OPERA: An Agent for Image Restoration with End-to-End Planning–Execution Optimization

Feng Zhu[1*] **Shuyang Xie**[1*] **Yihan Zeng**[2] **Ming Liu**[1†] **Wangmeng Zuo**[1,3]

[1] Harbin Institute of Technology [2] Huawei Noah's Ark Lab
[3] Shenzhen Loop Area Institute

Abstract

Real-world image restoration is challenging due to complex and interacting mixed degradations. Recent agent-based approaches address this problem by composing multiple task-specific restoration tools. However, empirical analysis reveals that their performance is fundamentally limited by implicitly constrained planning spaces and the lack of coordination among independently pretrained tools. To address these issues, we propose **OPERA** (Optimized Planning-Execution Restoration Agent), a framework that optimizes both restoration planning and tool execution in an end-to-end manner. On the planning side, OPERA uses reinforcement learning to directly optimize tool composition over a combinatorial plan space, with the final restoration quality as the reward. On the execution side, OPERA introduces agent-guided co-training of restoration tools, enabling them to learn cooperative behaviors under sequential composition. Extensive experiments on multi-degradation benchmarks and real-world datasets demonstrate that OPERA consistently outperforms both all-in-one restoration models and existing agent-based methods across diverse and complex degradation scenarios. Code is available at `https://github.com/xsyshuishui/Opera`.

1 Introduction

Real-world images often suffer from complex degradations involving multiple distortion types simultaneously, where noise, blur, haze, rain, and compression artifacts can coexist and interact in non-trivial ways. Unlike the clean, single-degradation scenarios commonly studied in benchmarks, mixed degradations pose significant challenges for image restoration systems, as the combined effects of multiple distortions are difficult to model.

Early efforts address this problem by developing all-in-one restoration models that aim to handle multiple degradation types within a single network [27, 21, 25]. While conceptually appealing, these models often face a trade-off between generalization and specialization: accommodating diverse degradation patterns often results in overly smooth outputs and the loss of fine details, especially in complex real-world settings [4].

Recently, agent-based image restoration methods have emerged as a promising alternative [49, 4]. Instead of relying on a single model, these approaches leverage a collection of off-the-shelf, task-specific restoration tools to restore the image. These systems use large language models (LLMs) or vision-language models (VLMs) to dynamically select and compose appropriate tools for a given degraded input. By exploiting the strengths of specialized models, agentic systems have the potential to achieve strong task-specific performance while handling diverse combinations of degradations. This paradigm has opened a new direction for handling complex real-world degradations beyond the scope of conventional single-model designs.

*Equal contribution. †Corresponding author.

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

Despite this, existing agent-based methods suffer from two fundamental limitations that hinder their effectiveness in complex multi-degradation scenarios.

- **(1) Implicitly constrained planning space.** Existing agent-based image restoration methods rely on implicit planning assumptions that significantly restrict the space of restoration plans. For example, many methods assume a one-to-one mapping between degradation types and restoration tools, planning by explicitly matching each detected degradation to a specific tool [49, 4]. Some methods plan in a stepwise greedy manner, where each action is selected to locally improve an intermediate quality metric [48]. These assumptions substantially narrow the planning space, limiting the agent's ability to discover more complex yet effective cooperative restoration strategies.

- **(2) Static pretrained tools without coordination.** Existing agent-based frameworks treat restoration tools as fixed, independently pretrained modules [49, 4]. When these tools are composed sequentially, the output distribution of one tool becomes the input to the next, but these tools were never trained to cooperate. Applying a tool may alter the distribution of remaining degradations, adversely affecting subsequent restoration steps. Prior work has attempted to alleviate this issue by searching for optimal tool orderings, but we argue that ordering alone is insufficient. When tools lack coordinated training, ordering alone may be insufficient to address distribution shifts to yield satisfactory results across diverse degradation combinations.

To concretely understand how these limitations affect performance, we conduct an empirical study on cooperative multi-tool image restoration in a constrained setting. By exhaustively enumerating restoration plans, we analyze which tool combinations yield high-quality restoration outcomes. As illustrated in Figure 1, many high-performing plans violate common planning assumptions. For example, effective restoration often involves out-of-scope tools as well as repeated tool applications. This also demonstrates the limitations of existing restoration tools when applied to this cooperative setting. The findings suggest that both a more expressive planning space and improved coordination among tools are necessary for effective agent-based restoration.

Motivated by these observations, we propose **OPERA**, a framework that enables end-to-end optimization for both restoration planning and tool execution. On the *planning* side, OPERA departs from a hand-crafted, step-by-step decision-making workflow and instead trains an agent to generate a complete tool-invocation plan end-to-end. Given the combinatorial nature of the tool composition space, we use reinforcement learning to optimize the agent, with the final restoration quality serving as the reward signal. This formulation enables the agent to reason globally over tool compositions and discover non-obvious combinations. On the *execution* side, OPERA proposes agent-guided tool model training. Rather than treating restoration tools as static, independently pretrained modules, we jointly fine-tune tool models under the agent's generated plans. In this process, the agent acts as a high-level planner that induces diverse tool compositions, while individual tools remain architecturally independent and are updated solely based on their contribution to downstream restoration quality. This co-training strategy enables tools to learn cooperative behaviors, effectively mitigating the distribution shift caused by sequential composition.

Extensive experiments demonstrate that OPERA substantially outperforms both all-in-one restoration models and existing agent-based systems across most metrics. Further analysis shows that the agent learns non-trivial planning strategies, while jointly trained tools adapt to operate robustly within agent-generated restoration plans. Finally, OPERA generalizes well to real-world datasets, highlighting the robustness of our coordinated planning and execution framework.

Our main contributions are summarized as follows:

- We present an empirical study of cooperative multi-tool image restoration, providing key insights into the design of agentic image restoration systems.

- We propose **OPERA**, an end-to-end agent-based restoration framework that optimizes tool composition planning via reinforcement learning and enables cooperative behavior through agent-guided tool training.

- Extensive experiments on multi-degradation benchmarks and real-world datasets show that OPERA substantially outperforms existing all-in-one models and agent-based methods across most metrics.

2 Related Work

2.1 All-in-One Image Restoration

All-in-One Image Restoration (AiOIR) aims to develop unified models that handle diverse degradation types within a single framework [14]. Early approaches such as AirNet [21] learn degradation representations through contrastive learning. Recent advances explore various conditioning mechanisms: prompt-based methods [27] use learnable prompts for task adaptation, degradation embedding approaches [21, 25] explicitly encode degradation representations, and Mixture-of-Experts architectures [43] route inputs to specialized sub-networks. While these methods achieve strong performance on standard benchmarks, they may struggle with complex real-world degradation mixtures [24] and face trade-offs between generalization and task-specific accuracy [4]. Moreover, incorporating new degradation types often requires additional training, which can limit extensibility [49].

2.2 Agent-based Image Restoration

Agent-based methods address AiOIR limitations by employing intelligent controllers that dynamically select and sequence restoration tools. We review existing approaches along two dimensions:

Planning Strategies. Existing agent-based methods universally adopt step-by-step planning with iterative execution. RestoreAgent [4] first leverages multimodal LLMs for degradation identification, introducing iterative replanning with rollback mechanisms to correct suboptimal decisions. AgenticIR [49] systematizes the process into a five-stage perception-scheduling-execution-reflection-rescheduling pipeline, while Q-Agent [48] incorporates quality-driven Chain-of-Thought [38] reasoning to guide tool selection. To improve efficiency, SimpleCall [24] replaces heavy LLM inference with a lightweight policy network trained via PPO [29], achieving label-free learning but still following the iterative state-action-state paradigm. Despite their diversity, these methods share implicit assumptions that constrain the planning space. Degradation-matching approaches (RestoreAgent, AgenticIR) assume a one-to-one correspondence between detected degradations and applicable tools, while step-by-step methods (Q-Agent, SimpleCall) select tools based on immediate quality improvement. However, as we demonstrate in Section 3, high-performing restoration plans frequently violate these assumptions: effective restoration often involves out-of-scope tools and repeated tool applications that cannot be discovered through local optimization. In contrast, our agent generates complete tool plans in a single VLM call and is trained end-to-end to optimize final restoration quality directly.

Tool Utilization. Early RL-based work RL-Restore [42] assembles small CNN-based tools trained independently for specific degradations. Subsequent LLM-based approaches [4, 49, 48, 24] inherit this practice, relying on off-the-shelf restoration models without adaptation. 4KAgent [50] attempts to improve tool selection via Mixture-of-Experts routing, but its experts remain independently trained. Despite varied architectures, these methods universally treat tools as fixed modules. In contrast, our work jointly fine-tunes tools under agent-generated plans.

3 Cooperative Multi-Tool Image Restoration

As discussed in Section 1, recent agentic image restoration approaches typically rely on a collection of task-specific restoration tools and restore a degraded image by sequentially composing these tools. While this paradigm has been widely adopted, prior work often relies on implicit, largely unverified assumptions about how restoration tools should be composed. In this section, we first formalize the problem of cooperative image restoration with multiple tools. Then, through a controlled empirical study in a constrained setting where the action space can be exhaustively explored, we analyze the characteristics of high-performing restoration plans.

3.1 Problem Formulation

Assume we have collected a library of task-specific restoration tools $\mathcal{M}$, where each tool is trained to target a specific type of degradation. Given a degraded input image I_{LQ}, the goal is to design a restoration plan $\mathcal{P} = [M_1, M_2, \ldots, M_K]$, where each element $M_i \in \mathcal{M}$. By sequentially applying these tools to I_{LQ}, we obtain the restored image I_{pred}. The quality of a restoration plan is evaluated by image quality assessment metrics applied to I_{pred}.

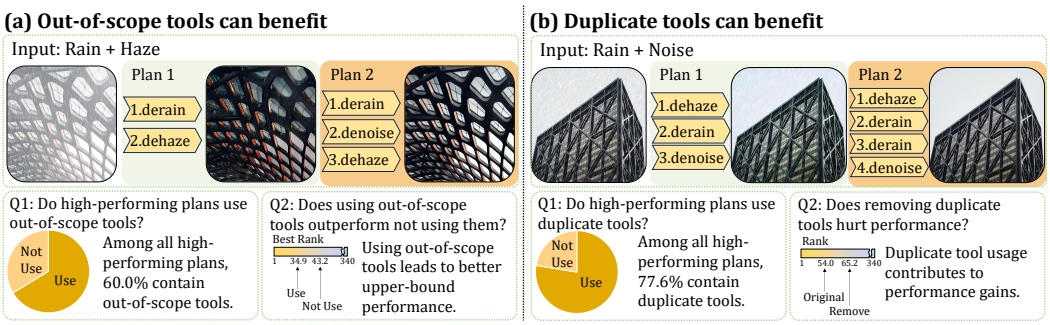


Figure 1: Empirical study of cooperative multi-tool image restoration. Zoom in for image details.

3.2 Empirical Analysis via Exhaustive Search

To gain insight into what constitutes an effective tool composition, we conduct controlled empirical experiments in which the space of restoration plans can be exhaustively enumerated. Our objective is to empirically analyze the characteristics of high-quality restoration plans in a multi-tool cooperative setting.

