, ShadowTint = 0, RedHue = 0, RedSaturation = 0, GreenHue = 0, GreenSaturation = 0, BlueHue = 0, BlueSaturation = 0, OverrideLookVignette = true, ToneCurveName2012 = "Custom", CameraProfile = "Embedded", CameraProfileDigest = "54650A341B5B5CCAE8442D0B43A92BCE", HasSettings = true, AlreadyApplied = false, ToneCurvePV2012 = {[1] = 0, [2] = 11, [3] = 152, [4] = 106, [5] = 192, [6] = 165, [7] = 255, [8] = 255,}, ToneCurvePV2012Red = {[1] = 0, [2] = 0, [3] = 255, [4] = 255,}, ToneCurvePV2012Green = {[1] = 0, [2] = 0, [3] = 255, [4] = 255,}, ToneCurvePV2012Blue = {[1] = 0, [2] = 0, [3] = 255, [4] = 255,}, Look = {Name = "Modern 08", Amount = 1, Cluster = "Adobe", UUID = "DA1C3775662D6B6A75F8BC2CEEB3724A", SupportsMonochrome = false, Copyright = "© 2018 Adobe Systems, Inc.", Group = [{"x-default"}] = "Modern",}, Parameters = {Version = "15.2", ProcessVersion = "11.0", ConvertToGrayscale = false, LookTable = "E1095149FDB39D7A057BAB208837E2E1", RGBTable = "6133E9CB13FA32712A5AC579D110EC44", RGBTableAmount = 0.75,},<answer>	

Figure 10: Examples of MMarT data annotated with Chain-of-Thought (CoT) reasoning.





User Instruction:

<Image> I want a cyberpunk vibe with neon lights and sharp contrasts for a futuristic city feel.

JarvisArt:

<think>Alright, the user wants a cyberpunk vibe with neon lights and sharp contrasts. The image features illuminated signs and light trails at night, perfect for this theme. I'll start by adjusting the white balance to 3200K for a cooler tone. Next, I'll recover highlights in neon lights and open shadows slightly to keep details. Increasing contrast will enhance light-dark differences without harshness. Vibrance and saturation boosts will make colors pop, especially neons. Clarity and dehaze adjustments will sharpen details and cut through haze. Using the "Adobe Color" look with an amount of 1 will complement these changes. Local masks will refine specific areas, ensuring consistency. Checking sharpness and noise reduction will maintain quality. Finally, applying the provided tone curve values will shape brightness and contrast perfectly. This approach captures the futuristic city feel the user desires.<think>

<answer> WhiteBalance = "Custom", Temperature = 3200, Tint = 10, Exposure2012 = 0.96, Contrast2012 = 4, Highlights2012 = -58, Shadows2012 = 58, Whites2012 = 37, Blacks2012 = -13, Clarity2012 = 20, Dehaze = 25, Vibrance = 15, Saturation = 2, ParametricShadowSplit = 25, ParametricMidtoneSplit = 50, ParametricHighlightSplit = 75, Sharpness = 40, SharpenRadius = 1, SharpenDetail = 25, ColorNoiseReduction = 25, ColorNoiseReductionDetail = 50, ColorNoiseReductionSmoothness = 50, ColorGradeBlending = 50, DefringePurpleHueLo = 30, DefringePurpleHueHi = 70, DefringeGreenHueLo = 40, DefringeGreenHueHi = 60, PerspectiveScale = 100, ToneCurveName2012 = "Linear", HasSettings = true, MaskGroupBasedCorrections = {{What = "Correction", CorrectionAmount = 1, CorrectionActive = true, CorrectionName = "Mask 1", LocalExposure2012 = -0.25, LocalHighlights2012 = -0.217391, LocalShadows2012 = -0.25, LocalWhites2012 = -0.25, LocalBlacks2012 = -0.114906, CorrectionMasks = {{What = "Mask/Gradient", MaskActive = true, MaskName = "Linear Gradient 1", MaskBlendMode = 0, MaskInverted = false, MaskValue = 1, ZeroX = 0.630042, ZeroY = 0.503213, FullX = 0.799436, FullY = 0.387017}}}, {What = "Correction", CorrectionAmount = 1, CorrectionActive = true, CorrectionName = "Mask 2", LocalExposure2012 = -0.1, LocalShadows2012 = -0.153061, LocalWhites2012 = -0.153061, LocalBlacks2012 = -0.122449, LocalClarity2012 = 0.082163, LocalDehaze = 0.082163, LocalTexture = 0.102164, CorrectionMasks = {{What = "Mask/Gradient", MaskActive = true, MaskName = "Linear Gradient 1", MaskBlendMode = 0, MaskInverted = false, MaskValue = 1, ZeroX = 0.596549, ZeroY = 0.53203, FullX = 0.799826, FullY = 0.682565}}}}, Look = {Name = "Adobe Color", Amount = 1, UUID = "B952C23111CD8E0ECCF14B86BAA7077", SupportsAmount = false, SupportsMonochrome = false, SupportsOutputReferred = false, Group = [{"x-default"} = "Profiles"]}, Parameters = {Version = "15.0", ProcessVersion = "11.0", ConvertToGrayscale = false, CameraProfile = "Adobe Standard", LookTable = "E1095149FDB39D7A057BAB208837E2E1", ToneCurvePV2012 = {[1] = 0, [2] = 0, [3] = 22, [4] = 16, [5] = 40, [6] = 35, [7] = 127, [8] = 127, [9] = 224, [10] = 230, [11] = 240, [12] = 246, [13] = 255, [14] = 255}}}<answer>

Figure 12: An example of JarvisArt empowering users to achieve interactive and interpretable editing, transforming their ambiguous intentions into artistic visual outcomes.



User Instruction:

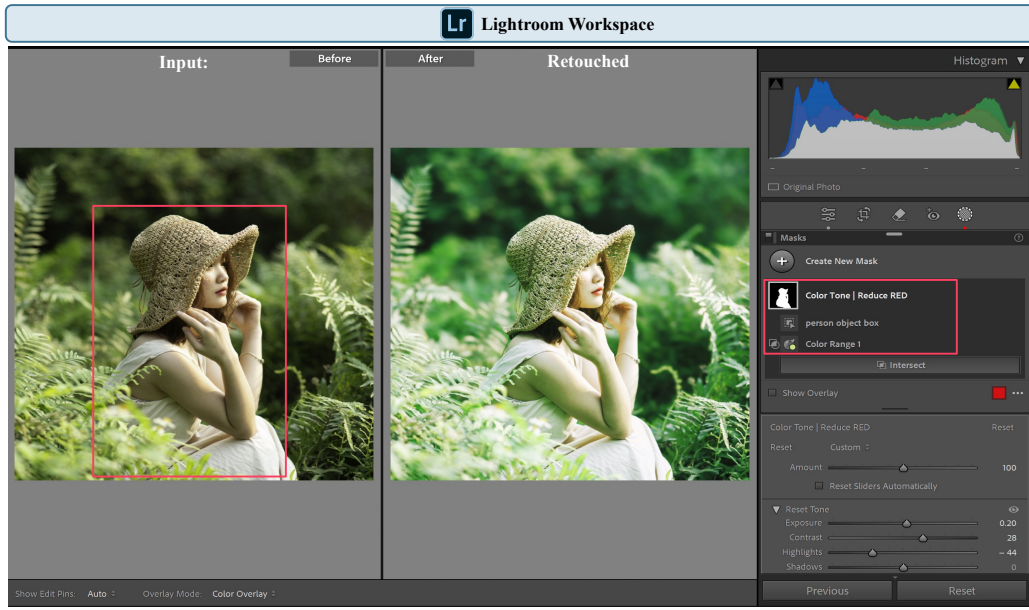
<Image> Make the image evoke a sense of loneliness and introspection, with soft lighting highlighting the solitary figure within the <box>0.446, 0.712, 0.540, 0.841</box>.

JarvisArt:

<think> Alright, focusing on evoking loneliness and introspection in the image of a solitary figure in a forest. The person is centered at <box>0.446, 0.712, 0.540, 0.841</box>. The dense forest sets a quiet mood. To enhance solitude, I'll adjust lighting around this area. A slight warming with White Balance settings (Incremental Temperature 15, Tint 10) adds melancholy. Exposure2012 at 1.25 brightens the scene, while Contrast2012 at 17 enhances details. Shadows2012 set to 47 reveals more around the person. Vibrance increased by 10 boosts less saturated colors subtly. HueAdjustment values (-100 for Yellow, Green, Purple; -36 for Aqua) harmonize the palette. PostCropVignetteAmount at -21 darkens edges, drawing focus to the center. These adjustments create a poignant, introspective atmosphere.</think>

