# RestoreAgent: Autonomous Image Restoration Agent via Multimodal Large Language Models

**Haoyu Chen**[1], **Wenbo Li**[2], **Jinjin Gu**[3], **Jingjing Ren**[1], **Sixiang Chen**[1],
**Tian Ye**[1], **Renjing Pei**[2], **Kaiwen Zhou**[2], **Fenglong Song**[2], **Lei Zhu**[1,4*]

[1]The Hong Kong University of Science and Technology (Guangzhou) [2]Huawei Noah's Ark Lab
[3]The University of Sydney [4]The Hong Kong University of Science and Technology
Project page: https://haoyuchen.com/RestoreAgent

## Abstract

Natural images captured by mobile devices often suffer from multiple types of degradation, such as noise, blur, and low light. Traditional image restoration methods require manual selection of specific tasks, algorithms, and execution sequences, which is time-consuming and may yield suboptimal results. All-in-one models, though capable of handling multiple tasks, typically support only a limited range and often produce overly smooth, low-fidelity outcomes due to their broad data distribution fitting. To address these challenges, we first define a new pipeline for restoring images with multiple degradations, and then introduce RestoreAgent, an intelligent image restoration system leveraging multimodal large language models. RestoreAgent autonomously assesses the type and extent of degradation in input images and performs restoration through (1) determining the appropriate restoration tasks, (2) optimizing the task sequence, (3) selecting the most suitable models, and (4) executing the restoration. Experimental results demonstrate the superior performance of RestoreAgent in handling complex degradation, surpassing human experts. Furthermore, the system's modular design facilitates the fast integration of new tasks and models.

## 1 Introduction

Image restoration, a classical research area in computer vision, focuses on recovering high-quality images from degraded observations. Traditional methods are usually tailored to specific tasks like denoising [55, 61, 49, 29, 30, 12, 3], super-resolution [56, 34, 53, 4, 45, 47, 48], and deblurring [28, 22, 51, 32, 44, 17]. However, real-world images often suffer from multiple simultaneous degradations. For example, a low-quality image may exhibit noise, blur, and rain concurrently. There may exist complex interactions and dependencies among different degradation phenomena, and each degradation may require distinct handling methods. The combination and sequence of these methods are crucial for the final restoration outcome. Recent advancements in the field have been driven by leveraging expert knowledge and developing all-in-one models. To provide a thorough understanding of this field and clarify our motivation, we present a detailed analysis below.

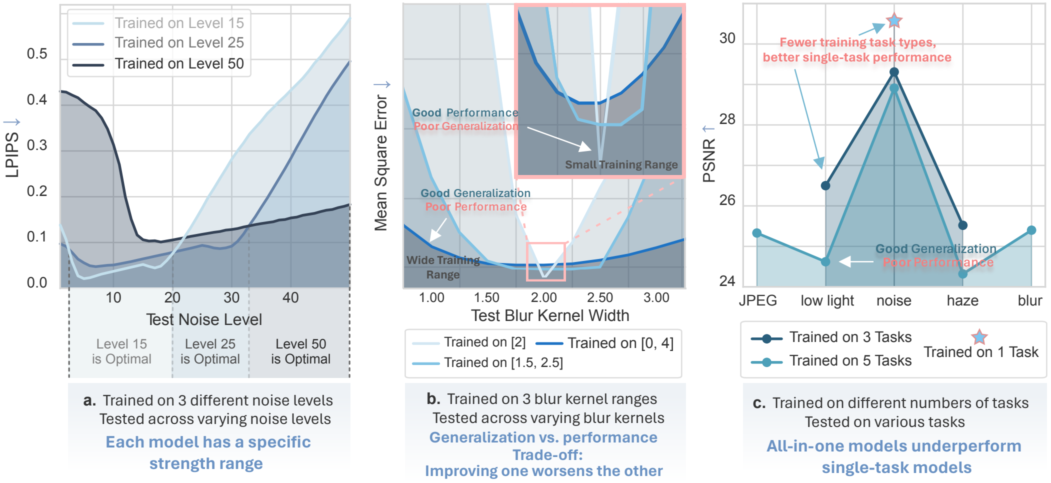
### 1.1 All-in-One Models

All-in-one models [38, 31, 24, 40, 33, 14, 27, 37, 1, 25] seek to use a single framework to handle multiple degradations simultaneously. By training on multi-task datasets, these models learn to manage various restoration tasks. However, several limitations continue to impede the practicality of these models in complex real-world scenarios:

**Restricted task scope.** All-in-one models often struggle to process degradations outside of their training data. Even for the same type of degradation, as shown in Figure 2 **a1**, these models may have difficulty effectively processing data if the degradation distribution varies between the training and testing sets. Given that existing models only cover a limited number of tasks, employing specialized single-task restoration models is often more flexible and effective.

[*] Lei Zhu (leizhu@ust.hk) is the corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).



**a.** Trained on 3 different noise levels Tested across varying noise levels
**Each model has a specific strength range**

**b.** Trained on 3 blur kernel ranges Tested across varying blur kernels
**Generalization vs. performance Trade-off: Improving one worsens the other**

**c.** Trained on different numbers of tasks Tested on various tasks
**All-in-one models underperform single-task models**

Figure 1: Limitations of all-in-one models. **(a)** Models trained on different noise levels excel in specific areas, so choosing models on demand leads to better results. **(b)** Models trained on a wider range of blur degradations offer improved generalization but compromised performance, showing a trade-off. **(c)** Multi-task models underperform on individual tasks compared to single-task models, illustrating that all-in-one models trade performance for generalization.

**Compromised performance.** All-in-one models often face trade-offs between generalization and restoration accuracy, as shown in Figure 1. While these models offer improved generalization across a broader range of degradation levels, their performance at specific levels may be compromised. Additionally, because they must handle multiple tasks with largely disparate degradation patterns, the performance for individual tasks may fall short, resulting in overly smoothed outputs. As illustrated in Figure 2 **a2**, single-task models typically outperform all-in-one models in most scenarios.