Experimental Settings. To make exhaustive search tractable, we consider a reduced setting with four common degradations: noise, rain, haze, and blur, each associated with a dedicated restoration tool. Restoration quality is evaluated using both full-reference metrics (PSNR, SSIM [37], LPIPS [46]) and no-reference metrics (CLIP-IQA [34], MUSIQ [18]), which capture complementary aspects of image fidelity and perceptual quality.

We randomly select 15 images from the MiOIR-Test [19] set as clean high-quality images, and synthesize low-quality inputs by applying 8 manually designed combinations of degradations, yielding 120 degraded images. For each degraded input, we enumerate all possible restoration plans with a maximum length of 4, where each step selects one of 4 available tools, yielding a total of 340 candidate plans per input. Details are shown in Appendix B.

Selection of High-Performing Plans. After evaluating all candidate plans, we rank them independently for each IQA metric. For a given metric, a plan is marked as *good* if it ranks within the top 10% among all plans. To identify plans that perform robustly across different quality criteria, we select those ranked *good* by at least 3 of the 5 metrics, with the additional requirement that both full-reference and no-reference metrics are included. This selection strategy mitigates metric-specific bias and emphasizes consistent performance across complementary evaluation signals. The resulting set of high-performing plans forms the basis of our subsequent analysis. On average, 12.05 out of the 340 plans per image are selected as high-performing. In addition, we compute an aggregated rank score for each plan by averaging its rank across all five metrics.

Finding 1: Out-of-Scope Tools Can Be Beneficial. Since the low-quality inputs are synthetically generated, the underlying degradation types are known for each image. Interestingly, despite this knowledge, we observe that a substantial portion of high-performing plans include restoration tools that do not correspond to any of the degradations present in the input. Specifically, 60.0% of the selected high-performing plans contain at least one out-of-scope tool. As illustrated in Figure 1 (a), in some cases, applying a denoising tool can improve restoration quality even when the input image is not corrupted by noise. To further quantify this effect, we compare, for each input image, the best aggregated rank achieved by plans that include out-of-scope tools with that of plans that strictly match the ground-truth degradation types. Plans incorporating out-of-scope tools outperform matched-only plans on 66.3% of the images, achieving a lower (better) mean best rank of 34.9 compared to 43.2.

Finding 2: Duplicate Tools Can be Beneficial. We further observe that repeated application of the same tool is common among high-performing plans: 77.6% of high-performing plans apply at least one tool multiple times. To assess whether such duplication meaningfully contributes to restoration quality, we compare each high-performing plan containing duplicate tools with its de-duplicated counterpart. As shown in Figure 1 (b), de-duplication degrades performance on average, increasing the average aggregated rank from 54.0 to 65.2. While a duplicate application is not universally beneficial, it can play a critical role in high-performing restoration plans.

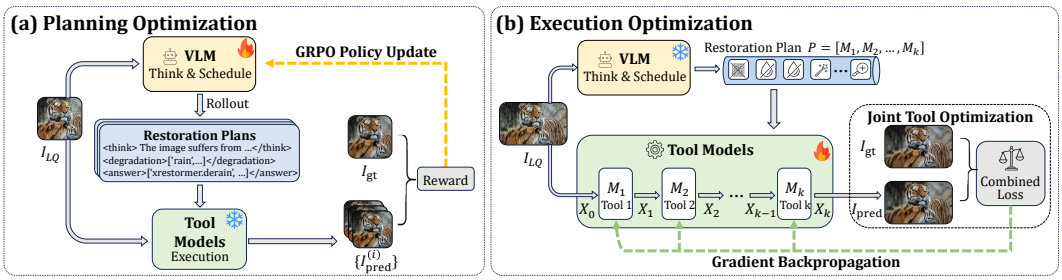


Figure 2: Overview of our OPERA framework. (a) **Planning Optimization**: The restoration agent is trained via Group Relative Policy Optimization (GRPO) to generate complete restoration plans end-to-end, receiving rewards based on final image quality. (b) **Execution Optimization**: At inference time, the agent generates a restoration plan that is executed by specialized tools. The tools are jointly optimized under agent guidance to cooperate effectively in multi-degradation scenarios.

Analysis. Together, these findings indicate that effective cooperative image restoration plans often deviate from intuitive heuristics such as matching tools to known degradations. Instead, complex tool compositions can outperform simpler, more constrained strategies. More broadly, our results highlight the limitations of existing single-degradation restoration tools when applied to cooperative settings. Models trained in isolation for specific degradations may require careful orchestration or iterative application. These observations motivate the co-evolution of the planning agent and the restoration tools.

4 Method

In this section, we present the proposed OPERA framework. We begin by describing the overall inference workflow in Section 4.1. We then detail the OPERA framework, including the planning optimization (Section 4.2) and the execution optimization (Section 4.3), as illustrated in Figure 2.

4.1 Overall Workflow

To approximate real-world image restoration scenarios, we consider a multi-degradation image restoration setting. Following prior works [4, 49, 50], we focus on eight commonly observed degradations: low resolution, noise, motion blur, defocus blur, rain, haze, JPEG compression artifacts, and low light. For each degradation, a set of specialized restoration tools is collected. Given a degraded input image, the objective of the restoration agent is to identify an optimal sequence of tool invocations that progressively restores the image toward a clean version.

The inference pipeline is illustrated in Figure 2. Given a degraded image, the planning agent generates a restoration plan in a single inference pass, specifying both the selected tools and their execution order. The restoration tools are then sequentially applied to the input image according to the plan, producing the final restored output.

4.2 Planning Optimization

As discussed in Section 1, existing agent-based image restoration methods often rely on carefully designed workflows or constrained planning strategies. While effective in specific cases, such designs inherently restrict the space of admissible restoration plans and limit the agent's ability to discover better tool compositions, as validated in Section 3. In this work, we remove such hand-crafted or step-by-step decision-making workflows, and instead enable the agent to plan freely over the full combinatorial space of tool sequences. To support this objective, the agent must possess both strong degradation perception and prior knowledge of restoration tool behaviors and their interactions. We therefore initialize the agent with a pretrained vision-language model exhibiting strong image quality assessment capabilities, providing a robust prior for degradation understanding. We further optimize this model to function as an effective restoration planner.

Importantly, for a given degraded image, there is typically no unique or well-defined "ground-truth" restoration plan, even among human experts, and the combinatorial space of possible tool compositions is prohibitively large. These characteristics make manually designed planning heuristics

impractical and exhaustive plan annotation costly, motivating reward-based optimization. Consequently, we adopt reinforcement learning (RL) to optimize the agent in an end-to-end manner. We choose group relative policy optimization (GRPO) [30] as the RL algorithm. By using the final restoration quality as the reward signal, the agent is encouraged to reason globally across tool compositions and autonomously discover effective, potentially non-obvious restoration strategies.

Notably, we adopt a single-shot inference strategy, in which the agent generates a complete restoration plan in a single inference pass conditioned on the degraded image, rather than relying on multi-turn iterative planning with intermediate tool execution. This design is motivated by both training efficiency and the characteristics of the task. Unlike tasks where step-by-step feedback is beneficial, restoration steps that appear locally optimal do not always align with the global objective. By generating the entire plan upfront, the agent is encouraged to perform global reasoning over tool interactions, thereby avoiding suboptimal decisions that may arise from sequential planning.

Reward Design. We adopt an end-to-end reward function to facilitate GRPO training. A restoration plan is considered effective if it leads to a high-quality restored image. Therefore, the primary reward is derived from Image Quality Assessment (IQA) metrics, serving as a proxy for the image fidelity and perceptual quality of the restored image. We additionally introduce auxiliary rewards for degradation prediction, structured output formatting, and reasoning–action consistency. The reward consists of four components:

- Restoration Reward $R_q \in \mathbb{R}^+$: This reward evaluates the quality of the restoration plan produced by the agent. Specifically, the restoration tools specified in the plan are applied to the input image, and the resulting image is assessed using image quality assessment (IQA) metrics. In practice, we adopt a composite objective combining full-reference and no-reference IQA metrics, as a balanced proxy for perceptual quality, following prior works [4, 48].

- Degradation Prediction Reward $R_d \in [0, 1]$: This reward measures whether the model correctly predicts the degradations present in the input image. It is computed as the standard $F1$ score between the predicted degradation set and the ground-truth set. Since degradations are synthetically applied during data generation, the ground-truth degradation labels are available. We introduce this auxiliary reward to facilitate RL exploration. Degradation prediction is simpler than plan generation, yet accurate degradation identification naturally informs the restoration strategy. By providing explicit supervision on this intermediate subtask, R_d guides the agent toward a better understanding of image conditions and enables more efficient discovery of effective restoration plans during GRPO training.

- Format Reward $R_f \in \{0, 1\}$: The format reward enforces a structured output that enables reliable parsing and tool execution. The model is instructed to think first, then predict a degradation set, and finally output a restoration plan.

- Consistency Reward $R_c \in \{0, 1\}$: Following prior works [47, 32], we add this reward that evaluates the alignment between the reasoning process and the final restoration plan, determined by an LLM judge. This reward encourages effective reasoning and prevents the model from collapsing into non-thinking behaviors.

The final reward is obtained by aggregating the four reward components: $R = R_q \times R_d \times R_f \times R_c$. The binary terms R_f and R_c act as gates, assigning zero reward to outputs that fail formatting or reasoning–action consistency checks. The continuous term R_d scales the restoration reward by degradation-prediction accuracy, favoring plans that combine accurate degradation identification with high restoration quality.

4.3 Execution Optimization

While the trained agent learns to generate effective restoration plans, the quality of the final output depends equally on the tools themselves. Off-the-shelf restoration tools are typically optimized for single-degradation scenarios in isolation. However, our analysis in Section 3 highlights the limitations of existing single-degradation restoration tools, and further optimization is needed for this cooperative setting. As illustrated in Figure 2 (b), we propose to jointly optimize the restoration tools under the agent's guidance.

Table 1: Quantitative comparison on Groups A, B, and C from AgenticIR [49]. We report full-reference metrics (PSNR↑, SSIM↑, LPIPS↓) and no-reference metrics (MANIQA↑, CLIP-IQA↑, MUSIQ↑). Groups A–C contain degradation combinations of varying complexity. "Ours (Planning)" denotes that only planning is optimized while using pretrained tools. "Ours (Full)" includes both planning and execution training. Best results are in **bold** and second best are underlined. Detailed results are shown in Appendix G.