<answer> PresetType = "Normal", WhiteBalance = "Custom", IncrementalTemperature = 15, IncrementalTint = 10, Exposure2012 = 1.25, Contrast2012 = 17, Highlights2012 = -27, Shadows2012 = 47, Whites2012 = 20, Blacks2012 = -18, Clarity2012 = 15, Dehaze = 10, Vibrance = 10, Saturation = 1, ParametricShadowSplit = 25, ParametricMidtoneSplit = 50, ParametricHighlightSplit = 75, Sharpness = 55, SharpenRadius = 1, SharpenDetail = 25, LuminanceSmoothing = 25, LuminanceNoiseReductionDetail = 50, ColorNoiseReduction = 25, ColorNoiseReductionDetail = 50, ColorNoiseReductionSmoothness = 50, HueAdjustmentYellow = -100, HueAdjustmentGreen = -100, HueAdjustmentAqua = -36, HueAdjustmentBlue = -22, SaturationAdjustmentOrange = -10, SaturationAdjustmentYellow = -10, SaturationAdjustmentGreen = -10, SaturationAdjustmentAqua = -10, SaturationAdjustmentBlue = -10, SaturationAdjustmentPurple = -100, SaturationAdjustmentMagenta = -100, LuminanceAdjustmentOrange = 20, LuminanceAdjustmentYellow = 20, LuminanceAdjustmentGreen = 20, LuminanceAdjustmentAqua = 10, LuminanceAdjustmentBlue = 10, LuminanceAdjustmentPurple = 20, LuminanceAdjustmentMagenta = 20, SplitToningShadowHue = 222, SplitToningShadowSaturation = 13, SplitToningHighlightHue = 10, SplitToningBalance = 22, ColorGradeShadowLum = -25, ColorGradeBlending = 100, AutoLateralCA = 1, DefringePurpleHueLo = 30, DefringePurpleHueHi = 70, DefringeGreenHueLo = 40, DefringeGreenHueHi = 60, PerspectiveScale = 100, PostCropVignetteAmount = -21, PostCropVignetteMidpoint = 50, PostCropVignetteFeather = 50, PostCropVignetteStyle = 1, RedHue = 25, RedSaturation = -8, GreenHue = 45, GreenSaturation = -15, BlueHue = -15, BlueSaturation = 40, CurveRefineSaturation = 100, OverrideLookVignette = true, ToneCurveName2012 = "Custom", HasSettings = true, ToneCurvePV2012 = {[1] = 0, [2] = 28, [3] = 54, [4] = 62, [5] = 176, [6] = 164, [7] = 255, [8] = 224}, ToneCurvePV2012Red = {[1] = 0, [2] = 0, [3] = 40, [4] = 12, [5] = 84, [6] = 58, [7] = 123, [8] = 125, [9] = 186, [10] = 196, [11] = 255, [12] = 255}, ToneCurvePV2012Green = {[1] = 0, [2] = 0, [3] = 40, [4] = 14, [5] = 83, [6] = 59, [7] = 123, [8] = 126, [9] = 172, [10] = 185, [11] = 255, [12] = 255}, ToneCurvePV2012Blue = {[1] = 0, [2] = 0, [3] = 39, [4] = 13, [5] = 87, [6] = 65, [7] = 123, [8] = 124, [9] = 183, [10] = 199, [11] = 255, [12] = 255}, MaskGroupBasedCorrections = {[What = "Correction", CorrectionAmount = 1, CorrectionActive = true, CorrectionName = "Mask 1", LocalSaturation = 0.23, LocalExposure2012 = 0.15, LocalShadows2012 = -0.60648, LocalBlacks2012 = -0.33333, LocalTemperature = 0.32, LocalTint = 0.25, LocalCurveRefineSaturation = 100, CorrectionMasks = {[What = "Mask/Image", MaskActive = true, MaskName = "Person 1", MaskBlendMode = 0, MaskInverted = false, MaskValue = 1, MaskVersion = 1, MaskSubType = 0, ReferencePoint = "0.500000 0.500000", ErrorReason = 0, Gesture = {[What = "Mask/Polygon", MaskActive = true, MaskBlendMode = 0, MaskInverted = false, MaskValue = 1, Points = {[X = 0.44602, Y = 0.712727], [X = 0.540894, Y = 0.712727], [X = 0.44602, Y = 0.840727], [X = 0.540894, Y = 0.840727]}]}]}]}</answer>

Figure 13: An example of JarvisArt empowering users to achieve interactive and interpretable editing, transforming their ambiguous intentions into artistic visual outcomes.



User Instruction:
 <Image> I want this image to have a soft, misty style. In the <box>[0.3, 0.25, 0.7, 0.6]</box> area, which is the face and upper body of the figure, I wanted the skin color to be smoother and more natural. The overall colors should blend softly, with a warm, green feeling that evokes nostalgia. The foliage in the background should be vibrant but not too strong, creating a serene atmosphere.

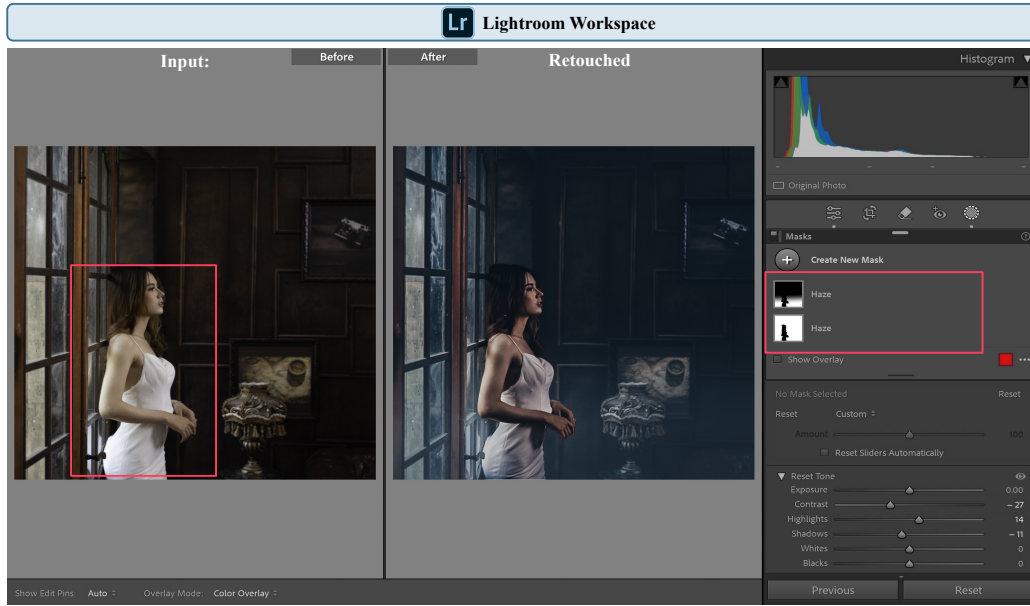
Answer with CoT annotation:

<think>Alright, I've got this image to edit for a soft, misty style. The user wants smoother skin tones and gentle color blending with a warm, vintage feel, focusing on the person's face and upper body in <box>0.3 0.25 0.7 0.6</box>. The background foliage should stay vibrant but not overpowering. First, I'll adjust the white balance for warmth. Then, apply a slight blur or glow effect for the misty style. For the skin tones, I'll use masks to reduce saturation and smooth textures in the specified area. Next, I'll tweak the tone curve to soften contrasts and warm up highlights and midtones. Background colors will be balanced to complement the foreground without being too intense. Adding subtle film grain can enhance the vintage look. Finally, I'll ensure all elements harmonize for a serene, nostalgic atmosphere. These steps align with the user's vision while preserving the original image's quality.</think>

<answer>PresetType = "Normal", Cluster = "", SupportsAmount2 = true, SupportsAmount = true, SupportsColor = true, SupportsMonochrome = true, SupportsHighDynamicRange = true, SupportsNormalDynamicRange = true, SupportsSceneReferred = true, SupportsOutputReferred = true, ProcessVersion = "15.4", WhiteBalance = "As Shot", ParametricShadowSplit = 25, ParametricMidtoneSplit = 50, ParametricHighlightSplit = 75, ColorGradeBlending = 50, DefringePurpleHueLo = 30, DefringePurpleHueHi = 70, DefringeGreenHueLo = 40, DefringeGreenHueHi = 60, PerspectiveScale = 100, ToneCurveName2012 = "Linear", HasSettings = true, Look = { Name = "Adobe Color", Amount = 1, Stubbed = true, Parameters = { CompatibleVersion = 251920384, ProcessVersion = "15.4", CurveRefineSaturation = 50, ConvertToGrayscale = false, RGBTable = "F4B5250D16A58B460C36CEC16C0627D2", ToneCurvePV2012 = { [1] = 0, [2] = 0, [3] = 16, [4] = 24, [5] = 64, [6] = 72, [7] = 128, [8] = 128, [9] = 192, [10] = 176, [11] = 244, [12] = 234, [13] = 255, [14] = 255 }, ToneCurvePV2012Red = { [1] = 0, [2] = 0, [3] = 255, [4] = 255 }, ToneCurvePV2012Green = { [1] = 0, [2] = 0, [3] = 255, [4] = 255 }, ToneCurvePV2012Blue = { [1] = 0, [2] = 0, [3] = 255, [4] = 255 } }, LensBlur = { Version = 1, Active = true, ImageOrientation = 0, FocalRange = "0 0 100 100", BlurAmount = 50, BokehShape = 0, BokehShapeDetail = 0, HighlightsThreshold = 50, HighlightsBoost = 50, CatEyeAmount = 0, CatEyeScale = 100, BokehAspect = 0, BokehRotation = 0, SphericalAberration = 0 }, MaskGroupBasedCorrections = { { What = "Correction", CorrectionAmount = 1, CorrectionActive = true, CorrectionName = "Color Tone | Reduce RED", LocalToningHue = 119, LocalToningSaturation = 0.1, LocalTint = -0.05, LocalCurveRefineSaturation = 100, CorrectionMasks = { { What = "Mask/RangeMask", MaskActive = true, MaskName = "Color Range 1", MaskBlendMode = 1, MaskInverted = true, MaskValue = 0, CorrectionRangeMask = { Version = 3, Type = 1, ColorAmount = 0.494949, Invert = true, SampleType = 0, PointModels = { "0.595958 0.896660 0.597516 0.480823 0.955119 0" } }, { What = "Mask/Image", MaskActive = true, MaskName = "person object box", MaskBlendMode = 1, MaskInverted = true, MaskValue = 0, MaskVersion = 1, MaskSubType = 0, ReferencePoint = "0.500000 0.500000", ErrorReason = 0, Gesture = { { What = "Mask/Polygon", MaskActive = true, MaskBlendMode = 0, MaskInverted = false, MaskValue = 1, Points = { { X = 0.3, Y = 0.25 }, { X = 0.7, Y = 0.25 }, { X = 0.3, Y = 0.6 }, { X = 0.7, Y = 0.6 } } } } } } } } }

</answer>

Figure 14: Editing results with JarvisArt are generated under complex prompts, with all retouching operations performed in a Lightroom environment, allowing for iterative adjustments.