All-in-one models can, in fact, be integrated into an agent system comprising multiple models, thereby going beyond a single solution. Often, using task-specific models customized for particular degradations and then integrating them with an all-in-one model yields improved performance, as shown by the two examples in Figure 2 **a3**. This hybrid approach maintains the adaptability of all-in-one models while leveraging the strengths of specialized models.

### 1.2 Task-Specific Models

An alternative approach to using all-in-one models, which struggle to effectively address various types of degradation, is to combine several specialized task-specific models, each focusing on a specific degradation type. This modular strategy allows for a more targeted and efficient handling of the different degradations present in the input images. Superior results can be achieved because these specialized models excel in their respective areas.

#### 1.2.1 Fixed or Random Execution Order

Current methods [50, 24, 14] typically detect the types of degradation in an image and apply the appropriate restoration models in a predetermined order, or manually selected by experts, or chosen at random. Nevertheless, there is a significant drawback to this approach: the processing order has a major impact on the final performance. A predetermined order, even if established by human experts, is not ideal and might fail to successfully restore the image, as demonstrated in Figure 2 **b**. Two primary causes can be identified for this.

**First, applying one restoration method can alter other degradation patterns**, rendering the following restoration models ineffective. For example, in an image with haze and rain, if haze is performed first (Figure 2 **b**), the dehazing model may address the blur but alter the rain distribution, thereby reducing the effectiveness of the deraining model.

**Second, removing some degradations can be challenging if other degradations have not been addressed first**. A common example is the enhancement of low-light images, which often requires denoising as a pre-processing step. Without prior denoising, the results of low-light enhancement are likely to be subpar. In Figure 2 **b**, we can observe that without prior denoising and deraining, the performance of the dehazing model is significantly compromised.