Method	Group A						Group B						Group C					
	PSNR	SSIM	LPIPS↓	MANIQA	CLIP-IQA	MUSIQ	PSNR	SSIM	LPIPS↓	MANIQA	CLIP-IQA	MUSIQ	PSNR	SSIM	LPIPS↓	MANIQA	CLIP-IQA	MUSIQ
All-in-One Models																		
AirNet	19.13	0.60	0.43	0.26	0.39	42.46	19.31	0.66	0.37	0.29	0.43	47.88	17.95	0.51	0.58	0.19	0.31	30.12
PromptIR	20.06	0.61	0.41	0.26	0.40	42.62	20.47	0.67	0.34	0.29	0.43	48.10	18.51	0.52	0.58	0.19	0.31	29.71
MiOIR	20.84	0.66	0.37	0.25	0.39	47.82	20.56	0.69	0.32	0.26	0.43	51.87	15.63	0.49	0.54	0.17	0.29	37.95
DA-CLIP	19.58	0.60	0.43	0.24	0.41	42.51	18.56	0.59	0.44	0.24	0.42	43.70	18.53	0.53	0.53	0.19	0.35	33.87
InstructIR	18.03	0.58	0.44	0.27	0.35	45.77	18.34	0.62	0.41	0.30	0.38	50.94	17.09	0.51	0.56	0.17	0.25	33.69
AutoDIR	19.64	0.63	0.40	0.25	0.38	47.01	19.90	0.66	0.35	0.25	0.40	49.64	18.61	0.54	0.50	0.20	0.29	37.86
Agentic Systems																		
AgenticIR	21.04	0.68	0.31	0.31	0.45	56.88	20.55	0.70	0.31	0.32	0.46	57.57	18.82	0.55	0.45	0.27	0.39	48.68
MAIR	21.02	0.67	0.30	0.33	0.48	59.19	20.92	0.70	0.28	0.35	0.51	60.98	19.42	0.55	0.41	0.28	0.42	51.36
4KAgent	21.48	0.67	0.30	**0.37**	0.55	**63.19**	20.95	0.67	0.30	**0.37**	0.55	**62.69**	19.77	0.56	0.43	**0.35**	0.52	**55.56**
Ours (Planning)	22.32	0.71	0.33	0.35	0.45	59.34	22.28	0.75	0.31	0.36	0.46	59.10	20.75	0.61	0.47	0.28	0.37	49.91
Ours (Full)	**24.81**	**0.77**	**0.23**	0.32	**0.72**	57.84	**24.63**	**0.78**	**0.20**	0.33	**0.71**	58.41	**23.04**	**0.66**	**0.30**	0.25	**0.72**	51.27

End-to-End Tool Training. Given a restoration plan $\mathcal{P} = [M_1, M_2, \ldots, M_K]$ generated by the agent, the tools are applied sequentially:

$$X_0 = I_{\text{LQ}}, \quad X_k = M_k(X_{k-1}), \quad I_{\text{pred}} = X_K \tag{1}$$

The training loss is computed directly between the final output I_{pred} and the ground truth I_{gt}, with gradients propagating through the entire tool chain. This end-to-end formulation encourages tools to cooperate rather than optimize independently, as each tool's parameters are updated based on the quality of the final restored image rather than intermediate results.

Training Objective. We employ a composite loss that balances multiple objectives:

$$\mathcal{L} = w_{\text{pixel}}\mathcal{L}_{\text{L1}} + w_{\text{perc}}\mathcal{L}_{\text{VGG}} + w_{\text{lpips}}\mathcal{L}_{\text{LPIPS}} + w_{\text{nr}}(\mathcal{L}_{\text{MUSIQ}} + \mathcal{L}_{\text{CLIPIQA}}) \tag{2}$$

where pixel-level loss ($\mathcal{L}_{\text{L1}}$) ensures fidelity, perceptual losses ($\mathcal{L}_{\text{VGG}}$, $\mathcal{L}_{\text{LPIPS}}$) capture visual similarity, and no-reference quality losses ($\mathcal{L}_{\text{MUSIQ}}$, $\mathcal{L}_{\text{CLIPIQA}}$) promote realistic appearance. To ensure stable training, we employ a progressive loss schedule that transitions from pixel-level to perceptual optimization. Implementation details and complete loss formulations are provided in Appendix D.

5 Experiments

This section presents a comprehensive experimental evaluation of the proposed OPERA framework. We evaluate OPERA against state-of-the-art all-in-one restoration models and recent agent-based systems on multi-degradation benchmarks, reporting both quantitative and qualitative results. We analyze planning optimization and the learned planning behavior in Section 5.3, and evaluate the effectiveness of tool optimization in Section 5.4. We further evaluate generalization performance on real-world degraded images in Section 5.5. Finally, we analyze the efficiency of our framework in Section 5.6.

5.1 Experimental Setting

Training Data. To train both the agent and restoration tools, we require a dataset of multi-degradation images paired with high-quality ground truth. Following AgenticIR [49], we use MiOIR-Train [19] as the source of high-quality images and apply different combinations of degradations to synthesize low-quality inputs. We design 145 degradation combinations, with up to three degradation types per image. To reflect realistic scenarios, we only retain combinations that are likely to occur in practice. During training, we sample degradation combinations with a ratio of single:dual:triple=1:3:5, emphasizing complex multi-degradation scenarios. This process yields approximately 16,000 training images.

Restoration Tools. We construct a pool of 18 restoration tools, comprising 16 pretrained models based on Restormer [44], X-Restormer [6], and SwinIR [23], together with two low-light enhancement operations. These tools cover common degradation types, including noise, blur, rain, haze, low

resolution, and low light. Detailed tool specifications are provided in Appendix C. Note that the chosen tool set is a subset of those of baseline agentic methods.

Metrics. For quantitative evaluation, we use full-reference metrics (PSNR, SSIM [37], and LPIPS [46]) and no-reference metrics (MANIQA [41], CLIP-IQA [34], and MUSIQ [18]). PSNR and SSIM are computed on the Y channel in YCbCr, following 4KAgent [50]. Full-reference metrics measure pixel-level fidelity against ground truth, while no-reference metrics assess perceptual quality.

Implementation Details. For planning optimization, we adopt VisualQuality-R1 [39] as the base model, a finetuned variant of Qwen2.5-VL-7B-Instruct [2] equipped with basic image quality assessment capabilities. We adopt Pangu-Embedded-7B [3] as the LLM judge to calculate the consistency reward, due to its strong reasoning capability and high inference throughput. Prompts are detailed in Appendix M. We employ verl [31] for GRPO training with a batch size of 32 and a group size of $G = 8$. The restoration reward R_q is defined as a weighted sum of five IQA metrics: PSNR, SSIM, LPIPS, CLIP-IQA, and MUSIQ. For tool training, we use the Adam optimizer with a learning rate of 1×10^{-6}. The training loss combines L1, VGG perceptual loss, LPIPS, MUSIQ, and CLIP-IQA with weights 0.4, 0.1, 0.15, 0.1, 0.1, respectively. We employ a progressive schedule that transitions from pixel-level to perceptual optimization over the first 30% of training. All tool backbones (Restormer, X-Restormer, SwinIR) are fine-tuned with a low learning rate to adapt to the agent's calling patterns.

Benchmarks. We follow the evaluation protocol of AgenticIR [49] and 4KAgent [50], using the same benchmark images: Groups A, B, and C. The three test sets contain a total of 1,440 low-quality images synthesized from MiOIR-Test [19] using 16 combinations of two or three degradation types.

5.2 Main Results

Baselines. We compare our method against two categories of approaches: (1) all-in-one restoration models, including AirNet [21], PromptIR [27], MiOIR [19], DA-CLIP [25], InstructIR [8], and AutoDIR [16]; and (2) agent-based restoration systems, including AgenticIR [49], MAIR [15], and 4KAgent [50].

Quantitative Comparison. Table 1 reports quantitative results for two variants of our approach. "Ours (Planning)" corresponds to our variant in which only planning is optimized, while using fixed, pretrained tools. "Ours (Full)" is the full variant in which both planning and execution are optimized.

First, even without tool optimization, the planning-only variant achieves performance on par with or even surpassing existing agentic systems, demonstrating that end-to-end optimization over complete tool composition plans is more effective than heuristic search or greedy-based planning. This highlights the advantage of learning a global restoration strategy directly from the quality of the final image as feedback. When tool training is enabled, performance improves substantially across all degradation groups and most metrics. In Group C, the most challenging setting with complex degradation combinations, the full system achieves improvements of +2.29 dB in PSNR and -0.17 in LPIPS compared to the planning-only variant. This confirms that coordinated tool training is critical for robust restoration under complex degradation scenarios.

Qualitative Comparison. Figure 3 presents visual comparisons. Compared to 4KAgent, our method better preserves fine textures such as fur details while effectively removing degradation artifacts.

5.3 Effectiveness of Planning Agent Optimization

In this section, we evaluate whether the proposed training method in Section 4.2 effectively equips the model with strong planning capabilities for image restoration.

Training Dynamics. The training process exhibits stable and consistent dynamics, as shown in Figure 4. The degradation prediction reward R_d rapidly increases to 0.8 within the first 100 steps, while the restoration reward R_q also improves steadily over time. This suggests that the agent effectively explores the planning space and progressively discovers higher-quality restoration strategies.

Behavior Analysis. We observe that the agent acquires meaningful planning strategies through reinforcement learning, with behaviors that closely align with established domain knowledge. For example, it prioritizes denoising in 89.5% of noise-corrupted images and follows a derain→dehaze sequence in 75.1% of cases where both degradations co-occur, consistent with findings in [49]. The agent also develops more nuanced strategies. Notably, it learns selective repetition: deblurring is

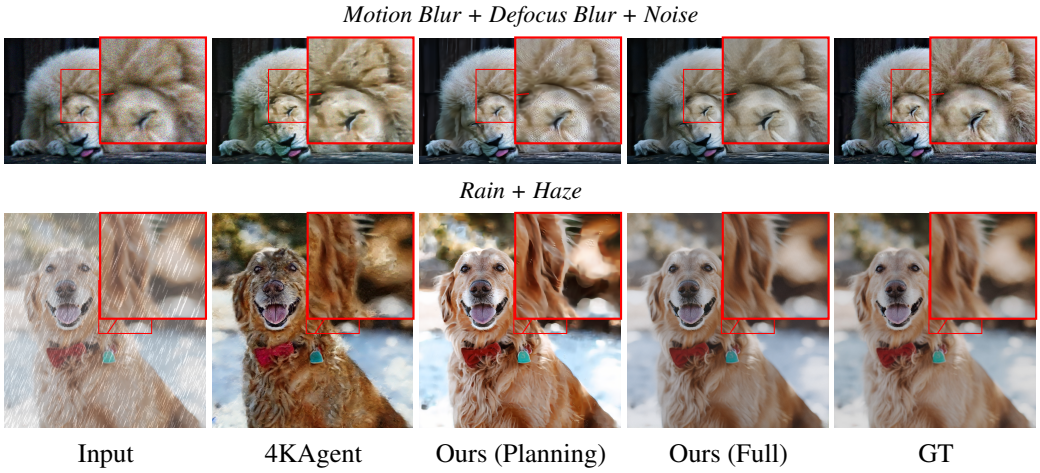


Figure 3: Qualitative comparison on benchmarks from AgenticIR [49].

Table 2: Effect of tool ordering and repetition with frozen pretrained tools over all test images in Groups A–C. Δ denotes the change relative to the original plans.

Plan	PSNR↑	SSIM↑	LPIPS↓	MANIQA↑	CLIP-IQA↑	MUSIQ↑
Original	21.87	0.70	0.36	0.34	0.43	56.65
Δ Order shuffling	−0.52	−0.021	+0.054	−0.031	−0.144	−5.16
Δ De-duplication	−0.32	−0.013	+0.049	−0.024	−0.198	−5.71

frequently applied iteratively (97.3%), while denoising is repeated less frequently, suggesting that the agent identifies which operations benefit from refinement. In addition, the agent tends to invoke more tools than the nominal number of degradations. For images with 1, 2, and 3 degradations, it applies an average of 2.8, 3.8, and 4.4 tools, respectively, in line with observations in Section 3.