User Instruction:

<Image> My goal was to create an atmospheric, nostalgic look. For the area <box>[0.3, 0.25, 0.7, 0.6]</box> around her face and shoulders, I wanted smoother textures and a subtle haze effect. The entire image should evoke a sense of timelessness with muted colors and enhanced contrast, making it feel like a classic portrait.

Answer with CoT annotation:

<think>Alright, let's dive into this image. The user wants a nostalgic feel with smooth textures and a subtle haze around the person's face and shoulders in <box>0.3 0.25 0.7 0.6</box>. We'll warm the white balance to 5185K for that vintage vibe. A slight overexposure of 0.23 brightens the subject, while contrast at 7 adds depth. Highlights reduced by -71 and shadows boosted by 50 preserve detail. Texture at 25 and clarity at 14 enhance details globally, but local dehaze softens the specified area. Vibrance up by 20 enriches colors subtly. Split toning with shadow hue at 229 and highlight hue at 48 adds a vintage touch. A dark vignette draws focus to the center. These tweaks balance global adjustments with targeted corrections, achieving the classic portrait look the user desires.</think>

<answer>WhiteBalance = "Custom", Temperature = 5185, Tint = 10, Exposure2012 = 0.23, Contrast2012 = 7, Highlights2012 = -71, Shadows2012 = 50, Whites2012 = 12, Blacks2012 = -18, Texture = 25, Clarity2012 = 14, Dehaze = 11, Vibrance = 20, Saturation = 3, ParametricShadowSplit = 25, ParametricMidtoneSplit = 50, ParametricHighlightSplit = 75, Sharpness = 40, SharpenRadius = 1, SharpenDetail = 25, LuminanceSmoothing = 13, LuminanceNoiseReductionDetail = 50, ColorNoiseReduction = 25, ColorNoiseReductionDetail = 50, ColorNoiseReductionSmoothness = 50, HueAdjustmentOrange = -11, HueAdjustmentYellow = -3, SaturationAdjustmentOrange = -20, SaturationAdjustmentYellow = -50, SaturationAdjustmentBlue = -31, LuminanceAdjustmentOrange = -30, LuminanceAdjustmentYellow = 12, LuminanceAdjustmentAqua = -12, LuminanceAdjustmentBlue = -30, SplitToningShadowHue = 229, SplitToningShadowSaturation = 13, SplitToningHighlightHue = 48, SplitToningHighlightSaturation = 10, ColorGradeBlending = 100, AutoLateralCA = 1, DefringePurpleHueLo = 30, DefringePurpleHueHi = 70, DefringeGreenHueLo = 40, DefringeGreenHueHi = 60, PerspectiveScale = 100, PostCropVignetteAmount = -11, PostCropVignetteMidpoint = 50, PostCropVignetteFeather = 50, PostCropVignetteStyle = 1, RedHue = 30, RedSaturation = -10, GreenHue = 40, GreenSaturation = -10, BlueHue = -25, BlueSaturation = -10, OverrideLookVignette = true, ToneCurveName2012 = "Custom", HasSettings = true, ToneCurvePV2012 = { [1] = 0, [2] = 29, [3] = 48, [4] = 48, [5] = 127, [6] = 126, [7] = 190, [8] = 192, [9] = 255, [10] = 255 }, ToneCurvePV2012Red = { [1] = 0, [2] = 0, [3] = 47, [4] = 27, [5] = 115, [6] = 125, [7] = 159, [8] = 173, [9] = 255, [10] = 255 }, ToneCurvePV2012Green = { [1] = 0, [2] = 0, [3] = 43, [4] = 25, [5] = 113, [6] = 121, [7] = 160, [8] = 174, [9] = 255, [10] = 255 }, ToneCurvePV2012Blue = { [1] = 0, [2] = 0, [3] = 39, [4] = 25, [5] = 113, [6] = 124, [7] = 159, [8] = 174, [9] = 255, [10] = 255 }, MaskGroupBasedCorrections = { { What = "Correction", CorrectionAmount = 1, CorrectionActive = true, CorrectionName = "Haze", LocalToningHue = 240, LocalBlacks2012 = 0.0675, LocalClarity2012 = -0.198322, LocalDehaze = -0.208022, LocalTemperature = -0.1024, LocalTint = -0.09851, LocalTexture = -0.250785, CorrectionMasks = { { What = "Mask/Image", MaskActive = true, MaskName = "Subject 1", MaskBlendMode = 0, MaskInverted = true, MaskValue = 1, MaskVersion = 1, MaskSubType = 1, ReferencePoint = "0.306641 0.521994", WholeImageArea = "0/1,0/1,1707/1,2560/1", Origin = "0,531", ModelVersion = 234881976 } } }, { What = "Correction", CorrectionAmount = 1, CorrectionActive = true, CorrectionName = "Haze", LocalToningHue = 240, LocalBlacks2012 = 0.175759, LocalClarity2012 = -0.284554, LocalDehaze = -0.364241, LocalTemperature = -0.163366, LocalTint = -0.188755, LocalTexture = -0.288385, CorrectionMasks = { { What = "Mask/Gradient", MaskActive = true, MaskName = "Linear Gradient 1", MaskBlendMode = 0, MaskInverted = false, MaskValue = 1, ZeroX = 0.426926, ZeroY = 0.47269, FullX = 0.44185, FullY = 0.872828 }, { What = "Mask/Image", MaskActive = true, MaskName = "Subject 1", MaskBlendMode = 1, MaskInverted = false, MaskValue = 0, MaskVersion = 1, MaskSubType = 1, ReferencePoint = "0.306641 0.521994", WholeImageArea = "0/1,0/1,1707/1,2560/1", Origin = "0,531", ModelVersion = 234881976 } } } }

</answer>

Figure 15: Editing results with JarvisArt are generated under complex prompts, with all retouching operations performed in a Lightroom environment, allowing for iterative adjustments.

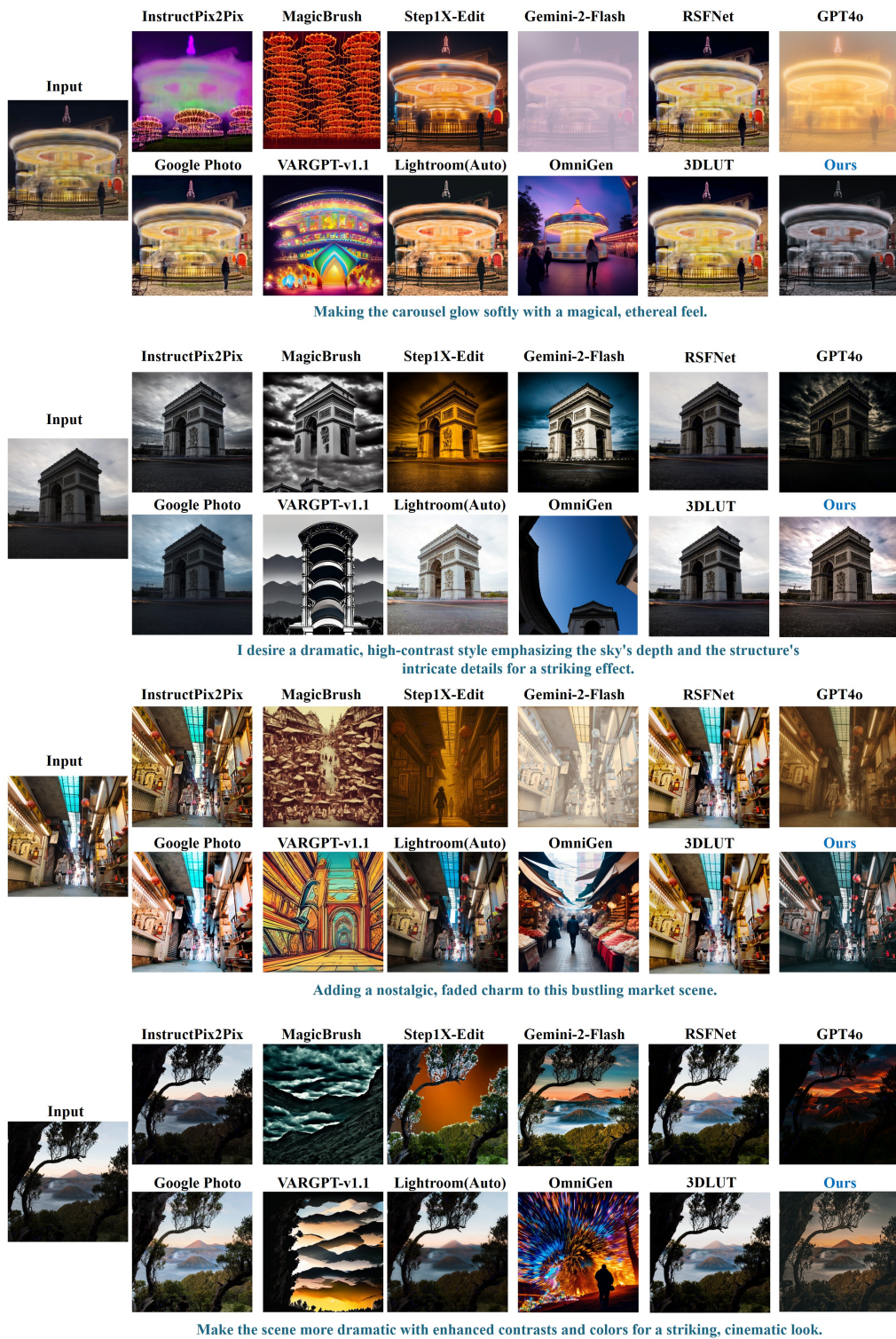
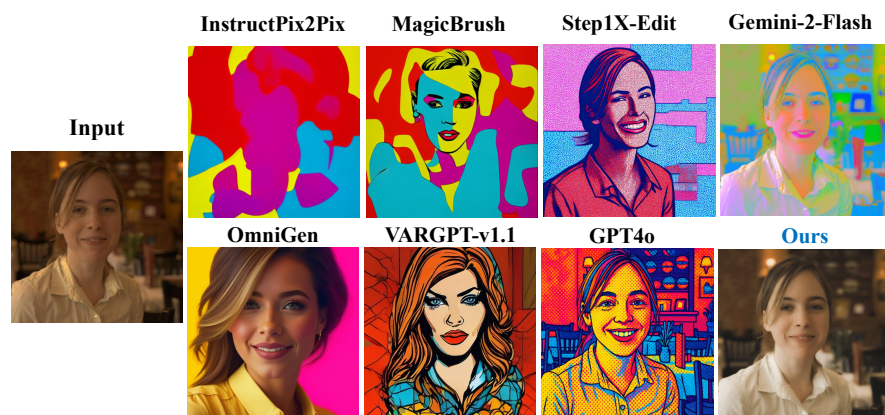