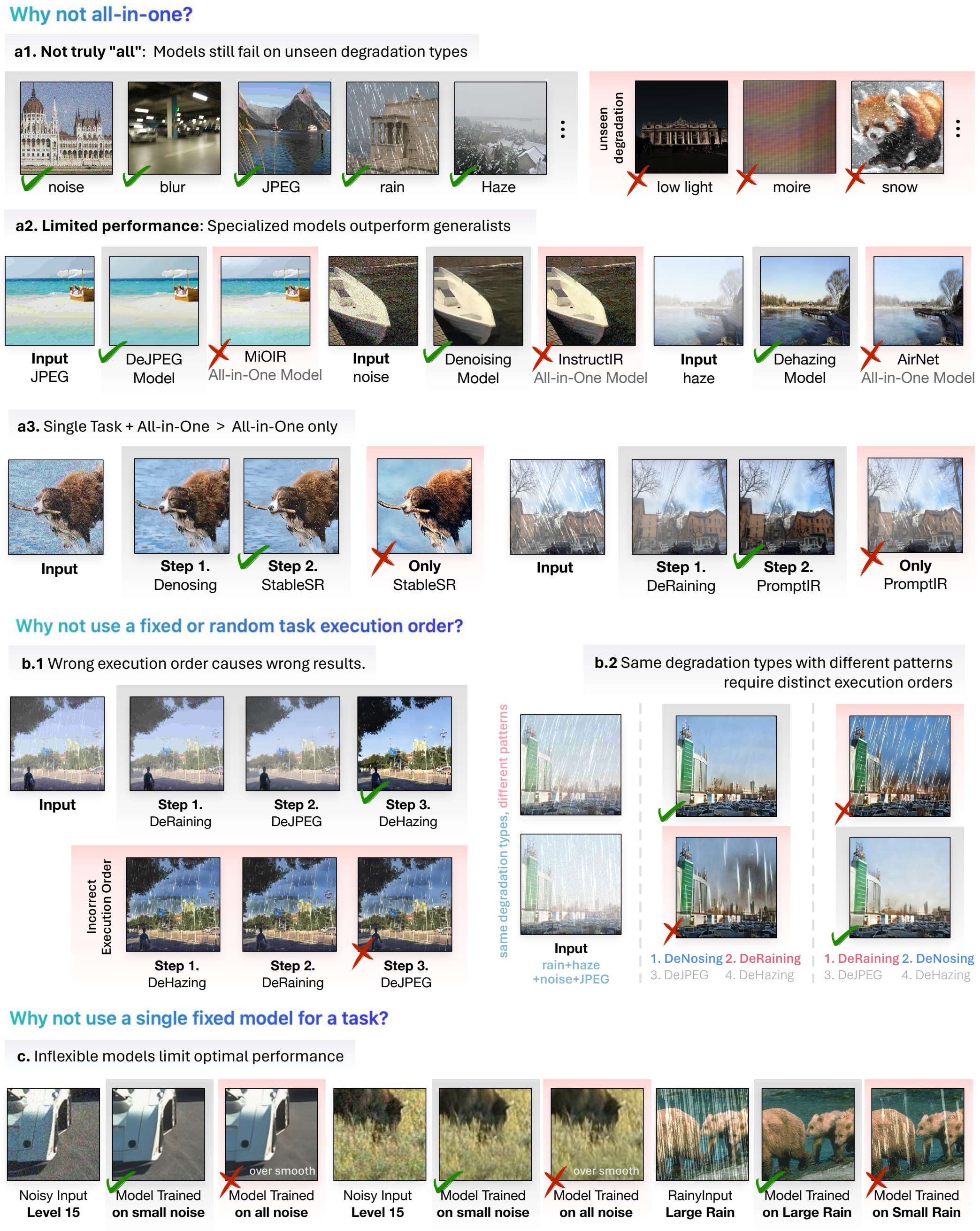

Why not all-in-one?
a1. Not truly "all": Models still fail on unseen degradation types
noise
blur
JPEG
rain
Haze
unseen degradation
low light
moire
snow
a2. Limited performance: Specialized models outperform generalists
Input JPEG
DeJPEG Model
MiOIR All-in-One Model
Input noise
Denoising Model
InstructIR All-in-One Model
Input haze
Dehazing Model
AirNet All-in-One Model
a3. Single Task + All-in-One > All-in-One only
Input
Step 1. Denosing
Step 2. StableSR
Only StableSR
Input
Step 1. DeRaining
Step 2. PromptIR
Only PromptIR
Why not use a fixed or random task execution order?
b.1 Wrong execution order causes wrong results.
Input
Step 1. DeRaining
Step 2. DeJPEG
Step 3. DeHazing
Incorrect Execution Order
Step 1. DeHazing
Step 2. DeRaining
Step 3. DeJPEG
b.2 Same degradation types with different patterns require distinct execution orders
same degradation types, different patterns
Input
rain+haze +noise+JPEG
1. DeNosing 2. DeRaining 3. DeJPEG 4. DeHazing
1. DeRaining 2. DeNosing 3. DeJPEG 4. DeHazing
Why not use a single fixed model for a task?
c. Inflexible models limit optimal performance
Noisy Input Level 15
Model Trained on small noise
over smooth
Model Trained on all noise
Noisy Input Level 15
Model Trained on small noise
over smooth
Model Trained on all noise
RainyInput Large Rain
Model Trained on Large Rain
Model Trained on Small Rain

Figure 2: Limitation illustration of all-in-one models, fixed task execution order, and fixed model. Images with a pink background indicate negative examples

.

In light of these findings, accurate identification of degradation patterns or careful testing of various task execution sequences is necessary for high-quality restoration. However, the search space grows significantly with the number of tasks. For example, there are 24 possible execution orders for 4 degradation types. Moreover, the number of permutations increases drastically when multiple models are available for a given task, leading to a significant rise in computational complexity.

### 1.2.2 Fixed or Random Model for a Single Task

In some scenarios, the system may opt to use a single model for a specific task or randomly select a model from a pool of available options [50]. However, this approach has significant drawbacks. Image restoration is a rapidly evolving field with various models tailored for a specific task, each with unique capabilities and areas of expertise for managing specific scenarios. Using a fixed model or randomly selecting from a pool of models to process complex degradations can lead to suboptimal results. As illustrated in Figure 2 **c** and Figure 1 **a**, different denoising models excel at different noise levels. Choosing the right model is crucial for achieving the best result.

Manually selecting the best model is impractical due to the numerous combinations of task execution orders and available models. For example, with 3 degradation types and 3 models per type, there are 162 possible combinations. Evaluating these permutations is time-consuming and labor-intensive. Consequently, we often rely on one or two experienced-based solutions, which may not achieve the desired restoration effect.

### 1.3 RestoreAgent

In response to the aforementioned challenges, we propose RestoreAgent, an autonomous and intelligent image restoration system based on a multimodal large language model (MLLM). The MLLM's exposure to vast and diverse data endows it with superior generalization capabilities and has showcased remarkable performance in visual understanding and logical reasoning [46, 35, 18, 39, 43, 6, 62]. Furthermore, its flexibility facilitates the quick addition of new tasks, the definition of desired output formats, and easier human interaction.

Our framework offers the following functionalities:

**(1) Degradation Type Identification.** RestoreAgent automatically identifies the types of degradation present in an input image and determines the corresponding restoration tasks required.

**(2) Adaptive Restoration Sequence.** RestoreAgent goes beyond the constraints of predefined, human-specified model execution orders by dynamically evaluating the individual properties of each input image to decide the best sequence for utilizing the restoration models, thereby enhancing the overall efficiency of the image restoration procedure.

**(3) Optimal Model Selection.** Based on the specific degradation patterns in the input image, RestoreAgent dynamically selects the most appropriate model from the available pool for each restoration task, ensuring optimal performance.

**(4) Automated Execution.** Once the restoration sequence and model selection are determined, RestoreAgent autonomously executes the entire restoration pipeline without the need for manual intervention.

To this end, we start by defining the multi-degradation task and constructing a training dataset. This dataset includes paired degraded images (with one or more degradation types) and their ground truth (only for evaluation), along with the optimal task execution sequence and best model choice based on user-preferred goals. We then fine-tune MLLM to enable RestoreAgent to autonomously make task decisions and determine the optimal processing sequence and models. Experiments show that RestoreAgent's decision-making capabilities significantly outperform existing methods and human experts, achieving superior performance in recovering multi-degradation images. Notably, our method can quickly adapt to unseen tasks and models.

## 2 Related Work

### 2.1 Single-Task Image Restoration

In the field of single-task image restoration, numerous methods have focused on addressing specific types of image degradation. In denoising, models like DnCNN [59] and RNAN [63] have demonstrated significant effectiveness, among others. In deblurring, algorithms like DeblurGAN [28] and MIMO-UNet [13] and others stand out. For reducing JPEG artifacts, methods such as DCSC [19] and FBCNN [23] are particularly well-suited. Additionally, there are specialized methods for restoration under adverse weather conditions, including dehazing [52, 41], deraining [11, 7], and desnowing [8, 9, 5]. Each task often requires a specialized approach, leading to highly optimized algorithms that achieve sota performance for their specific targets compared to universal approaches.

### 2.2 All-in-One Image Restoration

Recent research has explored the development of All-in-One models that attempt to handle multiple degradation types simultaneously within a single framework. This kind of methods are trained to recognize and correct various forms of degradation concurrently. AirNet [31] featuring the contrastive-based degraded encoder and degradation-guided all-in-one restoration network. ADMS [38] uses adaptive filters to efficiently restore images with unknown degradations. TAPE [36] embeds a task-agnostic prior into a transformer, utilizing a two-stage process of pre-training and fine-tuning to enhance image restoration. PromptIR [40] and PIP [33] both utilize uniquely designed prompts to guide their networks. MiOIR [27] employs sequential and prompt learning strategies, which guide the network to incrementally learn individual IR tasks in a sequential manner. MPerceiver [1] employs a multimodal prompt learning approach, utilizing Stable Diffusion priors to achieve high-fidelity all-in-one image restoration.