Effect of Tool Ordering and Repetition. To further assess whether the learned ordering and repeated tool calls contribute to restoration quality, we perturb the plans generated by OPERA and compare them with the original plans, using the same frozen pretrained tools. We consider two interventions: (1) *order shuffling*, which randomly permutes the execution order while preserving the multiset of tool calls; and (2) *de-duplication*, which removes repeated calls to the same tool, retaining only the first occurrence of each tool and preserving the relative order of the remaining calls.

As shown in Table 2, both interventions degrade all six evaluation metrics. These results support the contribution of the learned execution order and repeated tool calls to restoration quality.

5.4 Effectiveness of Tool Optimization

As shown in Table 1, tool optimization substantially improves restoration quality over the planning-only variant, demonstrating the effectiveness of our execution optimization. A key motivation behind this approach is to enable independently pretrained restoration tools to cooperate effectively during sequential execution. We further investigate how these benefits relate to the planning strategies that determine tool composition. Specifically, we evaluate a heuristic planner and the OPERA planner with both pretrained and OPERA-trained tools. The heuristic planner uses ground-truth degradation labels and a fixed execution order. For each planner, we keep the generated plans unchanged across the two tool configurations.

As shown in Table 3, tool optimization increases average PSNR by 0.66 dB with the heuristic planner and by 2.40 dB with the OPERA planner, with larger improvements in SSIM and LPIPS under OPERA as well. These results show that tool optimization benefits different planning strategies, while its gains are substantially greater when tools are composed by the OPERA planner. This highlights the complementarity between planning and execution optimization, where adapting tools to the compositions in which they are used enables more effective multi-tool restoration.

Table 3: Interaction between planning and tool optimization over all test images in Groups A–C. "Trained" denotes OPERA-trained tools. The heuristic planner uses ground-truth degradation labels and a fixed execution order.

Planner	Tools	PSNR↑	SSIM↑	LPIPS↓	MANIQA↑	CLIP-IQA↑	MUSIQ↑
Heuristic	Pretrained	20.72	0.66	0.42	0.25	0.37	47.85
Heuristic	Trained	21.38	0.67	0.35	0.24	0.40	47.12
OPERA	Pretrained	21.87	0.70	0.36	0.34	0.43	56.65
OPERA	Trained	24.27	0.74	0.24	0.30	0.72	56.17

Table 4: Generalization on RTTS (real haze).

Category	Method	CLIP-IQA↑	MUSIQ↑
All-in-one	TransWeather [33]	0.292	46.27
	DA-CLIP [25]	0.325	53.23
	InstructIR [8]	0.370	54.46
	WResVLM [40]	0.371	56.09
	PromptIR [27]	0.372	53.88
Task-specific	KA-Net [9]	0.290	54.51
	DEA-Net [7]	0.370	54.09
	DeHamer [12]	0.370	53.79
	IPC-Dehaze [10]	<u>0.440</u>	**59.60**
Ours	Ours (Planning)	0.393	<u>56.91</u>
	Ours (Full)	**0.463**	55.78

Table 5: Generalization on LHP-Rain.

Method	PSNR↑	SSIM↑
SPANet [35]	28.00	0.8905
PReNet [28]	27.57	0.8595
MPRNet [45]	28.41	0.8807
GT-Rain [1]	28.62	0.8675
Uformer-B [36]	28.74	**0.9262**
NeRD-Rain [5]	30.83	0.8854
SCD-Former [13]	29.41	<u>0.9127</u>
FAD-Former [11]	**31.27**	0.8969
Ours (Planning)	25.85	0.8301
Ours (Full)	<u>30.93</u>	0.8995

5.5 Generalization to Real-World Data

Real-world image degradations are inherently complex, where images often exhibit multiple interacting degradations whose types and boundaries cannot be reliably identified. To evaluate generalization, we test on two widely-used real-world datasets RTTS [20] (haze) and LHP-Rain [13] (rain). For RTTS, we compare against both all-in-one models and task-specific dehazing methods. For LHP-Rain, we select baseline methods where the models have not seen the LHP training data, for a fair comparison.

As shown in Tables 4 and 5, OPERA generalizes effectively to real-world distributions. Current real-world benchmarks are mostly dominated by a single degradation (e.g., rain), which inherently favors task-specific methods over multi-degradation systems like ours. Effective restoration of these datasets typically does not require the coordination of multiple tools. This makes them less suitable for evaluating our method, which focuses on multi-tool cooperative restoration under mixed degradations. Despite this disadvantage, our method achieves competitive or superior performance.

5.6 Efficiency Analysis

Our framework requires only a single call to the VLM agent to generate a complete restoration plan during inference. Compared to baseline agent-based methods that rely on heuristic search or greedy exploration over the planning space, our approach avoids repeated trial-and-error executions. On average, the agent invokes 2.8, 3.8, and 4.4 tools for images containing 1, 2, and 3 degradations, respectively. These tool invocations constitute the only calls to restoration models during inference, resulting in predictable, controllable computational overhead in practice.

6 Conclusion

In this paper, we introduce OPERA, a unified framework that optimizes both agent planning and tool execution for image restoration. OPERA enables the agent to generate complete restoration plans via end-to-end RL and adapts tools to cooperate within agent-generated pipelines. Experiments show that end-to-end optimization of both planning and execution enables OPERA to overcome the limitations of existing agentic methods. Despite being trained exclusively on synthetic datasets, OPERA generalizes well to real-world benchmarks. We believe that the insights from OPERA will inform future agent-based image restoration systems, highlighting the importance of optimizing both planning and execution for effective cooperative image restoration.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 62501191. The authors declare no competing interests.

References

[1] Yunhao Ba, Howard Zhang, Ethan Yang, Akira Suzuki, Arnold Pfahnl, Chethan Chinder Chandrappa, Celso de Melo, Suya You, Stefano Soatto, Alex Wong, and Achuta Kadambi. Not just streaks: Towards ground truth for single image deraining. In *ECCV*, 2022.

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[3] Hanting Chen, Yasheng Wang, Kai Han, Dong Li, Lin Li, Zhenni Bi, Jinpeng Li, Haoyu Wang, Fei Mi, Mingjian Zhu, et al. Pangu embedded: An efficient dual-system llm reasoner with metacognition. *arXiv preprint arXiv:2505.22375*, 2025.

[4] Haoyu Chen, Wenbo Li, Jinjin Gu, Jingjing Ren, Sixiang Chen, Tian Ye, Renjing Pei, Kaiwen Zhou, Fenglong Song, and Lei Zhu. Restoreagent: Autonomous image restoration agent via multimodal large language models. *Advances in Neural Information Processing Systems*, 37:110643–110666, 2024.

[5] Xiang Chen, Jinshan Pan, and Jiangxin Dong. Bidirectional multi-scale implicit neural representations for image deraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25627–25636, 2024.

[6] Xiangyu Chen, Zheyuan Li, Yuandong Pu, Yihao Liu, Jiantao Zhou, Yu Qiao, and Chao Dong. A comparative study of image restoration networks for general backbone network design. In *European Conference on Computer Vision*, pages 74–91. Springer, 2024.

[7] Zixuan Chen, Zewei He, and Zhe-Ming Lu. Dea-net: Single image dehazing based on detail-enhanced convolution and content-guided attention. *IEEE transactions on image processing*, 33:1002–1015, 2024.

[8] Marcos V Conde, Gregor Geigle, and Radu Timofte. Instructir: High-quality image restoration following human instructions. In *European Conference on Computer Vision*, pages 1–21. Springer, 2024.

[9] Yuxin Feng, Long Ma, Xiaozhe Meng, Fan Zhou, Risheng Liu, and Zhuo Su. Advancing real-world image dehazing: Perspective, modules, and training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9303–9320, 2024.

[10] Jiayi Fu, Siyu Liu, Zikun Liu, Chun-Le Guo, Hyunhee Park, Ruiqi Wu, Guoqing Wang, and Chongyi Li. Iterative predictor-critic code decoding for real-world image dehazing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12700–12709, 2025.

[11] Ning Gao, Xingyu Jiang, Xiuhui Zhang, and Yue Deng. Efficient frequency-domain image deraining with contrastive regularization. In *European conference on computer vision*, pages 240–257. Springer, 2024.

[12] Chun-Le Guo, Qixin Yan, Saeed Anwar, Runmin Cong, Wenqi Ren, and Chongyi Li. Image dehazing transformer with transmission-aware 3d position embedding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5812–5820, 2022.

[13] Yun Guo, Xueyao Xiao, Yi Chang, Shumin Deng, and Luxin Yan. From sky to the ground: A large-scale benchmark and simple baseline towards real rain removal. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12097–12107, October 2023.

[14] Junjun Jiang, Zengyuan Zuo, Gang Wu, Kui Jiang, and Xianming Liu. A survey on all-in-one image restoration: Taxonomy, evaluation and future trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[15] Xu Jiang, Gehui Li, Bin Chen, and Jian Zhang. Multi-agent image restoration. *arXiv preprint arXiv:2503.09403*, 2025.

[16] Yitong Jiang, Zhaoyang Zhang, Tianfan Xue, and Jinwei Gu. Autodir: Automatic all-in-one image restoration with latent diffusion. In *European Conference on Computer Vision*, pages 340–359. Springer, 2024.

[17] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[18] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5148–5157, 2021.

[19] Xiangtao Kong, Chao Dong, and Lei Zhang. Towards effective multiple-in-one image restoration: A sequential and prompt learning strategy. *arXiv preprint arXiv:2401.03379*, 2024.

[20] Boyi Li, Wenqi Ren, Dengpan Fu, Dacheng Tao, Dan Feng, Wenjun Zeng, and Zhangyang Wang. Benchmarking single-image dehazing and beyond. *IEEE transactions on image processing*, 28(1):492–505, 2018.

[21] Boyun Li, Xiao Liu, Peng Hu, Zhongqin Wu, Jiancheng Lv, and Xi Peng. All-in-one image restoration for unknown corruption. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17452–17462, 2022.

[22] Hao Li, Xiang Chen, Jiangxin Dong, Jinhui Tang, and Jinshan Pan. FoundIR: Unleashing million-scale training data to advance foundation models for image restoration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

[23] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *IEEE/CVF International Conference on Computer Vision Workshops, ICCVW 2021, Montreal, QC, Canada, October 11-17, 2021*, pages 1833–1844. IEEE, 2021.

[24] Jianglin Lu, Yuanwei Wu, Ziyi Zhao, Hongcheng Wang, Felix Jimenez, Abrar Majeedi, and Yun Fu. Simplecall: A lightweight image restoration agent in label-free environments with mllm perceptual feedback. *arXiv preprint arXiv:2512.18599*, 2025.

[25] Ziwei Luo, Fredrik K Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B Schön. Controlling vision-language models for multi-task image restoration. In *ICLR*, 2024.

[26] OpenAI. Introducing GPT-5.4. `https://openai.com/index/introducing-gpt-5-4/`, March 2026.

[27] Vaishnav Potlapalli, Syed Waqas Zamir, Salman H Khan, and Fahad Shahbaz Khan. Promptir: Prompting for all-in-one image restoration. *Advances in Neural Information Processing Systems*, 36:71275–71293, 2023.

[28] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: A better and simpler baseline. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3937–3946, 2019.

[29] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[30] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024.

[31] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

[32] Kwai Keye Team, Biao Yang, Bin Wen, Changyi Liu, Chenglong Chu, Chengru Song, Chongling Rao, Chuan Yi, Da Li, Dunju Zang, et al. Kwai keye-vl technical report. *arXiv preprint arXiv:2507.01949*, 2025.