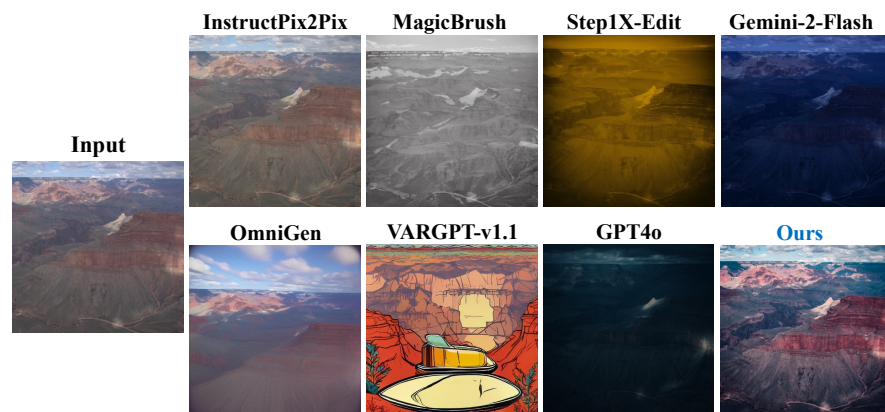
Figure 16: Visual comparisons of all state-of-the-art editing methods alongside two automatic retouching modes from commercial software.



Figure 17: Visual comparisons of all state-of-the-art editing methods alongside two automatic retouching modes from commercial software.



Make her look more vibrant and lively, like a pop art style.



I want a moody, atmospheric look with deeper shadows and cooler tones.



I aim for a melancholy blues style to evoke a sense of nostalgia and depth.



I want a dreamy, soft look with cooler tones for a nostalgic feel.

Figure 18: Visual comparisons of all instruction-based editing methods on MIT-FiveK [3].

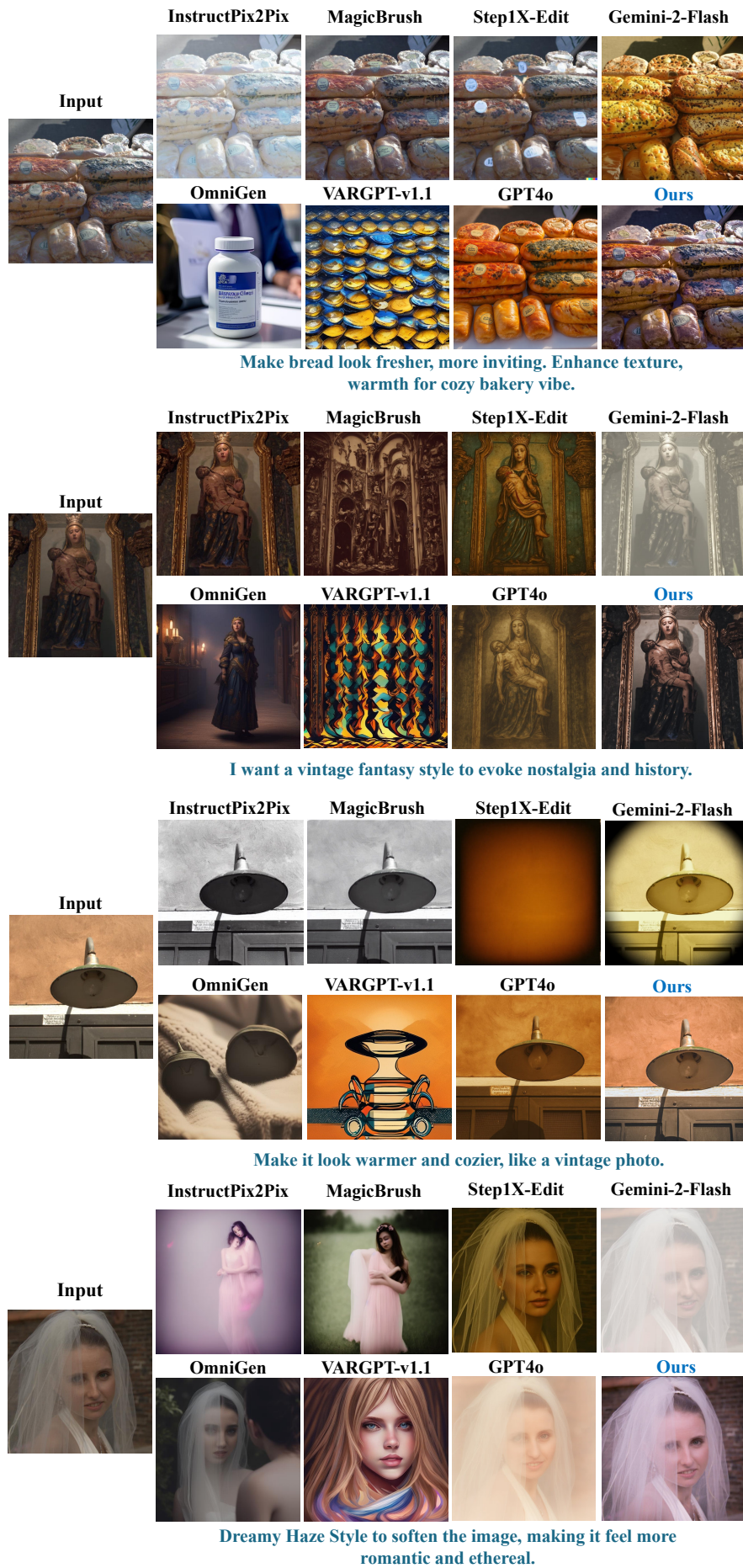


Figure 19: Visual comparisons of all instruction-based editing methods on MIT-FiveK [3].