### 2.3 Agent in Image Restoration

Another research direction focuses on more intelligent image restoration systems. One class of such methods employs a toolbox approach to address image degradation separately. RL-Restore [57] prepares a toolbox consisting of small-scale convolutional networks, each specialized in different tasks. The system then learns a policy to select appropriate tools from the toolbox to progressively restore the quality of a corrupted image. However, RL-Restore supports only three types of degradation: blur, noise, and JPEG compression, which constrains its application scenarios and prevents it from utilizing new state-of-the-art models. Clarity ChatGPT [50] combines the conversational intelligence of ChatGPT with multiple image restoration methods. It automatically detects types of image degradation and selects appropriate methods to restore images. Conversely, Clarity ChatGPT identifies the presence of degradation but lacks research and design on the execution order of tasks and the optimal model selection for specific degradations in the input image.

Another class involves all-in-one approaches with degradation-aware guidance. InstructIR [14] pioneers a novel approach by utilizing human-written instructions to guide the recovery from various types of degradation. AutoDIR [24] automatically detect and restore images with multiple unknown degradations. LLMRA [25] generates text descriptions and encodes them as context embeddings with degradation information, and integrates these context embeddings into the restoration network. DA-CLIP [37] presents a degradation-aware vision-language model that guides the model to learn high-fidelity image reconstruction. For these all-in-one restoration assistant methods, inherent limitations exist in the practical applications of all-in-one models.

How to overcome these limitations, fully leverage the wide array of state-of-the-art models for different tasks available on the market, and determine the optimal execution sequence of image restoration tasks and the most suitable model for specific degradation pattern remain unexplored. This gap presents a significant opportunity for future research in intelligent image restoration systems.

## 3 RestoreAgent

In this section, we introduce RestoreAgent, an advanced image restoration agent designed to find the optimal model and execution sequence from a model pool to process images containing multiple degradations. RestoreAgent is built upon a state-of-the-art multimodal large language model, which possesses remarkable reasoning, generalization, and cross-modal understanding capabilities. By leveraging the model's ability to draw insights from vast amounts of multimodal data, establish connections between visual and textual information, and apply that knowledge to new contexts, RestoreAgent can effectively analyze complex image degradation scenarios, infer the most suitable restoration techniques, and generate optimal pipelines that combine the strengths of various specialized models. As a result, RestoreAgent consistently produces high-quality results.

In Section 3.1, we first define the problem of identifying the most effective combination and order of models from a given pool to restore images affected by various types of degradation. Next, in Section 3.2.2, we describe the process of constructing the training data for the RestoreAgent. The training data consists of paired samples, each containing a degraded image and its corresponding optimal restoration pipeline. Finally, we detail the training process of RestoreAgent, which involves fine-tuning the Llava-Llama3-8b model using the constructed training data in Section 3.2. By learning from these examples, RestoreAgent acquires the ability to analyze degraded images and generate optimal restoration pipelines based on the available model pool.

### 3.1 Problem Definition

We consider a comprehensive set of degradation types, denoted as $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$, where each $d_i$ represents a specific type of image degradation such as noise, JPEG artifacts, blur, rain streaks, fog, and low light conditions. For each degradation type $d_i$, we tailor a model library $\mathcal{M}_{d_i}$, comprising models $\{M_{d_i}^1, M_{d_i}^2, \ldots\}$. Each model $M_{d_i}^j$ is specifically trained to mitigate the effects of degradation $d_i$. The problem is formally defined as follows:

**Input:** A degraded image $I$ subjected to various degradation types $\mathcal{D}$. A model library $\{\mathcal{M}_{d_1}, \mathcal{M}_{d_2}, \ldots, \mathcal{M}_{d_n}\}$ tailored for processing $\mathcal{D}$. The user-provided scoring function $S$ for evaluating the image restoration process.

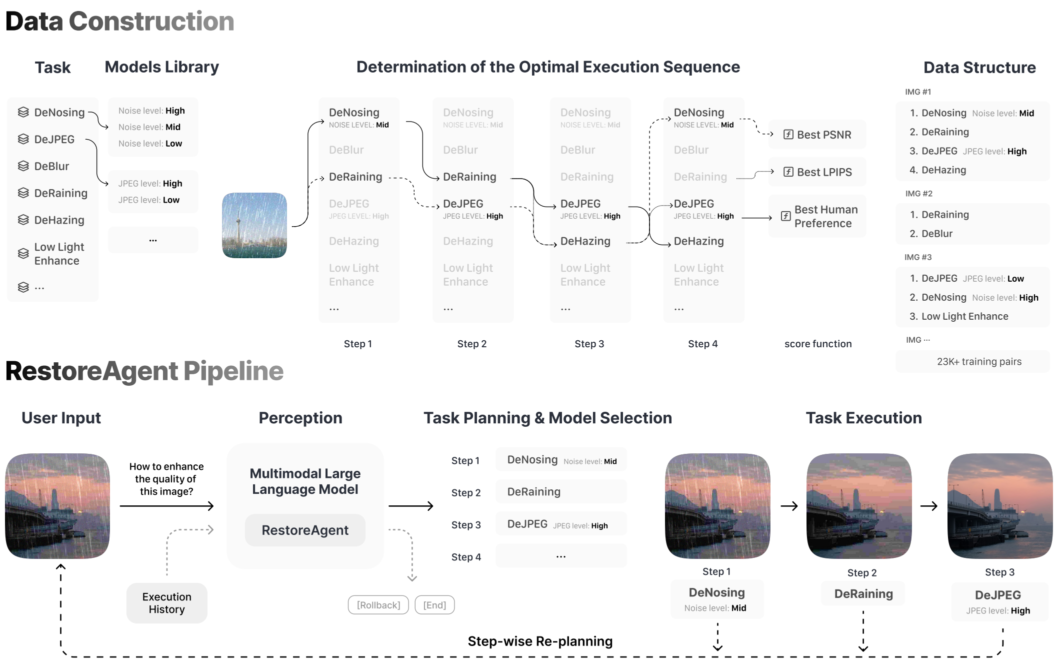


Figure 3: Illustration of the data construction workflow and RestoreAgent pipeline.