[33] Jeya Maria Jose Valanarasu, Rajeev Yasarla, and Vishal M Patel. Transweather: Transformer-based restoration of images degraded by adverse weather conditions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2353–2363, 2022.

[34] Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 2555–2563, 2023.

[35] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson WH Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12270–12279, 2019.

[36] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17683–17693, 2022.

[37] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[38] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[39] Tianhe Wu, Jian Zou, Jie Liang, Lei Zhang, and Kede Ma. VisualQuality-R1: Reasoning-induced image quality assessment via reinforcement learning to rank. *arXiv preprint arXiv:2505.14460*, 2025.

[40] Jiaqi Xu, Mengyang Wu, Xiaowei Hu, Chi-Wing Fu, Qi Dou, and Pheng-Ann Heng. Towards real-world adverse weather image restoration: Enhancing clearness and semantics with vision-language models. In *European Conference on Computer Vision*, pages 147–164. Springer, 2024.

[41] Sidi Yang, Tianhe Wu, Shuwei Shi, Shanshan Lao, Yuan Gong, Mingdeng Cao, Jiahao Wang, and Yujiu Yang. MANIQA: multi-dimension attention network for no-reference image quality assessment. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2022, New Orleans, LA, USA, June 19-20, 2022*, pages 1190–1199. IEEE, 2022.

[42] Ke Yu, Chao Dong, Liang Lin, and Chen Change Loy. Crafting a toolchain for image restoration by deep reinforcement learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2443–2452, 2018.

[43] Eduard Zamfir, Zongwei Wu, Nancy Mehta, Yuedong Tan, Danda Pani Paudel, Yulun Zhang, and Radu Timofte. Complexity experts are task-discriminative learners for any image restoration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 12753–12763. Computer Vision Foundation / IEEE, 2025.

[44] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5728–5739, 2022.

[45] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14821–14831, 2021.

[46] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[47] Yi-Fan Zhang, Xingyu Lu, Xiao Hu, Chaoyou Fu, Bin Wen, Tianke Zhang, Changyi Liu, Kaiyu Jiang, Kaibing Chen, Kaiyu Tang, et al. R1-reward: Training multimodal reward model through stable reinforcement learning. *arXiv preprint arXiv:2505.02835*, 2025.

[48] Yingjie Zhou, Jiezhang Cao, Zicheng Zhang, Farong Wen, Yanwei Jiang, Jun Jia, Xiaohong Liu, Xiongkuo Min, and Guangtao Zhai. Q-agent: Quality-driven chain-of-thought image restoration agent through robust multimodal large language model. *arXiv preprint arXiv:2504.07148*, 2025.

[49] Kaiwen Zhu, Jinjin Gu, Zhiyuan You, Yu Qiao, and Chao Dong. An intelligent agentic system for complex image restoration problems. In *The Thirteenth International Conference on Learning Representations*, 2025.

[50] Yushen Zuo, Qi Zheng, Mingyang Wu, Xinrui Jiang, Renjie Li, Jian Wang, Yide Zhang, Gengchen Mai, Lihong V. Wang, James Y. Zou, Xiaoyu Wang, Ming-Hsuan Yang, and Zhengzhong Tu. 4kagent: Agentic any image to 4k super-resolution. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025.

A Appendix Overview

Appendix B details the experimental settings for the empirical studies in Section 3.

Appendix C lists OPERA's restoration tools and the degradation types they address.

Appendix D provides implementation details for the tool training procedure introduced in Section 4.3.

Appendix E analyzes the computational cost of tool training.

Appendix F compares OPERA's trained planning agent with GPT-5.4.

Appendix G reports quantitative results for individual degradation combinations and additional evaluations on real-world images.

Appendix H presents additional ablation studies of planning optimization, degradation sampling, tool training, and reward design.

Appendix I examines how tool training affects the behavior of individual restoration tools.

Appendix J describes the GRPO algorithm used to train the planning agent.

Appendix K discusses OPERA's limitations and societal impacts.

Appendix L presents additional qualitative comparisons.

Appendix M provides the prompt templates used in our framework.

B Empirical Study of Cooperative Multi-Tool Image Restoration

This section provides implementation details of the empirical study experiment.

B.1 Restoration Tools

In this controlled setting, we only consider 4 common degradations, each of which corresponds to one tool, including:

- noise: Restormer [44] (Trained with $\sigma = 50$)
- rain: Restormer [44]
- haze: Restormer [44]
- blur: X-Restormer [6]

B.2 Degradation Combinations

The input degraded images are synthesized by adding the following eight different degradation combinations:

- rain, noise
- rain, haze
- haze, noise
- rain, blur
- haze, blur
- blur, noise
- rain, haze, noise
- rain, haze, blur

C Image Restoration Tools

We collect 18 common image restoration models to form the tool pool $\mathcal{M}$. The detailed tool list is shown in Table 7.

Table 6: Comparison of tool sets used by different agentic systems. We use a significantly small set of tools, which is a subset of baseline systems.

Tool	AgenticIR	MAIR	4KAgent	Ours
Restormer	✓	✓	✓	✓
X-Restormer	✓	✓	✓	✓
SwinIR	✓	✓	✓	✓
FBCNN	✓	✓	✓	
DiffBIR	✓	✓	✓	
DRBNet		✓	✓	
DehazeFormer	✓	✓	✓	
RIDCP	✓	✓	✓	
MPRNet	✓	✓	✓	
MAXIM		✓	✓	
HAT	✓	✓	✓	
RetinexFormer		✓		
DWGAN		✓		
CoTF		✓		
IFAN	✓		✓	
CLAHE	✓		✓	
NAFNet			✓	
ConvIR			✓	
LaKDNet			✓	
EVSSM			✓	
DiffPlugin			✓	
FourierDiff			✓	
GFPGAN			✓	
CodeFormer			✓	

We further compare our tool pool with those adopted by baseline agentic methods in Table 6. Notably, our method relies on a substantially smaller set of tools, which is a strict subset of the tools used by the baseline systems. This observation highlights that the performance improvements stem from the planning capability learned by the planning agent, rather than from access to a larger or more powerful set of tools.

D Tool Training Implementation Details

This section provides implementation details for the joint tool optimization framework described in Section 4.3. We detail the progressive loss schedule (Section D.2) and loss function formulations (Section D.3).

D.1 Training Configuration

We jointly finetune 16 restoration models from three architectures (Restormer, X-Restormer, SwinIR) covering denoising, deblurring, deraining, dehazing, and super-resolution tasks. Training uses approximately 16,000 synthetic degraded images with an 80/20 train-validation split. We train for 23 epochs with a batch size of 2 and select the checkpoint at epoch 10 based on validation performance.

Optimization. We use the Adam optimizer with learning rate 1×10^{-6}. Gradient clipping with max norm 0.5 is applied uniformly across all models in each pipeline.

D.2 Progressive Loss Schedule

During tool training, we employ a progressive loss schedule that transitions from pixel-level to perceptual optimization using cosine annealing. This helps stabilize early training when cascaded tools may produce poorly aligned outputs.

Table 7: Image restoration models used for different degradation types.

Degradation Type	**Model**
Noise	Restormer [44] (trained with $\sigma = 15$) Restormer [44] (trained with $\sigma = 25$) Restormer [44] (trained with $\sigma = 50$) X-Restormer [6] SwinIR [23] (trained with $\sigma = 15$) SwinIR [23] (trained with $\sigma = 25$) SwinIR [23] (trained with $\sigma = 50$)
Rain	Restormer [44] for deraining X-Restormer [6] for deraining
Dehazing	X-Restormer [6] for dehazing
Defocus blur	Restormer [44] X-Restormer [6]
Motion blur	Restormer [44]
Low-resolution	X-Restormer [6] SwinIR [23]
JPEG Compression Artifact	SwinIR [23]
Low Light	Constant Shift Gamma Correction

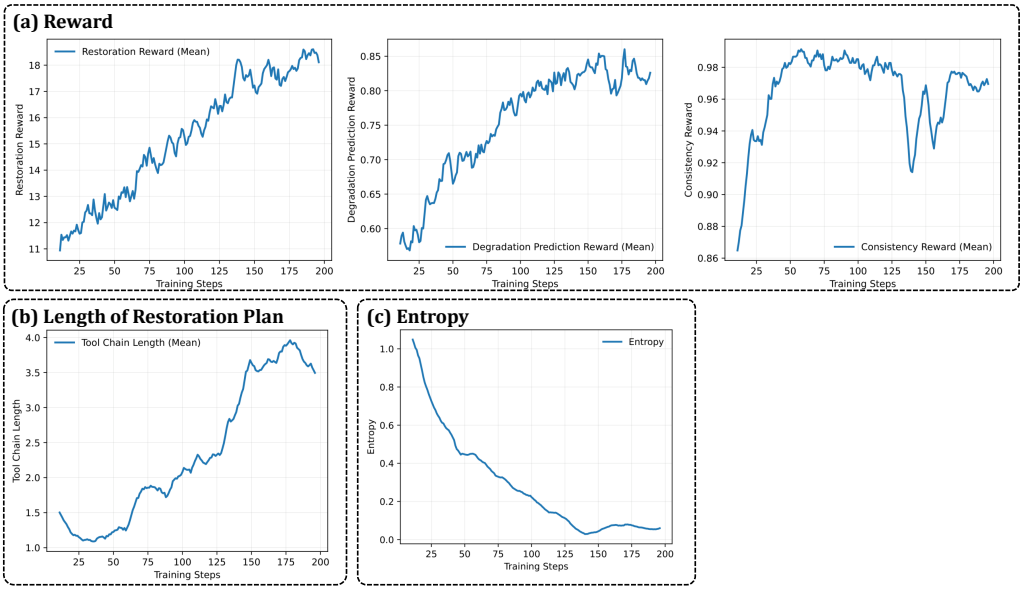


Figure 4: GRPO training dynamics during planning optimization.

Let e denote the current epoch, E the total epochs, and $T = \lfloor 0.3 \cdot E \rfloor$ the transition period. The annealing factor is:

$$\gamma(e) = \frac{1}{2}\left(1 - \cos\left(\pi \cdot \frac{e}{T}\right)\right), \quad e \in [1, T] \tag{3}$$

Loss weights evolve as follows during the transition period:

$$w_{\text{L1}}(e) = 1.0 - (1.0 - w_{\text{L1}}^{\text{target}}) \cdot \gamma(e) \tag{4}$$

$$w_*(e) = w_*^{\text{target}} \cdot \gamma(e), \quad * \in \{\text{VGG}, \text{LPIPS}, \text{MUSIQ}, \text{CLIPIQA}\} \tag{5}$$

After the transition period ($e > T$), all weights remain at their target values. Target weights are: $w_{\text{L1}}^{\text{target}} = 0.4$, $w_{\text{VGG}}^{\text{target}} = 0.1$, $w_{\text{LPIPS}}^{\text{target}} = 0.15$, $w_{\text{MUSIQ}}^{\text{target}} = 0.1$, $w_{\text{CLIPIQA}}^{\text{target}} = 0.1$.