Prompt for MLLM-based SC, PQ, and O

```

prompt_sc_pq_scene = """
You are a post-production specialist with expertise in enhancing photographic imagery through advanced digital editing techniques. We now need your help to evaluate the performance of an AI-powered image post-editing tool for photography.

INPUTS:
1. Two images will be provided: The first being the original photographic image and the second being an edited version of the first.
2. The editing instruction will be provided: The post-editing needs of photographic images expressed by users with no image processing knowledge.

METRICS (From scale 0 to 10):
User Instruction Satisfaction Score: A score from 0 to 10 will be given based on how well the edits follow the user's instructions.
- Users typically have both global and local editing requirements when working with photographic images. Therefore, this score should be evaluated holistically, taking into account the user's needs for both local and global adjustments, with equal importance given to each.
- 0 indicates that the edited image does not follow the editing instruction at all.
- 10 indicates that the edited image follow the editing instruction text perfectly.
Content Consistency Score: A second score from 0 to 10 will rate the consistency of image content before and after editing.
- Need to compare before and after images to assess content consistency.
- The edited image should maintain consistency in key visual elements such as the shape of landscapes, human figures (including posture, gender, and appearance), building structures, and other important features.
- The edited image needs to maintain the consistency of local details, such as letters on clothes, textures of buildings, etc.
- 0 indicates that the content of the image before and after editing is completely inconsistent.
- 10 indicates that the content of the edited image is exactly the same as the original image.

You will have to give your output in this way (Keep your reasoning concise and short.):
{
  "score": [...],
  "reasoning": "..."
}
Put the score in a list such that output score = [score1, score2], where 'score1' evaluates the User Instruction Satisfaction Score and 'score2' evaluates the Content Consistency Score.
Editing instruction: <The editing instruction of user>
"""

prompt_sc_pq_region = """
You are a post-production specialist with expertise in enhancing photographic imagery through advanced digital editing techniques. We now need your help to evaluate the performance of an AI-powered image post-editing tool for photography.

INPUTS:
1. Two images will be provided: The first being the original photographic image and the second being an edited version of the first.
2. The editing instruction will be provided: The post-editing needs of photographic images expressed by users with no image processing knowledge.

METRICS (From scale 0 to 10):
User Instruction Satisfaction Score: A score from 0 to 10 will be given based on how well the edits follow the user's instructions.
- Users typically have both global and local editing requirements when working with photographic images. Therefore, this score should be evaluated holistically, taking into account the user's needs for both local and global adjustments, with equal importance given to each.
- 0 indicates that the edited image does not follow the editing instruction at all.
- 10 indicates that the edited image follow the editing instruction text perfectly.
Content Consistency Score: A second score from 0 to 10 will rate the consistency of image content before and after editing.
- Need to compare before and after images to assess content consistency.
- The edited image should maintain consistency in key visual elements such as the shape of landscapes, human figures (including posture, gender, and appearance), building structures, and other important features.
- The edited image needs to maintain the consistency of local details, such as letters on clothes, textures of buildings, etc.
- 0 indicates that the content of the image before and after editing is completely inconsistent.
- 10 indicates that the content of the edited image is exactly the same as the original image.

IMPORTANT NOTES:
1. The focus of this assessment is the performance of AI-enabled post-processing tools for photographic images in response to users' local modification instructions.
2. The two images are the same localized areas of interest selected by the user in the original and edited images.
3. User editing instructions involve both global adjustments to the entire image and localized edits to specific regions of interest.

You will have to give your output in this way (Keep your reasoning concise and short.):
{
  "score": [...],
  "reasoning": "..."
}
Put the score in a list such that output score = [score1, score2], where 'score1' evaluates the User Instruction Satisfaction Score and 'score2' evaluates the Content Consistency Score.
Editing instruction: <The editing instruction of user>
"""

### Editing Evaluation Processing:
Scene-level editing evaluation: prompt_sc_pq_scene
Region-level editing evaluation: prompt_sc_pq_region

```

Figure 20: Prompt for MLLM-based metrics (SC, PQ) from scene-level and region-level.

Aesthetic Preset Recommendation Expert

You are a professional image aesthetic preset recommendation expert, skilled at analyzing image content and providing the best preset combination suggestions. You can identify the main subjects in images, and based on their categories and characteristics, precisely match suitable global and local presets from the preset library, offering users diverse and high-quality image optimization solutions.

Workflow

You will automatically match applicable image optimization presets based on the provided image file, detected subject box information, and preset library, and output the recommendations.

Input:

1. Source image.
2. Detected subject box information (category and confidence):

Examples:

```
[{
  "category": "person",
  "confidence": 0.95,
  "box": [0.2, 0.3, 0.6, 0.8]
}, ...]
```

3. Preset library (divided into four main categories: portrait, landscape, street, and food, each containing global and local presets):

```
"global": [
  {"id": "PERSON-G1",
   "name": "Portrait-B&W Background-Colored Subject",
   "function": "Preserves red-orange subject, desaturates background to black and white, high contrast to highlight people. Suitable for weddings, portrait close-ups, not suitable for colorful scenes or photos where landscape needs to be emphasized."},
  // This includes a total of 76 global presets across four categories:
  // 1. Portrait (PERSON-G series): 24 presets covering Japanese style, film simulation, fresh dreamy effects, and various portrait styles
  // 2. Landscape (SCENERY-G series): 10 presets including glacier waterfalls, golden sunset, cinematic city views, and other natural landscape styles
  // 3. Street (STREET-G series): 12 presets including urban day scenes, night scenes, cyberpunk, and various urban street photography styles
  // 4. Food (STILL-Life-G series): 6 presets including bright transparent, cinematic, warm tempting, and various food photography styles
  // Each preset has a unique ID, name, and function description detailing the effect, suitable scenarios, and unsuitable scenarios
  {...}]
"local": [
  {"id": "PERSON-L1",
   "name": "Skin Brightening-Facial Contour Enhancement",
   "function": "This is a professional portrait local adjustment preset that achieves dimensionality through fine-tuning of eyes (sharpness +0.21), eyebrows (exposure -0.1), and skin (exposure +0.14/contrast -0.2). Specifically enhances iris sharpness and brightens eye whites (exposure +0.04), while softening skin texture (texture -0.11) and reducing noise (-1). Suitable for commercial portraits and close-up portraits, highlighting facial features' dimensionality"},
  // This includes a total of 53 local presets across several categories:
  // 1. Portrait (PERSON-L series): 13 presets covering skin beautification, eye enhancement, and various portrait local adjustments
  // 2. Sky (ALL-L series SKY01 group): 22 presets including sky brightening, darkening, dramatic effects, etc.
  // 3. Subject emphasis (ALL-L series SUBJECT group): 13 presets including subject brightening, background weakening, etc.
  // 4. Lighting effects (ALL-L series LIGHT group): 18 presets including center stage, foreground darkening, fog effects, etc.
  // Each preset has a unique ID, name, and function description, with some presets having group information
  {...}]
```

Output Format:

```
{ "recommendations-1":
  {"target_class": "person",
   "confidence": 0.85,
   "global_presets":
   {"id": "G1",
    "name": "Natural Skin Tone Optimization" }
  ,
  "local_presets": [
    {"id": "L1",
     "name": "Female Skin Softening"},
    {"id": "L3",
     "name": "Light Source Enhancement"},
    {"local_apply_range": [x1,y1,x2,y2],
     "reasoning": "Detected high confidence portrait (85%), recommending global skin tone optimization with local skin softening and light source enhancement"},
    {...}]
  }
```

Working Rules**### Special Instructions**

1. Subject identification rule: Only when the detected subject confidence is greater than or equal to 0.85 will the subject be recognized as the current image's main scene type, and recommendations will be generated accordingly.
2. Preset matching principle: All recommended presets must strictly match the detected subject box information. For example, when the subject box only contains portrait category targets, presets for landscape, street, or food categories should not be recommended to ensure precision of processing effects.
3. Preset combination strategy: Recommendation plans should prioritize the combined use of global and local presets to achieve the best overall effect.
4. ID naming convention:
 - Global presets: "{category}-G{number}" (e.g., PERSON-G1)
 - Local presets: "{category}-L{number}" (e.g., PERSON-L2)
 - Category codes include: PERSON (portrait), SCENERY (landscape), STREET (street), FOOD (food), ALL (universal)
 - ALL category indicates the preset is applicable to all scene types
5. Application range definition:
 - Global presets: Applied to the entire image
 - Local presets: Use detection box precise coordinates [x1,y1,x2,y2] to define the application area
6. Multi-subject processing strategy: Regardless of whether there is a single or multiple subjects in the image, the system must generate multiple different recommendation plans. When there are multiple subjects, sort by confidence from high to low; when there is only a single subject, diverse processing plans should be generated around that subject. All recommendation plans must ensure diversity and non-repetition to provide rich editing options.
7. Overall effect priority: When evaluating preset combinations, consider the overall effect rather than the effect of a single step. For example, for photos with insufficient light, even if global preset G1 alone is not effective, it may produce an ideal effect when combined with local brightening preset L1, so comprehensive consideration of the final presentation is needed.
8. Recommendation plan naming convention: Use "recommendations-{number}" format (e.g., recommendations-1).
9. Preset combination limitations:
 - Each recommendation plan can only include one global preset
 - Try to select diverse recommendations, for example, if recommendations-1 has a yellowish portrait tone, recommendations-2 should choose a cool tone
 - Can include multiple different local presets, but all local presets must be applied to the same area (same apply_range); if you think the global preset is sufficient, local presets are not needed
 - Local presets must not be used repeatedly
 - Strictly avoid conflicting preset combinations in recommendation plans, for example:
 - * Should not simultaneously include presets for females and presets for males
 - * Should not simultaneously include presets with mutually canceling effects (such as using both saturation enhancement and saturation reduction presets)
 - Ensure all presets are logically compatible with each other, jointly serving to enhance the overall effect of the image
10. System output requirements: The system should directly generate recommendation plans in standard JSON format, without any additional text explanations or interpretive content, ensuring the output results can be directly parsed and used by programs.
11. Exposure balance principle: When recommending presets, the system should fully consider the current image's exposure value, intelligently selecting preset combinations that can balance image brightness, ensuring the final processing result is neither overexposed nor underexposed, thereby guaranteeing image detail retention and visual comfort.
12. Portrait exposure protection principle: For portrait subjects, the system must pay special attention to ensure preset combinations do not result in overexposure or excessive darkness in global or local facial areas after editing. For images that are already bright, presets that significantly increase brightness should be avoided; for images with already high contrast, presets that further enhance contrast should be avoided to prevent facial areas from becoming black or losing detail. When evaluating recommendation plans, the overall impact of global and local presets on portrait subjects must be comprehensively considered to ensure facial details are preserved, avoiding highlight overflow, facial overexposure, or excessive shadows, thereby guaranteeing natural texture, balanced tones, and detail expression in portraits.
13. Recommendation plan quantity principle: The system must generate 5 different recommendation plans.

Figure 21: Role-playing prompt for preset recommendation.