**Objective:** Identify the optimal model execution sequence $\sigma = (M_{a_1}^{b_1}, M_{a_2}^{b_2}, \ldots, M_{a_m}^{b_m})$ that maximizes the restoration quality $S$ of the degraded image $I$, where $a_i$ denotes the degradation type and $b_i$ represents the corresponding model. It is formulated as:

$$\sigma^* = \arg \max_{\sigma \in \mathfrak{S}(\mathcal{D}, \mathcal{M})} S(I, \sigma)\,,$$

where $\mathfrak{S}(\mathcal{D}, \mathcal{M})$ represents the set of all possible sequences of degradation and model pairs.

By tackling this problem, we strive to identify the optimal combination of restoration sequence and model selections, ultimately enhancing the quality of images affected by multiple degradations in real-world settings, and thus providing a more effective and efficient solution for complex image restoration tasks.

## 3.2 RestoreAgent: An Advanced Image Restoration System

### 3.2.1 RestoreAgent Pipeline

We introduce an advanced image restoration agent, dubbed RestoreAgent, implemented using the state-of-the-art multimodal model Llava-Llama3-8b [46]. LoRA [21] is utilized to fine-tune both the vision and language modules. As shown in Figure 3, given a degraded input image, RestoreAgent can provide the best decisions, including which image restoration tasks need to be performed, the order of their execution, and which model is most suitable for each task. The model's input consists of a degraded image and the prompt such as `User: How to enhance the quality of this image? [Execution history: ...]`. In response, RestoreAgent generates an output sequence representing the optimal restoration pipeline, comprising a series of tasks, each associated with a specific model best suited to address particular degradation patterns. In our implementation, the output template is defined as: `Agent:1.<task name><model name>. 2.<task name><model name>. 3. ...`, ensuring interpretability and actionability.

RestoreAgent also supports an iterative step-wise decision-making process, reevaluating the state of the image after each restoration step. During this reassessment, the execution history is provided, offering valuable context for decision-making. This allows for real-time strategy adjustments based on cumulative effects and past actions. The system also features a rollback capability, enabling it to revert to a previous state if undesirable results are detected. This combination of iterative evaluation with historical context and rollback allows for finer control over the restoration process, facilitating mid-course corrections.

Table 1: Comparison of RestoreAgent with other decision-making strategies for multi-degraded image restoration. The "balanced" column represents the sum of the four normalized metrics, which is our score function to train the model. The "ranking" column indicates the ranking of the given decision among all possible decisions, with the total number of decisions for each test set provided. The final group presents the Average Result Across All Datasets , providing an overall performance.

| | **Noise + JPEG** | | | | | | **Low Light + Noise** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | DISTS ↓ | balanced ↑ | ranking /17 | PSNR ↑ | SSIM ↑ | LPIPS ↓ | DISTS ↓ | balanced ↑ | ranking /10 |
| Random Order & Model | 24.52 | 0.7273 | 0.2889 | 0.2212 | 1.47 | 6.7 | 15.57 | 0.6541 | 0.4351 | 0.2588 | 1.98 | 3.9 |
| Random Oder + Predict Model | 25.24 | 0.7765 | 0.2327 | 0.1960 | 3.07 | 4.2 | 15.62 | 0.6887 | 0.3651 | 0.2283 | 3.03 | 3.0 |
| Random Model + Predict Order | 24.90 | 0.7568 | 0.2597 | 0.2132 | 2.03 | 6.0 | 17.57 | 0.7044 | 0.3685 | 0.2324 | 3.75 | 2.3 |
| Pre-defined Oder and Model | 25.29 | 0.7828 | 0.2366 | 0.2037 | 2.47 | 5.3 | 17.75 | 0.7098 | 0.3385 | 0.2260 | 3.93 | 2.1 |
| Human Expert | 25.06 | 0.7588 | 0.2551 | 0.2121 | 2.25 | 5.5 | 18.05 | 0.7239 | 0.3278 | 0.2220 | 4.29 | 1.9 |
| **RestoreAgent** | 25.32 | 0.7806 | 0.2308 | 0.1958 | 3.17 | 3.9 ↑1.6 | 17.80 | 0.7226 | 0.3259 | 0.2138 | 4.39 | 1.7 ↑0.2 |
| | **Motion Blur + Noise + JPEG** | | | | | | **Rain + Noise + JPEG** | | | | | |
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | DISTS ↓ | balanced ↑ | ranking /64 | PSNR ↑ | SSIM ↑ | LPIPS ↓ | DISTS ↓ | balanced ↑ | ranking /64 |
| Random Order & Model | 24.81 | 0.7816 | 0.2381 | 0.1747 | 2.32 | 19.5 | 25.64 | 0.7970 | 0.2412 | 0.2020 | 2.90 | 16.1 |
| Random Oder + Predict Model | 24.73 | 0.7787 | 0.2261 | 0.1684 | 2.69 | 16.1 | 25.67 | 0.8008 | 0.2368 | 0.1956 | 3.11 | 15.0 |
| Random Model + Predict Order | 24.95 | 0.7912 | 0.2263 | 0.1647 | 3.18 | 13.6 | 26.14 | 0.8074 | 0.2314 | 0.1996 | 3.49 | 13.3 |
| Pre-defined Oder and Model | 24.84 | 0.7895 | 0.2305 | 0.1662 | 2.97 | 15.0 | 25.80 | 0.7981 | 0.2360 | 0.2041 | 2.83 | 16.7 |
| Human Expert | 25.20 | 0.795 | 0.2205 | 0.1646 | 3.82 | 9.0 | 25.99 | 0.8063 | 0.2258 | 0.1992 | 3.58 | 12.6 |
| **RestoreAgent** | 25.16 | 0.7939 | 0.2042 | 0.1546 | 4.35 | 4.6 ↑4.4 | 26.38 | 0.8136 | 0.2200 | 0.1891 | 4.67 | 6.4 ↑6.2 |
| | **Haze + Noise + JPEG** | | | | | | **Haze + Rain + Noise + JPEG** | | | | | |
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | DISTS ↓ | balanced ↑ | ranking /64 | PSNR ↑ | SSIM ↑ | LPIPS ↓ | DISTS ↓ | balanced ↑ | ranking /287 |
| Random Order & Model | 18.98 | 0.7156 | 0.3267 | 0.2212 | 1.52 | 23.4 | 15.13 | 0.6300 | 0.4464 | 0.2800 | 1.28 | 102.5 |
| Random Oder + Predict Model | 19.00 | 0.7235 | 0.3133 | 0.2081 | 2.03 | 20.4 | 17.45 | 0.6897 | 0.3692 | 0.2400 | 2.86 | 72.8 |
| Random Model + Predict Order | 19.67 | 0.7653 | 0.2778 | 0.2010 | 2.95 | 15.9 | 19.79 | 0.7833 | 0.2815 | 0.1991 | 5.66 | 16.9 |
| Pre-defined Oder and Model | 19.47 | 0.7803 | 0.2641 | 0.1912 | 3.51 | 12.4 | 19.29 | 0.7785 | 0.2815 | 0.1974 | 5.502 | 26.1 |
| Human Expert | 19.50 | 0.7753 | 0.2703 | 0.1982 | 3.36 | 12.7 | 19.39 | 0.7802 | 0.2928 | 0.2043 | 5.503 | 21.3 |
| **RestoreAgent** | 19.55 | 0.7794 | 0.25663 | 0.1863 | 3.93 | 8.4 ↑4.3 | 19.72 | 0.7816 | 0.2741 | 0.1903 | 5.86 | 9.7 ↑11.6 |
| | **Motion Blur + Rain + Noise + JPEG** | | | | | | **Average Result Across All Datasets** | | | | | |
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | DISTS ↓ | balanced ↑ | ranking /287 | PSNR ↑ | SSIM ↑ | LPIPS ↓ | DISTS ↓ | balanced ↑ | ranking (%) |
| Random Order & Model | 21.96 | 0.6672 | 0.3366 | 0.2239 | 2.57 | 85.3 | 21.31 | 0.7139 | 0.3246 | 0.2241 | 1.92 | 34.7 |
| Random Oder + Predict Model | 22.11 | 0.6667 | 0.3038 | 0.2122 | 3.66 | 58.8 | 21.74 | 0.7385 | 0.2848 | 0.2045 | 2.89 | 26.1 |
| Random Model + Predict Order | 22.74 | 0.6996 | 0.2794 | 0.1979 | 5.39 | 24.7 | 22.42 | 0.7574 | 0.2750 | 0.2027 | 3.44 | 22.7 |
| Pre-defined Oder and Model | 22.35 | 0.6862 | 0.2858 | 0.1997 | 4.65 | 35.7 | 22.38 | 0.7639 | 0.2644 | 0.1986 | 3.48 | 22.1 |
| Human Expert | 22.96 | 0.7092 | 0.2861 | 0.2031 | 5.42 | 21.2 | 22.51 | 0.7634 | 0.2670 | 0.2014 | 3.73 | 19.5 |
| **RestoreAgent** | 22.95 | 0.7097 | 0.2615 | 0.1887 | 6.35 | 5.7 ↑15.5 | 22.61 | 0.7700 | 0.2513 | 0.1890 | 4.38 | 12.9 ↑6.6 |