D.3 Loss Function Details

The total tool training loss combines five components:

$$\mathcal{L} = w_{\text{L1}}\mathcal{L}_{\text{L1}} + w_{\text{VGG}}\mathcal{L}_{\text{VGG}} + w_{\text{LPIPS}}\mathcal{L}_{\text{LPIPS}} + w_{\text{MUSIQ}}\mathcal{L}_{\text{MUSIQ}} + w_{\text{CLIPIQA}}\mathcal{L}_{\text{CLIPIQA}} \tag{6}$$

L1 Loss ensures pixel-level fidelity:

$$\mathcal{L}_{\text{L1}} = \frac{1}{N}\sum_{i=1}^{N} |I_{\text{pred}}^{(i)} - I_{\text{gt}}^{(i)}| \tag{7}$$

VGG Perceptual Loss [17] captures mid-level feature similarity using VGG19:

$$\mathcal{L}_{\text{VGG}} = \sum_{l \in \mathcal{S}} \frac{1}{C_l H_l W_l} \|\phi_l(I_{\text{pred}}) - \phi_l(I_{\text{gt}})\|_1 \tag{8}$$

where ϕ_l denotes the l-th layer features and $\mathcal{S} = \{\text{relu2_2}, \text{relu3_4}, \text{relu4_4}\}$.

LPIPS Loss uses learned perceptual similarity:

$$\mathcal{L}_{\text{LPIPS}} = \sum_{l} \|w_l \odot (\hat{\phi}_l(I_{\text{pred}}) - \hat{\phi}_l(I_{\text{gt}}))\|_2^2 \tag{9}$$

where w_l are learned channel weights and $\hat{\phi}_l$ are normalized features.

MUSIQ Loss maximizes multi-scale quality score:

$$\mathcal{L}_{\text{MUSIQ}} = 1 - \frac{Q_{\text{MUSIQ}}(I_{\text{pred}})}{100} \tag{10}$$

CLIP-IQA Loss maximizes CLIP-based quality assessment:

$$\mathcal{L}_{\text{CLIPIQA}} = 1 - Q_{\text{CLIPIQA}}(I_{\text{pred}}) \tag{11}$$

The no-reference losses (MUSIQ, CLIP-IQA) encourage realistic appearance without requiring pixel-perfect alignment with ground truth, which is particularly beneficial when multiple valid restoration solutions exist.

E Details of Tool Execution Optimization Training Cost

Since the forward and backward passes of execution optimization must traverse the entire active tool chain, we analyze the training cost of tool optimization in this section.

GPU memory usage is primarily determined by the chain length, batch size, and input resolution. We conduct training on 2× NVIDIA H20 GPUs (96 GiB each) using PyTorch DDP, with a per-GPU batch size of 2 and input patches of size 128×128. The full training process takes approximately two days. In total, 16 tool models are loaded (each with ∼26M parameters, corresponding to ∼100 MB per checkpoint).

We log per-iteration GPU memory statistics over 15,720 iterations, summarized in Table 8.

As shown in Table 8, the chain length during training ranges from 1 to 10, with the majority (62%) falling between 4 and 6. GPU memory usage scales approximately linearly with chain length, increasing by around 5–6 GiB per additional tool. Given a 96 GiB GPU memory budget, we estimate the maximum feasible chain length to be approximately 12–14 without further optimization.

In practice, this limit is sufficient. The agent invokes 3–5 tools on average, and even in the most complex cases, the maximum observed chain length is 10, which remains well within the available memory budget.

Table 8: Per-iteration GPU memory usage under different tool chain lengths.

Chain Length	Proportion	Forward Memory (GiB)	Peak Reserved (GiB)
1–2	4.2%	7.4	18
3–4	32.4%	16.7	26
5	35.7%	26.2	34
6–7	26.5%	33.5	41
8–10	1.3%	42.8	56

Table 9: Quantitative comparison of the planning ability of OPERA and GPT-5.4.

Method	Group A						Group B						Group C					
	PSNR	SSIM	LPIPS↓	MANIQA	CLIP-IQA	MUSIQ	PSNR	SSIM	LPIPS↓	MANIQA	CLIP-IQA	MUSIQ	PSNR	SSIM	LPIPS↓	MANIQA	CLIP-IQA	MUSIQ
Ours (Planning)	**22.32**	**0.71**	**0.33**	**0.35**	**0.45**	**59.34**	**22.28**	**0.75**	**0.31**	**0.36**	**0.46**	**59.10**	**20.75**	**0.61**	**0.47**	**0.28**	**0.37**	**49.91**
GPT-5.4	21.46	0.69	0.37	0.28	0.41	50.05	20.93	0.72	0.34	0.31	0.43	52.49	20.17	0.60	0.53	0.20	0.33	36.22

F Comparison of Planning Agent with GPT-5.4

We compare our trained agent against GPT-5.4 [26], which contains strong reasoning and visual understanding capabilities. For a fair comparison, both models are provided with the same tool set and identical prompts. The tools are instantiated by pretrained weights without execution optimization. As shown in Table 9, GPT-5.4 demonstrates competitive zero-shot performance. However, our trained agent consistently outperforms GPT-5.4 across all settings and metrics, despite its significantly smaller model size. This demonstrates that while strong proprietary VLMs provide a solid zero-shot baseline, task-specific training remains crucial for achieving robust and high-quality restoration planning, validating the effectiveness of our planning optimization.

G Detailed Quantitative Results

Table 10 provides a detailed breakdown of our full model's performance on each degradation combination in the benchmark used by AgenticIR [49]. This complements the group-level averages reported in the main text (Table 1).

G.1 Additional Evaluation on Real-World Images

We additionally evaluate on 42 images from GT-RAIN [1] and 110 images from FoundIR [22], with 10 images sampled from each of 11 mixed-degradation subsets of FoundIR. We compare OPERA with its trained planner and tools against 4KAgent [50], using the latter's official implementation. As shown in Table 11, OPERA improves PSNR by 5.28 and 3.09 dB on the two evaluated subsets, respectively, while results on the other metrics vary.

H Ablation Studies

In this section, we present additional ablation studies of OPERA. The planning agent used for comparison is trained with 16 restoration tools.

H.1 Contribution of Planning Optimization

To assess the contribution of OPERA's planning optimization beyond its initialization, we compare VisualQuality-R1 [39] with Ours (Planning), using the same prompt and frozen pretrained restoration tools. As shown in Table 12, planning optimization improves PSNR from 20.17 to 21.87 dB and improves the other five evaluation metrics. These results demonstrate that OPERA's GRPO training improves restoration planning beyond the capabilities provided by VisualQuality-R1.

H.2 Sampling Ratio and Execution Optimization Ablation

We conduct a 2×2 factorial ablation analyzing two factors: (1) degradation sampling ratio (1:1:1 balanced vs. 1:3:5 emphasizing multi-degradation), and (2) training scope (planning-only vs. full planning-execution training). Table 13 presents results on the MiOIR-Test benchmark.

Table 10: Per-degradation performance breakdown of our full model (Ours Full) on Groups A, B, and C from AgenticIR. Results are grouped by degradation complexity (Group A: 2 degradations, Group B: 2 degradations, Group C: 3 degradations).

Group	Degradation Combination	PSNR↑	SSIM↑	LPIPS↓	CLIP-IQA↑	MUSIQ↑	MANIQA↑
A	Rain + Haze	25.25	0.94	0.06	0.84	69.07	0.44
	Motion Blur + Low Resolution	25.78	0.72	0.18	0.84	58.78	0.28
	Low Light + Noise	23.93	0.81	0.19	0.83	64.70	0.35
	Defocus Blur + JPEG	25.75	0.68	0.43	0.34	30.69	0.19
	Noise + JPEG	26.23	0.65	0.44	0.43	46.20	0.29
	Rain + Low Resolution	25.99	0.72	0.18	0.85	66.10	0.34
	Motion Blur + Low Light	22.57	0.78	0.20	0.81	60.27	0.32
	Defocus Blur + Haze	22.98	0.82	0.14	0.86	66.88	0.34
B	Haze + Noise	22.82	0.81	0.16	0.88	66.96	0.36
	Defocus Blur + Low Resolution	26.74	0.73	0.17	0.86	62.85	0.31
	Motion Blur + JPEG	23.88	0.68	0.36	0.30	35.37	0.20
	Rain + Low Light	25.06	0.90	0.11	0.79	68.44	0.44
C	Haze + Motion Blur + Low Resolution	21.71	0.69	0.21	0.83	56.61	0.27
	Rain + Noise + Low Resolution	25.56	0.70	0.22	0.85	63.62	0.31
	Low Light + Defocus Blur + JPEG	21.44	0.63	0.50	0.32	27.55	0.16
	Motion Blur + Defocus Blur + Noise	23.46	0.61	0.27	0.87	57.30	0.27
Group A Average		24.81	0.77	0.23	0.72	57.84	0.32
Group B Average		24.63	0.78	0.20	0.71	58.41	0.33
Group C Average		23.04	0.66	0.30	0.72	51.27	0.25

Table 11: Additional real-world evaluation on subsets of GT-RAIN and FoundIR. Results are averaged over the evaluated images in each dataset; bold indicates the better result.

Dataset	Method	PSNR↑	SSIM↑	LPIPS↓	MANIQA↑	CLIP-IQA↑	MUSIQ↑
GT-RAIN	4KAgent	16.25	0.474	0.375	**0.295**	0.364	**55.21**
	Ours (Full)	**21.53**	**0.593**	**0.278**	0.251	**0.646**	38.99
FoundIR	4KAgent	16.45	0.619	**0.315**	0.252	0.294	45.50
	Ours (Full)	**19.54**	**0.628**	0.317	**0.282**	**0.606**	**50.28**

Effect of sampling ratio. Comparing rows with the same training scope, the 1:3:5 ratio consistently outperforms 1:1:1. For agent-only training, MUSIQ improves substantially on Group A, indicating that emphasizing multi-degradation combinations during training improves the agent's tool selection for complex scenarios. Similar trends hold for full training.

Effect of joint training. Comparing planning-only vs. full training within each ratio, joint optimization provides consistent improvements across all metrics. Under the 1:1:1 ratio, joint training yields moderate gains. Under the 1:3:5 ratio, the improvements are more pronounced, with perceptual quality metrics showing the largest relative gains. This demonstrates that jointly optimizing tools amplifies the benefits of the sampling strategy.

Interaction effect. The best configuration (1:3:5 + Full) achieves substantially better results than either factor alone, with improvements across all distortion and perceptual metrics compared to the baseline (1:1:1 + Agent only). These results validate our joint training approach: the agent learns better tool composition when tools are simultaneously optimized for multi-degradation cooperation. Based on these results, we adopt the 1:3:5 ratio with full joint training as our final configuration.

H.3 Reinforcement Learning Reward Ablation

During planning optimization, we introduce a *consistency reward* that evaluates whether the reasoning process aligns with the model's final decision. We employ Pangu-Embedded-7B [3] to estimate this reward. To assess its effect, we compare the full model with a version where the consistency reward is ablated. As shown in Figure 5, including the consistency reward prevents the rapid decline in response length, ensuring that the model continues to provide rationales when making decisions. We further evaluate both versions under the same setting as in Table 1. As shown in Table 14, the ablated

Table 12: Planning performance before and after OPERA's GRPO training over all test images in Groups A–C, using the same prompt and frozen pretrained tools.