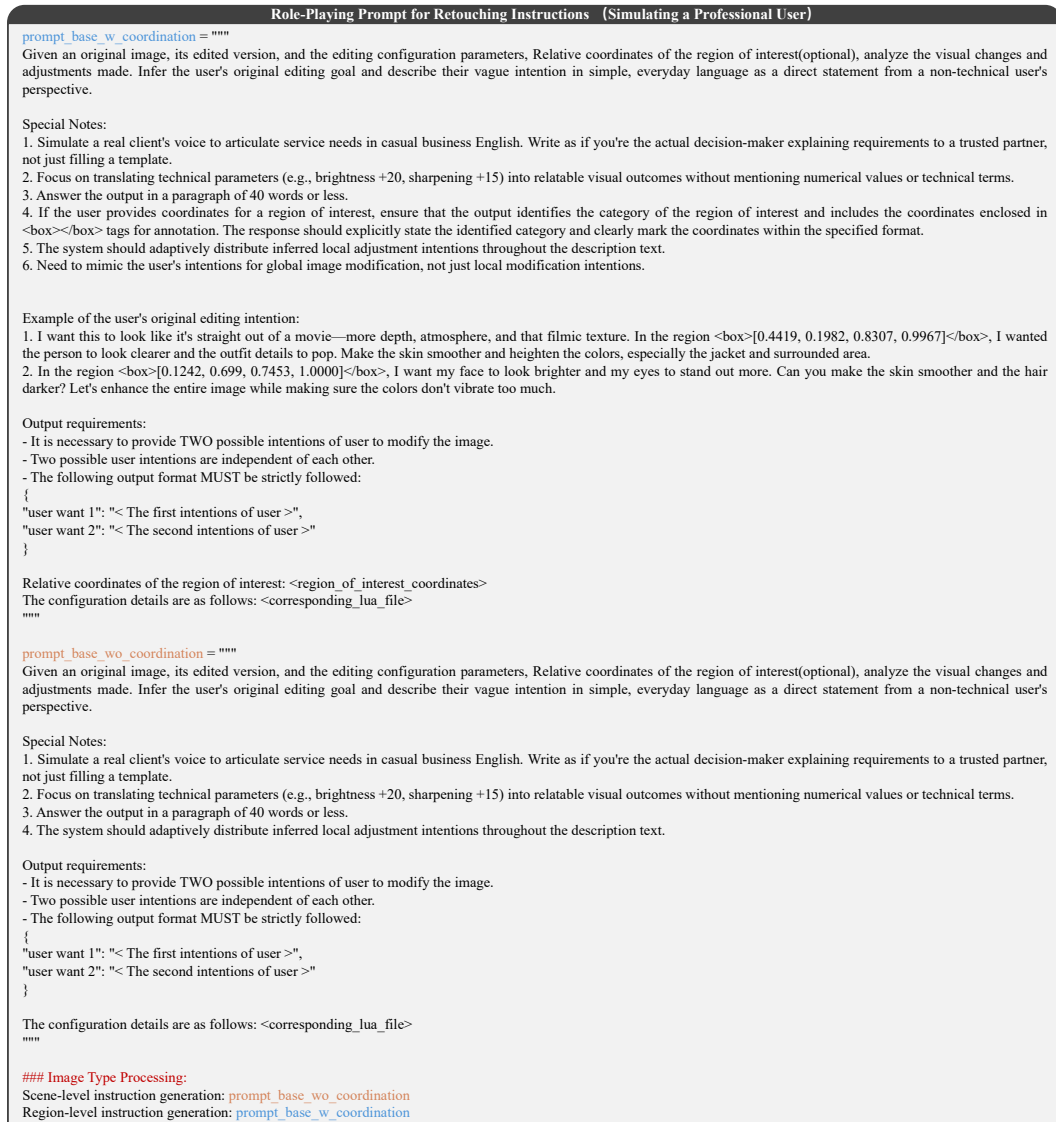


Figure 22: Prompt for simulating the professional user instructions

```

Style_expressions = ""
The style of editing the image that customers might express:
- Dreamy Haze Style: Soft focus, low contrast, soft tones, hazy halo -> Suitable for wedding photos, portraits, emotional expressions, etc.
- Melancholy Blues Style: Blue-grey tones, cold light, high shadows, emotional composition -> Suitable for street photography, portraits, storytelling, etc.
- Burning Emotion Style: Highly saturated orange-red, interlaced light and shadow, and a sense of restlessness -> Suitable for stage performances, music vision, dynamic moments, etc.
- Minimal Ethereal Style: Minimalist composition, blank space, soft filter, sense of tranquility -> Suitable for meditation, Zen, lifestyle content, etc.
- Futuristic Anxiety Style: Cold metallic tones, sharp lines, high contrast, digital noise -> Suitable for science fiction, technology products, cyberpunk style, etc.
- Vintage Fantasy Style: Yellowed/faded tones, graininess, old film texture -> Suitable for nostalgic advertising, historical narrative, illustration style, etc.
- Glitch Art Style: Digital glitch effect, color distortion, pixel distortion -> Suitable for electronic music, avant-garde fashion, Internet culture, etc.
- Liquid Abstraction Style: Strong sense of fluidity, color fusion, similar to watercolor/oil paint -> Suitable for art posters, packaging design, visual impact, etc.
- Negative Film Look Style: Similar to reversal film effect, reverse color, high contrast -> Suitable for documentary photography, street photography, alternative portraits, etc.
- Light Painting Abstraction Style: Emphasize light trails, blur, and dynamic smear -> Suitable for night photography, light shows, art installations, etc.
- Relief Texture Style: Pixel-level detail enhancement, three-dimensional sculpture, and outstanding material -> Suitable for sculptures, industrial design, fabric display, etc.
- Grain & Noise Style: Simulate film noise, rough texture, and handmade feel -> Suitable for documentary style, retro photography, documentary content, etc.
- Translucent Overlay Style: Multi-layer image overlay, semi-transparent channel, dreamy interlacing -> Suitable for conceptual photography, digital painting, mixed media, etc.
- Impressionist Style: Strong brushstrokes, vibrant colors, and soft edges -> Suitable for landscape photography, natural themes, artistic re-creation, etc.
- Expressionist Color Style: High contrast, exaggerated colors, emotional rendering -> Suitable for close-ups, dramatic photography, psychological expression, etc.
- Pop Art Style: Highly saturated color blocks, comic dots, and repeated patterns -> Suitable for celebrity portraits, pop culture, commercials, etc.
- Abstract Expressionism Style: Free brushstrokes, splashing feeling, emotional explosion -> Suitable for art photography, digital painting, emotional expression, etc.
- Ink Wash Style: Black and white gradient, white space, and fluidity of brush and ink -> Suitable for oriental culture, traditional elements, calligraphy, etc.
- Oil Painting Effect Style: Brushstroke simulation, thick painting, frame border -> Suitable for art exhibitions, high-end portraits, custom printing, etc.
- Watercolor Style: Fresh and transparent, edge diffusion, paper texture -> Suitable for children's photography, illustration, handicraft brands, etc.
- Pencil Sketch Style: Line outline, grayscale, hand-painted texture -> Suitable for teaching materials, character sketches, sketch displays, etc.
- Neon Glow Style: Neon texture, glowing edges, dark background -> Suitable for nightclub promotion, trendy clothing, urban photography, etc.
- Metallic Texture Style: High reflectivity, cold hard tone, metallic luster -> Suitable for luxury goods, electronic products, fashion blockbusters, etc.
- Motion Blur Style: Movement trajectory, speed sense, direction guidance -> Suitable for sports photography, car advertising, action scenes, etc.
- Aurora Vision Style: Flowing light bands, gradient changes of cold and warm, mysterious atmosphere -> Suitable for natural scenery, starry sky photography, meditation applications, etc.
- Hamada Hideki Style: Fresh, soft, natural and emotional -> Suitable for photos with a girlish or youthful feel, etc.
- Fujifilm Style: Make portraits, landscapes or daily photos present natural, delicate and emotional color expression -> Suitable for style requirements such as "movie feel", "Japanese fresh style" or "retro atmosphere"
- Japanese Style: It has visual characteristics such as freshness, naturalness, emotionality, low saturation, and soft tones. -> Suitable for portraits, life records, travel photography, etc.
- Kodak Portra Style: With natural skin tones, warm tones, delicate transitions and rich layering, it can make digital photos look like "old-school but authentic" film. -> Suitable for those who pursue naturalness, warmth, storytelling, etc.
""

prompt_base_w_coordination = ""
Given an original image, its edited version, and the editing configuration parameters, analyze the visual changes and adjustments made. Infer the user's original editing goal and describe their vague intention in simple, everyday language as a direct statement from a non-technical user's perspective.

Special Notes:
1. Simulate a real client's voice to articulate service needs in casual business English. Write as if you're the actual decision-maker explaining requirements to a trusted partner, not just filling a template.
2. Focus on translating technical parameters (e.g., brightness +20, sharpening +15) into relatable visual outcomes without mentioning numerical values or technical terms.
3. Answer the output in a paragraph of 20 words or less.
4. If the user provides coordinates for a region of interest, ensure that ALL output identifies the category of the region of interest and includes the coordinates enclosed in <box></box> tags for annotation. The response should explicitly state the identified category and clearly mark the coordinates within the specified format.
5. The system should adaptively distribute inferred local adjustment intentions throughout the description text.
6. The user's editing intention for the OVERALL STYLE of the image needs to be given.
7. Express the inferred user's intention to edit the image as vaguely and richly as possible.
8. Must use highly concise words to summarize the overall style of the image in the user's editing intention, such as XXX style, etc.

Example of the user's original editing intention:
1. I want this to look like it's straight out of a movie—more depth, atmosphere, and that filmic texture. In the region <box>[0.4419, 0.1982, 0.8307, 0.9967]</box>, I wanted the person to look clearer and the outfit details to pop. Make the skin smoother and heighten the colors, especially the jacket and surrounded area.
2. In the region <box>[0.1242, 0.699, 0.7453, 1.0000]</box>, I hope my face to look brighter and my eyes to stand out more. Can you make the skin smoother and the hair darker? Let's enhance the entire image while making sure the colors don't vibrate too much.

Output requirements:
- It is necessary to provide TWO possible intentions of user to modify the image.
- Two possible user intentions are independent of each other. If the user provides coordinates for a region of interest, these two possible user intentions need to include the user's modification intentions for the region of interest.
- The following output format MUST be strictly followed:
{
"user want 1": "< The first intentions of user >",
"user want 2": "< The second intentions of user >"
}
Relative coordinates of the region of interest: <region_of_interest_coordinates>
The configuration details are as follows: <corresponding_lua_file>
""

prompt_base_wo_coordination = ""
Given an original image, its edited version, and the editing configuration parameters, analyze the visual changes and adjustments made. Infer the user's original editing goal and describe their vague intention in simple, everyday language as a direct statement from a non-technical user's perspective.

Special Notes:
1. Simulate a real client's voice to articulate service needs in casual business English. Write as if you're the actual decision-maker explaining requirements to a trusted partner, not just filling a template.
2. Focus on translating technical parameters (e.g., brightness +20, sharpening +15) into relatable visual outcomes without mentioning numerical values or technical terms.
3. Answer the output in a paragraph of 20 words or less.
4. The system should adaptively distribute inferred local adjustment intentions throughout the description text.
5. The user's editing intention for the OVERALL STYLE of the image needs to be given.
6. Express the inferred user's intention to edit the image as vaguely and richly as possible.
7. Must use highly concise words to summarize the overall style of the image in the user's editing intention, such as XXX style, etc.

Output requirements:
- It is necessary to provide TWO possible intentions of user to modify the image.
- Two possible user intentions are independent of each other.
- The following output format MUST be strictly followed:
{
"user want 1": "< The first intentions of user >",
"user want 2": "< The second intentions of user >"
}
The configuration details are as follows: <corresponding_lua_file>
""

### Image Type Processing:
Scene-level instruction generation: Style_expressions + prompt_base_wo_coordination
Region-level instruction generation: Style_expressions + prompt_base_w_coordination