### 3.2.2 Data Construction

To fully leverage the potential of multimodal large models, we construct a substantial dataset of paired training samples. The process begins with applying various types of degradation to an image. Subsequently, we determine the optimal restoration pipeline using model tools for processing. For each image undergoing multiple degradations, a comprehensive search is conducted to identify the best restoration pipeline, as shown in Figure 3. This involves generating all possible permutations of task execution sequences and model combinations, applying each pipeline to the degraded image, and assessing the quality of the restored outputs using a scoring function $S(I, \sigma)$. By comparing the scores of all permutations, the pipeline with the highest score is selected as the optimal processing strategy $\sigma$ for the given image. Users can choose from various image quality assessment methods as the scoring function, customizing the evaluation process to their specific needs. Additional details regarding dataset construction are provided in the supplementary materials.

# 4 Experiment

## 4.1 Experimental Settings

**Scoring function.** To construct a comprehensive evaluation system, we integrate multiple diverse metrics. Specifically, we first standardized each individual metric separately and then summed the standardized results. This process can be described as follows. Let $X_i$ represent the $i$-th metric. We standardize each metric by calculating its z-score: $Z_i = \frac{X_i - \mu_i}{\sigma_i}$ where $\mu_i$ is the mean and $\sigma_i$ is the standard deviation of the $i$-th metric. After standardizing all metrics, we aggregate the standardized scores to form the comprehensive evaluation score $S$: $S = \sum_{i=1}^{n} Z_i$ where $n$ is the total number of metrics. This method ensures that each metric contributes equally to the final evaluation, regardless of its original scale. Follow [26, 20], evaluation metrics primarily include PSNR, SSIM, LPIPS [60], and DISTS [16]. These metrics are widely recognized for their ability to comprehensively reflect the outcomes of image restoration. We also provided the results of models trained on individual metrics.