Planner	PSNR↑	SSIM↑	LPIPS↓	MANIQA↑	CLIP-IQA↑	MUSIQ↑
VisualQuality-R1	20.17	0.64	0.46	0.25	0.39	43.72
Ours (Planning)	**21.87**	**0.70**	**0.36**	**0.34**	**0.43**	**56.65**

Table 13: Ablation study on Groups A, B, and C from AgenticIR. We analyze two factors: (1) degradation sampling ratio (1:1:1 balanced vs. 1:3:5 emphasizing multi-degradation), and (2) training scope (planning-only vs. full optimization). The 1:3:5 ratio combined with joint training yields consistent improvements across all groups.

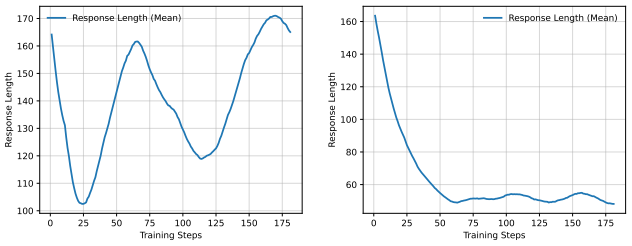
(a) Ratio 1:1:1

Tool	Group	PSNR↑	SSIM↑	LPIPS↓	CLIPIQA↑	MUSIQ↑
Planning Only	A	21.47	0.70	0.35	0.49	53.30
	B	20.95	0.73	0.34	0.50	52.02
	C	19.91	0.60	0.52	0.40	42.26
Full	A	22.14	0.72	0.29	0.51	53.08
	B	21.44	0.74	0.27	0.52	52.04
	C	20.32	0.61	0.40	0.44	44.83

(b) Ratio 1:3:5

Tool	Group	PSNR↑	SSIM↑	LPIPS↓	CLIPIQA↑	MUSIQ↑
Planning Only	A	21.53	0.70	0.32	0.46	**59.02**
	B	21.22	0.73	0.31	0.46	**58.54**
	C	20.27	0.60	0.46	0.37	50.62
Full	A	**23.26**	**0.76**	**0.22**	**0.72**	57.65
	B	**22.81**	**0.76**	**0.20**	**0.70**	57.96
	C	**21.85**	**0.64**	**0.30**	**0.72**	**51.94**

version consistently underperforms the full model, highlighting the positive impact of the consistency reward.



(a) Add Consistency Reward (b) Without Consistency Reward

Figure 5: Comparison of mean response length during training with and without consistency reward.

H.4 Ablation Study on Mixed IQA Metrics for Optimization Objective

In both planning and execution optimization, we adopt an end-to-end training paradigm, where the final image quality serves as the reward/loss. In practice, we optimize a mixture of multiple IQA metrics, as described in Section 4.2 and Section 4.3. This section provides further justification for this design.

Image restoration quality is inherently multi-dimensional, involving both pixel-level fidelity and perceptual realism. It is well known that optimizing a single metric (e.g., PSNR alone) often leads to suboptimal perceptual quality. To address this, we adopt a composite objective that combines both full-reference and no-reference IQA metrics, a strategy commonly used in image restoration literature. These metrics provide complementary supervision signals and help mitigate bias toward any single quality aspect. Similar practices have also been adopted in prior agent-based image restoration methods [4, 48].

We further conduct an ablation study on planning optimization using a reduced set of metrics (CLIP-IQA and MUSIQ). The results are presented in Table 15. The ablated variant shows improved performance on no-reference metrics, but slightly degrades full-reference metrics. This observation highlights the inherent trade-off in objective design, and validates our choice of leveraging multiple complementary metrics to achieve a more balanced overall performance.

Table 14: Ablation study of the consistency reward on Groups A, B, and C from AgenticIR [49].

Method	Group A						Group B						Group C					
	PSNR	SSIM	LPIPS↓	MANIQA	CLIP-IQA	MUSIQ	PSNR	SSIM	LPIPS↓	MANIQA	CLIP-IQA	MUSIQ	PSNR	SSIM	LPIPS↓	MANIQA	CLIP-IQA	MUSIQ
Without Consistency Reward																		
Ours (Planning)	21.83	0.70	0.33	**0.36**	0.47	58.96	22.03	0.76	0.31	0.35	0.46	**59.22**	19.91	0.62	0.45	**0.28**	0.37	50.79
Ours (Full)	23.26	0.76	0.24	0.31	0.70	57.54	24.61	**0.79**	0.22	0.32	0.70	58.18	22.67	**0.67**	0.32	0.23	0.72	48.92
With Consistency Reward																		
Ours (Planning)	22.32	0.71	0.33	0.35	0.45	**59.34**	22.28	0.75	0.31	**0.36**	0.46	59.10	20.75	0.61	0.47	0.28	0.37	49.91
Ours (Full)	**24.81**	**0.77**	**0.23**	0.32	**0.72**	57.84	**24.63**	0.78	**0.20**	0.33	**0.71**	58.41	**23.04**	0.66	**0.30**	0.25	**0.72**	**51.27**

Table 15: Ablation study of IQA metrics in planning optimization reward. The ablated version retains only no-reference metrics CLIP-IQA and MUSIQ.

Method	Group A						Group B						Group C					
	PSNR	SSIM	LPIPS↓	MANIQA	CLIP-IQA	MUSIQ	PSNR	SSIM	LPIPS↓	MANIQA	CLIP-IQA	MUSIQ	PSNR	SSIM	LPIPS↓	MANIQA	CLIP-IQA	MUSIQ
Ours (Planning)	**21.53**	**0.70**	**0.32**	0.35	0.46	59.02	**21.22**	**0.73**	**0.31**	0.35	0.46	58.54	**20.27**	0.60	0.46	**0.28**	0.37	50.62
Reduced Metrics	20.86	0.70	0.33	**0.37**	**0.53**	**66.29**	20.16	0.70	0.35	**0.38**	**0.53**	**64.65**	19.83	**0.61**	**0.44**	0.27	**0.47**	**61.73**

I Tool Behavior Analysis

We analyze how tool training affects tool behavior through two experiments.

Misuse Experiment. Pretrained restoration tools are designed for specific degradations and tend to over-process inputs regardless of whether restoration is necessary. To evaluate whether trained tools exhibit adaptive behavior, we apply each tool to 100 high-quality images containing no degradation and measure the quality degradation introduced.

As shown in Table 16 (left), pretrained tools cause noticeable quality degradation when applied to clean images. In contrast, trained tools preserve image quality more effectively, achieving +7.31 dB higher PSNR and 0.073 lower LPIPS. This demonstrates that trained tools better preserve image quality when applied to images that do not require restoration.

Single-Degradation Experiment. We further examine whether the adaptive behavior observed above comes at the cost of reduced restoration capability. We evaluate each tool on 20 images containing only its corresponding target degradation.

As shown in Table 16 (right), trained tools maintain comparable performance to their pretrained counterparts on single-degradation inputs. The minor performance differences suggest that training does not cause catastrophic forgetting of the original restoration capability.

Summary. These results indicate that trained tools exhibit adaptive behavior, reducing unnecessary modifications while maintaining restoration capability.

J Group Relative Policy Optimization

For reinforcement learning, we adopt Group Relative Policy Optimization (GRPO) [30]. Given an input x, the old policy $\pi_{\theta_{\text{old}}}$ generates a group of G responses $\{y_i\}_{i=1}^{G}$, where $y_i = (y_{i,1}, \ldots, y_{i,T_i})$ contains T_i tokens and receives a reward R_i. With outcome supervision, every token in response y_i shares the group-normalized advantage $A_i = (R_i - \text{mean}(\mathbf{R}))/\,\text{std}(\mathbf{R})$, where $\mathbf{R} = (R_1, \ldots, R_G)$. The policy is optimized by maximizing

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{\substack{x\sim\mathcal{D} \\ \{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)}} \Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{T_i}\sum_{t=1}^{T_i} \times \Big\{ \min\big(r_{i,t}(\theta)A_i,\ \text{clip}(r_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon)A_i\big) - \beta K_{i,t}(\theta) \Big\} \Bigg]. \tag{12}$$

Here, the probability ratio is defined for each token conditioned on the input and preceding tokens:

$$r_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid x, y_{i,<t})}.$$

Table 16: Tool behavior analysis. **Left**: Misuse experiment evaluates tools on high-quality images without degradation. Trained tools substantially reduce quality degradation, indicating adaptive behavior. **Right**: Single-degradation experiment evaluates tools on images with only their target degradation. Trained tools maintain comparable performance, confirming no catastrophic forgetting.

(a) Misuse (HQ images)

Setup	PSNR↑	SSIM↑	LPIPS↓
HQ Reference	∞	1.000	0.000
Pretrained	40.80	0.938	0.084
Trained	**48.11**	**0.990**	**0.011**
Δ (Trained - Pretrained)	+7.31	+0.052	-0.073

(b) Single-degradation

Setup	PSNR↑	SSIM↑	LPIPS↓
Pretrained	**28.62**	**0.767**	**0.282**
Trained	28.50	0.739	0.317
Δ (Trained - Pretrained)	-0.12	-0.028	+0.035

The clipping parameter is ε, and β controls the KL penalty relative to the reference policy π_{ref}. Following GRPO, the token-level KL estimator is

$$K_{i,t}(\theta) = u_{i,t}(\theta) - \log u_{i,t}(\theta) - 1, \qquad u_{i,t}(\theta) = \frac{\pi_{\text{ref}}(y_{i,t} \mid x, y_{i,<t})}{\pi_\theta(y_{i,t} \mid x, y_{i,<t})}.$$

Thus, clipping is applied per token, and the objective averages over tokens within each response and then over responses in the group.

K Limitations and Societal Impacts

Limitations. We acknowledge several limitations of our proposed OPERA framework:

- **Dependence on the tool pool.** The performance of OPERA is inherently bounded by the coverage, diversity, and quality of the available restoration tools. When certain degradation types are unsupported in the tool pool, the system may fail to fully recover the image, leading to suboptimal restoration results.
- **Inference overhead.** Although OPERA is more efficient than search-based agentic systems, it still requires one call to a vision-language model followed by the sequential execution of multiple restoration tools. As a result, the overall inference cost is higher than that of a single end-to-end model, which may limit its applicability in real-time or resource-constrained scenarios.
- **Performance trade-offs between generalization and specialization.** Jointly training restoration tools to cooperate improves robustness in complex, multi-degradation settings. However, as shown in our analysis, such joint optimization may slightly degrade the performance of individual tools on single-degradation tasks. This suggests a trade-off between collaborative generalization and task-specific specialization.
- **Reward mismatch with human perception.** OPERA uses IQA metrics and an LLM-based consistency judgment as optimization signals. These proxies do not fully capture human preferences or the fidelity of restored details. Combining multiple metrics mitigates dependence on a single measure, but does not eliminate the possibility of improving reward scores without a corresponding improvement in perceived quality. Human evaluation and more reliable perceptual objectives remain important directions for future work.
- **Potentially harmful or unnecessary tool selections.** The planner may invoke tools whose target degradations are absent from the input. We analyze such invocations across all samples in the main experiment by comparing the original plans with counterparts in which the irrelevant tool is removed. As shown in Table 17, the effects are tool-dependent: irrelevant denoising and dehazing reduce average PSNR and CLIP-IQA, whereas the evaluated deblurring tools yield positive average changes. Among images invoking deblurring tools, 67% contain irrelevant deblurring calls; the corresponding rate for denoising is 7.3%. This pattern is consistent with the planner favoring tools that are more tolerant of out-of-scope use. Nevertheless, harmful tool selections remain a source of restoration errors, motivating more reliable assessment of whether individual tool calls are beneficial.