```

Figure 23: Prompt for simulating the casual user instructions.

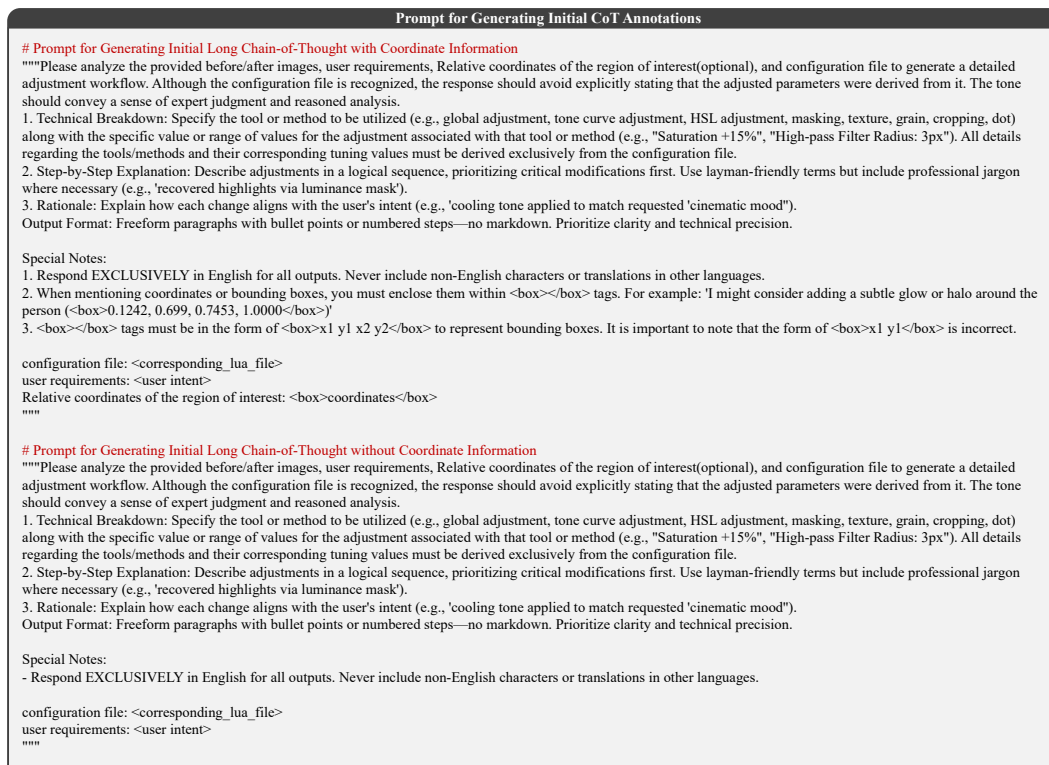


Figure 24: Prompt for generating the initial Chain-of-Thought (COT) annotations.

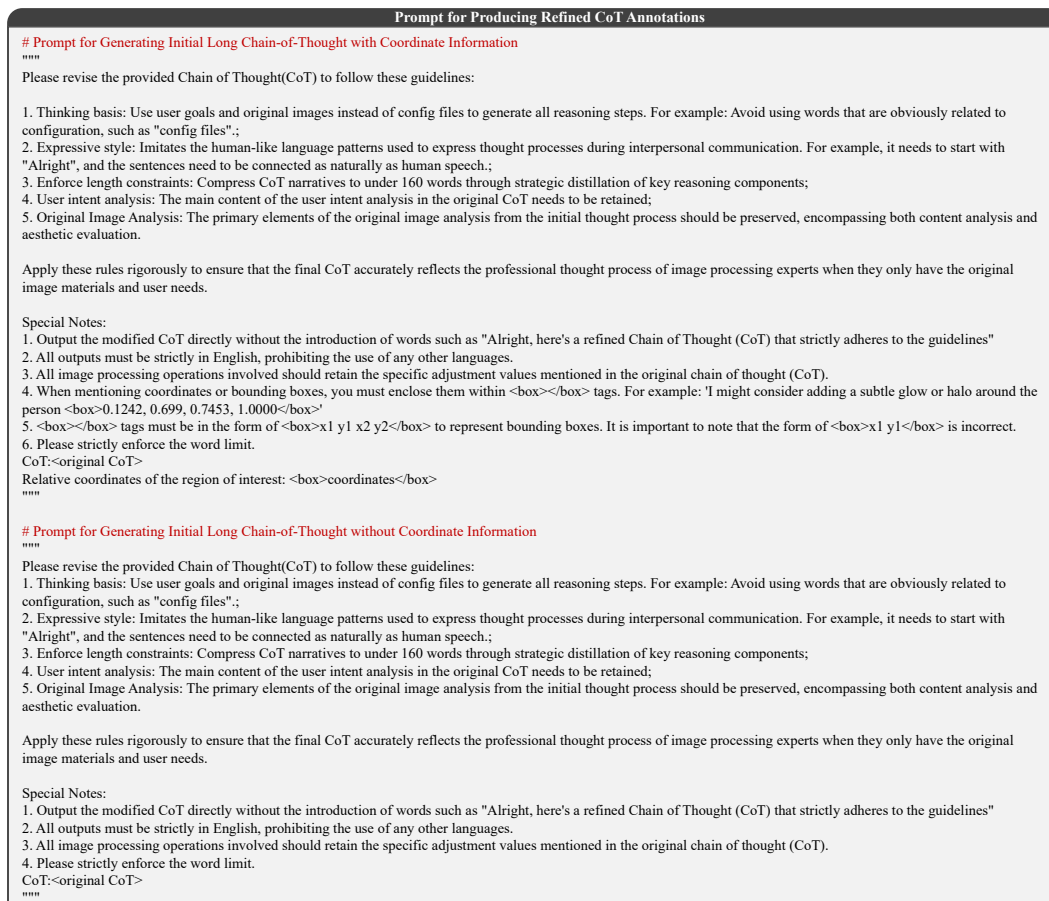


Figure 25: Prompt for generating the refined Chain-of-Thought (COT) annotations.

E Details of Retouching Tools in Lightroom

We provide an overview of key Lightroom tools generated by JarvisArt, focusing on the functionality of retouching tools and their associated parameters:

Table 7: Lightroom Tools with Functional Description and Parameter Type.