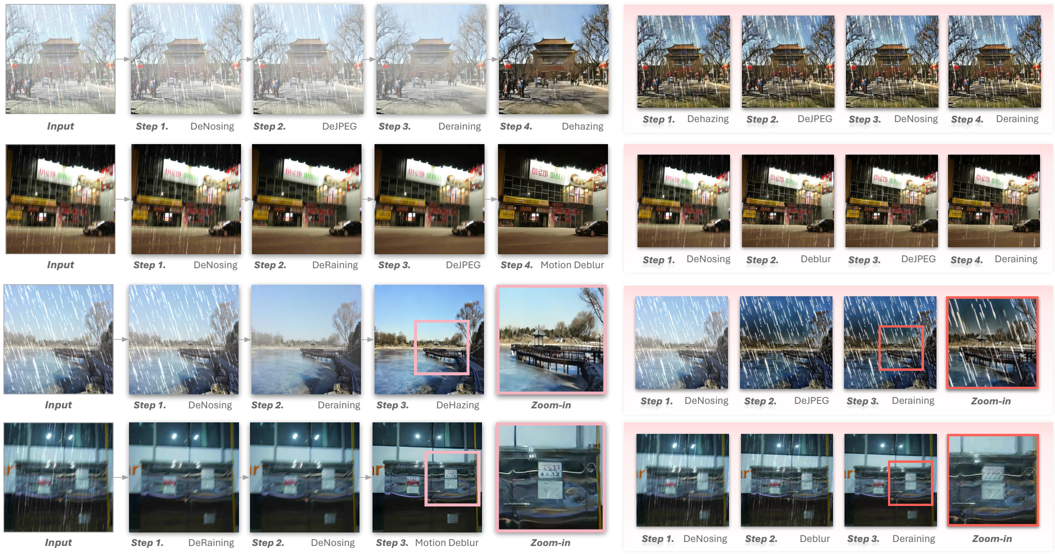


Figure 4: Illustrations of RestoreAgent's choices demonstrate that our approach predicts the correct task sequence. Images with a pink background show inappropriate decisions.

**Dataset and model tool setings.** To explore the feasibility of automating image restoration using multimodal models, we selected six distinct image restoration tasks: denoising, motion deblurring, deJPEG, deraining, dehazing, and low-light image enhancement. Each image in the dataset can exhibit up to four types of degradation. To validate the decision-making ability of the model when multiple models are available for a single task, we constructed three specialized models for the denoising task, and three models have different noise levels: low, medium and high noise. Similarly, for the deJPEG task, we developed models specifically designed to handle severe and mild JPEG compression artifacts. For the remaining tasks, each has a corresponding dedicated model. For the testing datasets, we assemble 200 images, mirroring the degradation types found in the training datasets, to facilitate evaluation. Detailed information is in the supplementary material.

### 4.2 Comparisons with Other Strategies

**Compared methods.** In this study, we conducted a comparative analysis of RestoreAgent against several alternative approaches:

- Random selection of both the task order and the models, assuming accurate determination of task types.
- Random task order, but models predicted by RestoreAgent.
- Random model selection, but task orders predicted by RestoreAgent.
- For all images, using the human expert's predefined order and models, assuming accurate task type determination.
- Human expert personally crafting a solution for each image, determining the task sequence and models for each task. This method represents the most common scenario in real-world applications, where a human decides how to restore an image on a case-by-case basis.

The human expert in this study has more than five years of research experience in low-level vision. Before crafting solutions, the expert familiarized themselves with each task degradation and the corresponding model's actual performance to ensure they could provide the best human-level solution.

**Results.** Table 1 reports the average metric results of our RestoreAgent and other decision-making approaches on seven different degradation combination datasets. As shown in Table 1, using a random order and model selection ranked lowest, achieving only a 34.7% performance rating among all possible strategies. By setting predefined sequences and models for image processing by human experts, traditional methods rank in the top 22.1% of all possible strategies. This demonstrates that experience-based predefined rules often used in practical applications are more effective than completely random strategies. Human experts making specific decisions for each test image can further improve upon predefined rules, increasing the ranking from 22.1% to 19.5%. This proves that using the same predefined rules to process all images is not optimal, while individualized decision-making for specific images can better enhance the effects. Then, the superior performance of our RestoreAgent (12.9%) over expert-based customization (19.5%) shows that automated and

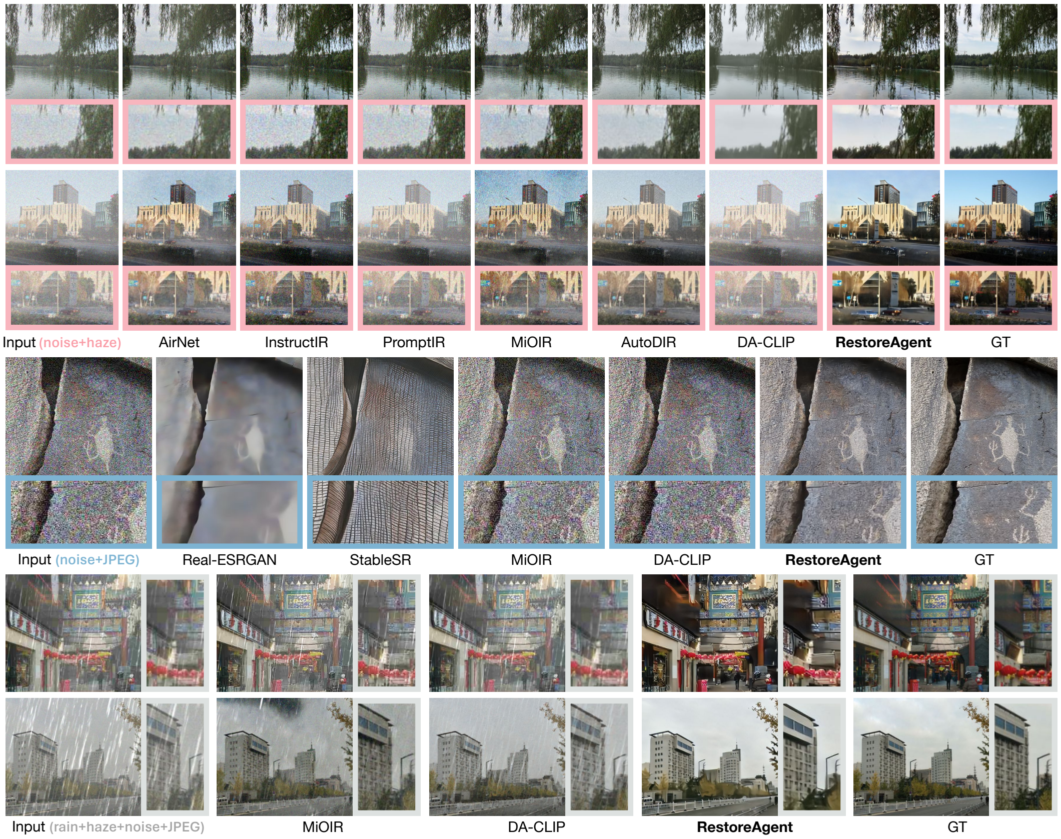


Figure 5: Visual comparisons with All-in-One Methods. To ensure a fair comparison, All-in-One methods are tested only on the degradation types and datasets they support. The all-in-one approach still lacks the ability to effectively handle images containing multiple types of degradation.