Table 17: Effects of irrelevant tool use in the main experiment. Values are average metric differences between the original plan and the plan with the irrelevant tool removed. Positive values indicate a benefit from retaining the tool; negative values indicate harm.

Tool (when irrelevant)	ΔPSNR (dB)	ΔCLIP-IQA
Restormer Denoise	-3.98	-0.28
X-Restormer Dehaze	-1.72	-0.01
X-Restormer Denoise	-1.64	-0.02
Restormer Defocus Deblur	$+1.40$	$+0.14$
Restormer Motion Deblur	$+0.16$	$+0.05$

Societal Impacts. This paper presents a method for advancing image restoration under complex real-world degradations by improving agent-based planning and tool cooperation. Overall, we believe this work contributes positively to the field of machine learning and computer vision, and its broader societal implications are consistent with established research in image restoration. No significant potential societal consequences of this work must be specifically highlighted here.

L More Qualitative Comparisons

Figure 6 and Figure 7 present more visual comparisons.

M Prompt Templates

The prompt for the LLM judge to decide the consistency reward is shown in Table 18. The prompt for the planning agent is shown in Table 19.

Table 18: Full prompt used for calculating consistency reward from the LLM judge.

Usage	**Prompt**
System Prompt	You are a rigorous planning problem evaluator. I will provide you with two parts: A Reasoning Process describing how a planning problem is analyzed and solved A Final Plan representing the final planning decision or outcome Your task is to evaluate them according to the following criteria: 1. Evaluate the Reasoning Process - The reasoning process must NOT be empty - It must contain meaningful, coherent, and logical reasoning steps - It should include analysis of constraints, assumptions, or decision logic - If the reasoning process is missing, empty, superficial, or logically flawed, mark it as unreasonable 2. Check Consistency Between Reasoning Process and Final Plan - The final plan must be logically derivable from the reasoning process - There should be no contradictions between the reasoning process and the final plan - If the reasoning supports one conclusion but the final plan states another, mark them as inconsistent 3. Provide a Clear Judgment and Explanation Only output a single “Yes” or “No”. Do not provide other explanations or text.

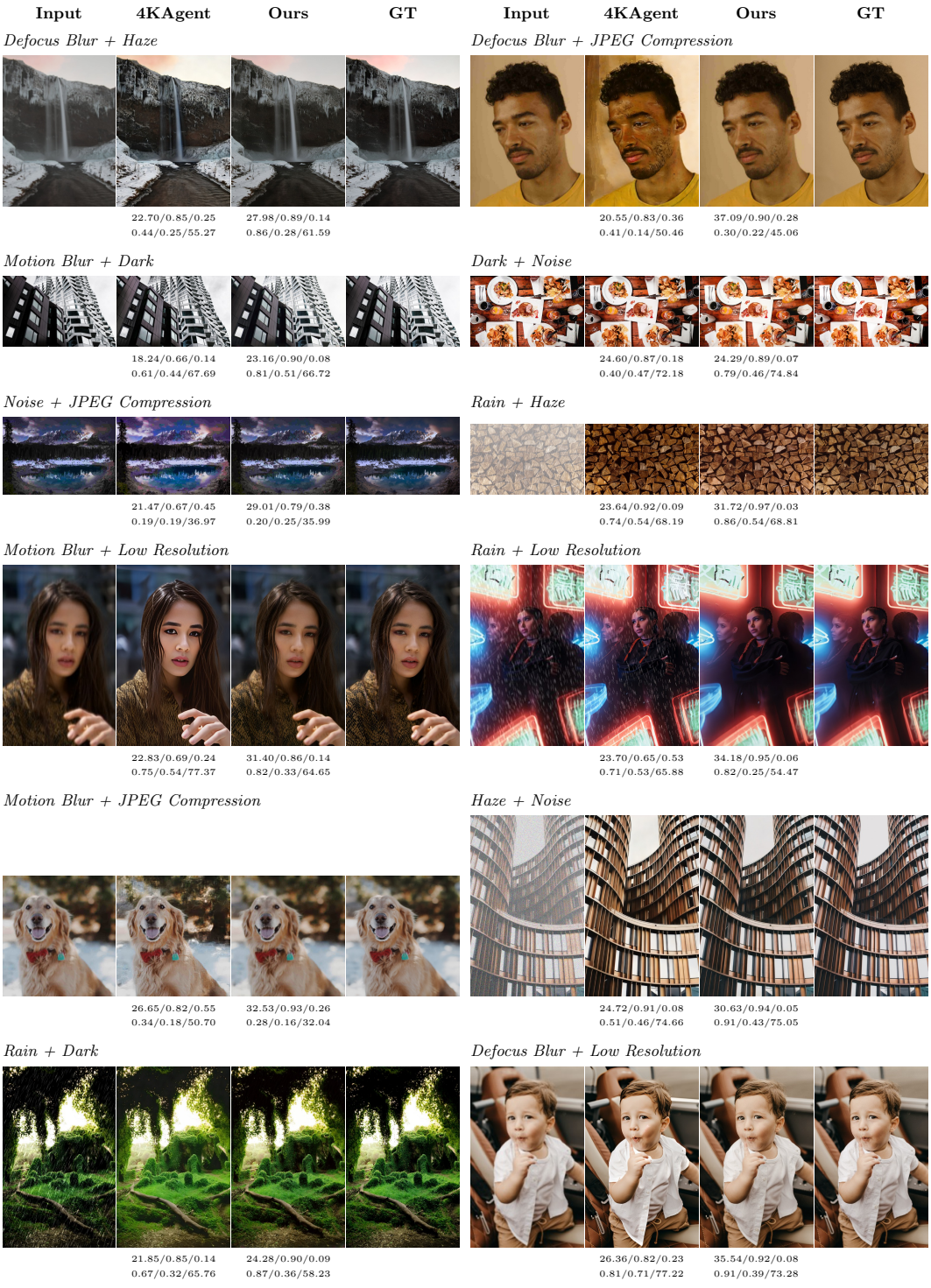


Figure 6: Qualitative comparison across degradation categories of Groups A and B on the AgenticIR benchmark (part 1/2). Metrics reported: PSNR/SSIM/LPIPS↓/CLIP-IQA/MANIQA/MUSIQ.

Table 19: Full prompt used for the planning agent.

Usage	Prompt
System Prompt	You are a professional image restoration assistant. You will be given an image as input. Your task is to: 1. Visually analyze the image and identify what degradations it contains. 2. Design an optimal sequence of restoration tool calls to enhance the image quality. # Possible Degradations: - noise - rain - haze - defocus_blur - motion_blur - low_resolution - jpeg - low_light # Tools from Restormer - restormer.gaussian_denoise_15 - restormer.gaussian_denoise_25 - restormer.gaussian_denoise_50 - restormer.derain - restormer.defocus_deblur - restormer.motion_deblur # Tools from X-Restormer - xrestormer.denoise_50 - xrestormer.derain - xrestormer.dehaze - xrestormer.deblur - xrestormer.super_resolution # Tools from SWIN-IR - swinir.super_resolution - swinir.gaussian_denoise_15 - swinir.gaussian_denoise_25 - swinir.gaussian_denoise_50 - swinir.dejpeg # Tools for low light - opencv.constant_shift - opencv.gamma_correction First, think visually in <think> </think> by describing the quality of the image and how you plan to restore it. Then, output a Python list of the detected degradations in <degradation> </degradation>. Finally, output a Python list of the restoration plan in the order in <answer> </answer>. e.g.: <think> thinking progress here </think> <degradation>['rain', 'noise']</degradation> <answer> ['restormer.gaussian_denoise_25', 'xrestormer.derain']</answer>. Note that the **order** of tools matters, and tools from different repos may behave differently. Select carefully.
User	<image> How to restore this image? Think first then answer.

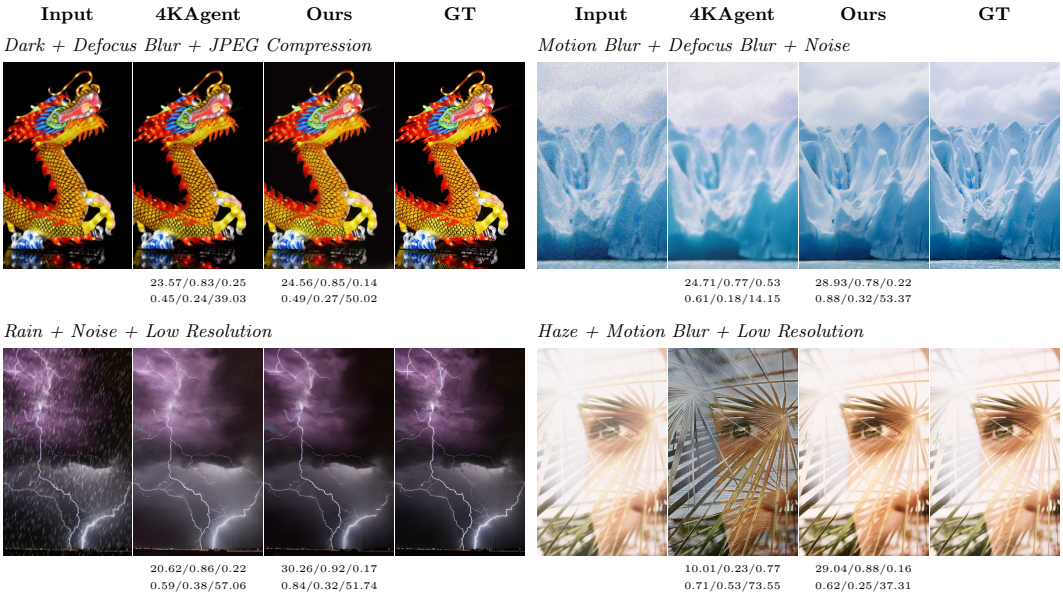


Figure 7: Qualitative comparison on Group C triple-degradation categories (part 2/2). Metrics reported: PSNR/SSIM/LPIPS↓/CLIP-IQA/MANIQA/MUSIQ. Continuation of Fig. 6.

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 accurately state the contributions and scope of the paper.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discussed the limitations in Appendix K.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

Justification: The paper does not include theoretical results, such as theorems or proofs, as it focuses on methodological contributions and empirical validation.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: We provide comprehensive implementation details in the paper, including Appendix D and Section 5.1. In addition, we release our codebase along with detailed documentation in the supplementary materials, which are sufficient for reproducing our experimental results.

Guidelines:

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

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

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

   Answer: [Yes]

   Justification: We release our codebase along with detailed documentation in the supplementary materials, which are sufficient for reproducing our experimental results.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We provide comprehensive implementation details in the paper, including Appendix D and Section 5.1. Code is also provided in the supplemental material for unambiguous details.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: The seeds are fixed for both restoration tools and the planning agent, which ensures the consistency between multiple runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide this information in Appendix E.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We carefully confirm that the paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discussed the broader impacts in Appendix K.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We use several existing assets in the paper. We confirm they are properly credited and respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: New assets include the codebase and trained model checkpoints. We have provided comprehensive documentation.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: Our method includes training an LLM as the planning agent, which is detailed in Section 4.2 and Section 5.1.

    Guidelines:

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