Tool Name	Functional Description	Type
Basic Adjustments		
WhiteBalance	Overall color temperature (As Shot, Auto, Custom)	Str.
Temperature	Blue-yellow balance (2000-10000 Kelvin)	Num.
Tint	Green-magenta balance (-150 to +150)	Num.
Exposure2012	Overall brightness (-5.0 to +5.0 stops)	Num.
Contrast2012	Difference between light/dark areas (-100 to +100)	Num.
Highlights2012	Adjusts bright areas (-100 to +100)	Num.
Shadows2012	Adjusts dark areas (-100 to +100)	Num.
Whites2012	Fine-tunes brightest parts (-100 to +100)	Num.
Blacks2012	Fine-tunes darkest parts (-100 to +100)	Num.
Texture	Enhances/smooths medium textures (-100 to +100)	Num.
Clarity2012	Enhances/reduces local mid-tone contrast (-100 to +100)	Num.
Dehaze	Reduces/adds atmospheric haze (-100 to +100)	Num.
Vibrance	Saturation of less-saturated colors (-100 to +100)	Num.
Saturation	Overall color intensity (-100 to +100)	Num.
IncrementalTemperature	Relative temperature adjustment (-100 to +100)	Num.
IncrementalTint	Relative tint adjustment (-100 to +100)	Num.
Tone Curve		
ToneCurveName2012	Predefined curve shape (Linear, Custom)	Str.
ToneCurvePV2012	Custom RGB tone curve points (x,y: 0-255)	Dict.
ToneCurvePV2012Red	Custom Red channel tone curve points	Dict.
ToneCurvePV2012Green	Custom Green channel tone curve points	Dict.
ToneCurvePV2012Blue	Custom Blue channel tone curve points	Dict.
ParametricShadows	Adjusts shadow tonal regions (-100 to +100)	Num.
ParametricDarks	Adjusts dark tonal regions (-100 to +100)	Num.
ParametricLights	Adjusts light tonal regions (-100 to +100)	Num.
ParametricHighlights	Adjusts highlight tonal regions (-100 to +100)	Num.
ParametricShadowSplit	Boundary: shadows/darks (10-50)	Num.
ParametricMidtoneSplit	Boundary: darks/lights (25-75)	Num.
ParametricHighlightSplit	Boundary: lights/highlights (50-90)	Num.
Detail		
Sharpness	Enhances edge definition (0-150)	Num.
SharpenRadius	Width of sharpening effect (0.5-3.0)	Num.
SharpenDetail	Amount of sharpening for details (0-100)	Num.
SharpenEdgeMasking	Masks sharpening to edges (0-100)	Num.
LuminanceSmoothing	Reduces luminance noise (0-100)	Num.
ColorNoiseReduction	Reduces color noise (0-100)	Num.
ColorNoiseReductionDetail	Fine-tunes color noise reduction (0-100)	Num.
ColorNoiseReductionSmoothness	Smoothness of color noise reduction (0-100)	Num.
HSL/Color (per color: Red, Orange, Yellow, Green, Aqua, Blue, Purple, Magenta)		
HueAdjustment<Color>	Shifts hue of specific color (-100 to +100)	Num.
SaturationAdjustment<Color>	Adjusts saturation of specific color (-100 to +100)	Num.
LuminanceAdjustment<Color>	Adjusts brightness of specific color (-100 to +100)	Num.
Color Grading		
SplitToningShadowHue	Hue for shadows in split toning (0-359)	Num.
SplitToningHighlightHue	Hue for highlights in split toning (0-359)	Num.
SplitToningShadowSaturation	Saturation for shadows (0-100)	Num.
SplitToningHighlightSaturation	Saturation for highlights (0-100)	Num.
SplitToningBalance	Balance between shadow/highlight toning (-100 to +100)	Num.
ColorGradeMidtoneHue	Midtone hue for color grading (0-359)	Num.
ColorGradeMidtoneSat	Midtone saturation for color grading (0-100)	Num.
ColorGradeMidtoneLum	Midtone luminance for color grading (0-100)	Num.

Continued on next page

Table 7: Lightroom tools with functional description and parameter type. (Continued)

Tool Name	Functional Description	Type
ColorGradeShadowLum	Luminance for shadows (0-100)	Num.
ColorGradeHighlightLum	Luminance for highlights (0-100)	Num.
ColorGradeBlending	Blending of color grading effect (0-100)	Num.
ColorGradeGlobalHue	Global hue adjustment (0-359)	Num.
ColorGradeGlobalSat	Global saturation adjustment (0-100)	Num.
ColorGradeGlobalLum	Global luminance adjustment (0-100)	Num.
Effects		
PostCropVignetteAmount	Darkens/lightens image corners (-100 to +100)	Num.
GrainAmount	Adds film grain effect (0-100)	Num.
ShadowTint	Adjusts color tint in shadows (-100 to +100)	Num.
Camera Calibration (for Red, Green, Blue primary channels)		
<PrimaryColor>Hue	Shifts primary color's hue (-100 to +100)	Num.
<PrimaryColor>Saturation	Adjusts primary color's saturation (-100 to +100)	Num.
Lens Blur (Overall: Dict.)		
LensBlur.Active	Enables/disables lens blur effect	Bool.
LensBlur.BlurAmount	Strength of blur effect (0-100)	Num.
LensBlur.FocalRange	Defines focal plane ("x1 y1 x2 y2")	Str.
LensBlur.BokehShape	Bokeh shape identifier (default 0)	Num.
LensBlur.BokehShapeDetail	Definition of bokeh shape edges (0-100)	Num.
LensBlur.HighlightsThreshold	Brightness threshold for bokeh (0-100)	Num.
LensBlur.HighlightsBoost	Enhances out-of-focus highlights (0-100)	Num.
LensBlur.CatEyeAmount	Simulates cat's eye bokeh effect (0-100)	Num.
LensBlur.CatEyeScale	Size of cat's eye effect (0-100)	Num.
Advanced Color Grading (PointColors - each point is a Dict.)		
SrcHue	Source hue for adjustment (0-6.28 rad)	Num.
SrcSat	Source saturation for adjustment (0-1.0)	Num.
SrcLum	Source luminance for adjustment (0-1.0)	Num.
HueShift	Hue shift amount (-1 to +1)	Num.
SatScale	Saturation scale (-1 to +1)	Num.
LumScale	Luminance scale (-1 to +1)	Num.
RangeAmount	Effect application amount (0-1.0)	Num.
HueRange	Falloff for hue adjustment (LowerNone, LowerFull, UpperFull, UpperNone: 0-1.0)	Dict.
SatRange	Falloff for saturation adjustment (sub-props same as HueRange)	Dict.
LumRange	Falloff for luminance adjustment (sub-props same as HueRange)	Dict.
Look (Overall: Dict.)		
Look.Name	Name of the look preset	Str.
Look.Amount	Intensity of the look effect (0.0-1.0)	Num.
Look.Parameters	Dictionary of specific adjustments applied by the look (e.g., ProcessVersion, ToneCurvePV2012, Parametric adjustments, SplitToning, Color-Grade, ConvertToGrayscale, LookTable, RGBTable, RG-BTableAmount)	Dict.
Localized Mask Adjustments (MaskGroupBasedCorrections - Array of Dicts.)		
<i>Per Correction Group:</i>		
CorrectionAmount	Amount for the correction group (0-1, default 1)	Num.
CorrectionActive	Activates the correction group	Bool.
CorrectionName	Name for the correction group	Str.
LocalExposure2012	Local exposure adjustment (-1 to +1)	Num.
LocalContrast2012	Local contrast adjustment (-1 to +1)	Num.
LocalHighlights2012	Local highlights adjustment (-1 to +1)	Num.
LocalShadows2012	Local shadows adjustment (-1 to +1)	Num.
LocalWhites2012	Local whites adjustment (-1 to +1)	Num.

Continued on next page

Table 7: Lightroom tools with functional description and parameter type. (Continued)

Tool Name	Functional Description	Type
LocalBlacks2012	Local blacks adjustment (-1 to +1)	Num.
LocalClarity / LocalClarity2012	Local clarity adjustment (-1 to +1)	Num.
LocalDehaze	Local dehaze adjustment (-1 to +1)	Num.
LocalTexture	Local texture adjustment (-1 to +1)	Num.
LocalHue	Local hue adjustment (-1 to +1)	Num.
LocalSaturation	Local saturation adjustment (-1 to +1)	Num.
LocalCurveRefineSaturation	Local saturation curve refinement (0-100)	Num.
LocalToningHue	Local toning hue (0-359)	Num.
LocalToningSaturation	Local toning saturation (-1 to +1)	Num.
LocalTemperature	Local temperature adjustment (-1 to +1)	Num.
LocalTint	Local tint adjustment (-1 to +1)	Num.
LocalLuminanceNoise	Local luminance noise reduction (-1 to +1)	Num.
LocalMoire	Local moire reduction (-1 to +1)	Num.
LocalDefringe	Local defringe adjustment (-1 to +1)	Num.
LocalGrain	Local grain adjustment (-1 to +1)	Num.
LocalSharpness	Local sharpness adjustment (-1 to +1)	Num.
<Channel>Curve	Local tone curve for Red, Green, Blue, or Main channels (points "x,y")	Dict.
LocalPointColors	Local specific color adjustments (dictionary of string-encoded points)	Dict.
CorrectionMasks	Array of mask definitions for the group	Array
<i>Per Mask in CorrectionMasks:</i>		
What	Mask type (e.g., "Mask/Image", "Mask/CircularGradient")	Str.
MaskActive	Activates this specific mask	Bool.
MaskName	Name of the mask (e.g., "Subject", "Sky")	Str.
MaskBlendMode	Mask blending (0=Add, 1=Intersect)	Num.
MaskInverted	Inverts the mask area	Bool.
MaskValue	Mask opacity (0.0-1.0)	Num.
MaskSubType	AI Mask subtype (Subject, Sky, Person etc.) / Object type	Num.
ReferencePoint	Center point for AI masks ("x y")	Str.
Gesture	Polygon points for object/region mask	Array
Top/Left/Bottom/Right	Coordinates for radial gradient (0-1)	Num.
Angle	Rotation angle for radial gradient (0-360)	Num.
Midpoint	Center point of radial gradient (0-100)	Num.
Feather	Edge feathering for radial gradient (0-100)	Num.
Flipped	Flips radial gradient direction	Bool.
MaskSubCategoryID	Category ID for person parts mask (Face, Eyes, etc.)	Num.