Table 2: Comparison of RestoreAgent with All-in-One methods for multi-degraded image restoration. We highlight best and second-best values for each metric.

| | noise + JPEG | | | | haze + noise | | | | rain + haze + noise | | | | rain + haze + noise + JPEG | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | DISTS ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ | DISTS ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ | DISTS ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ | DISTS ↓ |
| Real-ESRGAN [48] | 23.43 | 0.7242 | 0.3022 | 0.2106 | - | - | - | - | - | - | - | - | - | - | - | - |
| StableSR [47] | 17.61 | 0.4464 | 0.3705 | 0.2124 | - | - | - | - | - | - | - | - | - | - | - | - |
| AirNet [31] | - | - | - | - | 17.56 | 0.5897 | 0.5569 | 0.2964 | 18.22 | 0.6767 | 0.4314 | 0.2336 | - | - | - | - |
| PromptIR [40] | - | - | - | - | 16.13 | 0.5428 | 0.6696 | 0.3544 | 17.81 | 0.7099 | 0.4506 | 0.2317 | - | - | - | - |
| MiOIR [27] | 23.98 | 0.6961 | 0.3266 | 0.2325 | 15.79 | 0.4790 | 0.7118 | 0.3628 | 16.22 | 0.6388 | 0.4719 | 0.2771 | 13.80 | 0.6410 | 0.4875 | 0.2939 |
| InstructIR [14] | - | - | - | - | 17.36 | 0.4288 | 0.7696 | 0.3646 | 19.45 | 0.6897 | 0.3994 | 0.2170 | - | - | - | - |
| DA-CLIP [37] | 22.47 | 0.6128 | 0.3525 | 0.2287 | 16.98 | 0.7061 | 0.3901 | 0.2737 | 15.44 | 0.6011 | 0.4597 | 0.2754 | 15.30 | 0.6863 | 0.3871 | 0.2627 |
| AutoDIR [24] | - | - | - | - | 17.51 | 0.6942 | 0.4248 | 0.2444 | 19.22 | 0.7705 | 0.3043 | 0.1802 | - | - | - | - |
| **RestoreAgent** | 25.32 | 0.7806 | 0.2308 | 0.1958 | 20.47 | 0.8053 | 0.2193 | 0.1758 | 19.53 | 0.8237 | 0.2166 | 0.1638 | 19.72 | 0.7816 | 0.2741 | 0.1903 |

data-driven decision-making in our method clearly outperforms traditional and experience-based human expert judgments. This is because human experts from their own experience can not make precise judgments about the advantageous scenarios of all models and the order of task execution, especially when numerous tasks and models are involved.

### 4.3 Comparisons with All-in-One Methods

To demonstrate the limitations of all-in-one methods in handling multi-degraded images, we compared our approach with various types of all-in-one models. To ensure a fair comparison, tests were only conducted on degradation types and datasets that these all-in-one models were trained to support. Moreover, we repeatedly run the all-in-one model as many times as the number of degradation types of the test images to fully leverage its capabilities, thus ensuring a fair comparison. The results are shown in Figure 5 and Table 2. Our RestoreAgent achieved a significant leading advantage across all tested degradation combinations. For the degradation types commonly encountered in traditional image super-resolution, such as noise and JPEG compression artifacts, our approach significantly outperformed established methods like Real-ESRGAN and the sota SR method, StableSR. For a broader range of degradation types, our method retained a considerable advantage. Among these all-in-one approaches, InstructIR and AutoDIR face two major issues: manually predetermined or

|  | PSNR |  | SSIM |  | LPIPS |  | DISTS |  |
|---|---|---|---|---|---|---|---|---|
|  | Value | ranking (%) | Value | ranking (%) | Value | ranking (%) | Value | ranking (%) |
| Pre-defined Oder and Model | 22.38 | 25.4 | 0.7639 | 26.5 | 0.2644 | 25.2 | 0.1986 | 22.1 |
| Human Expert | 22.51 | 20.5 | 0.7634 | 22.6 | 0.2670 | 25.1 | 0.2014 | 23.8 |
| RestorAgent - balanced | 22.61 | 19.9 | 0.7700 | 21.4 | 0.2513 | 17.1 | 0.1890 | 13.8 |
| RestorAgent - PSNR | 22.72 | 13.9 | - | - | - | - | - | - |
| RestorAgent - SSIM | - | - | 0.7763 | 16.5 | - | - | - | - |
| RestorAgent - LPIPS | - | - | - | - | 0.2477 | 13.4 | - | - |
| RestorAgent - DISTS | - | - | - | - | - | - | 0.1875 | 12.9 |

Table 3: RestoreAgent possesses the flexibility to adapt to various optimization objectives, enabling the generation of decision-making results tailored to specific target criteria.

Table 4: Analysis of the effect of training data size. Our model shows strong performance with smaller datasets (7k), but increasing the data volume (23k) results in further enhanced outcomes.

|  | PSNR | SSIM | LPIPS | DISTS | balanced | ranking (%) |
|---|---|---|---|---|---|---|
| Random | 21.31 | 0.7139 | 0.3246 | 0.2241 | 1.92 | 34.7 |
| Human Expert | 22.51 | 0.7634 | 0.2670 | 0.2014 | 3.73 | 19.5 |
| **RestoreAgent** |  |  |  |  |  |  |
| - 7k | 22.63 | 0.7669 | 0.2568 | 0.1922 | 4.10 | 16.2 |
| - 14k | 22.57 | 0.7664 | 0.2528 | 0.1902 | 4.26 | 13.6 |
| - 23k | 22.61 | 0.7700 | 0.2513 | 0.1890 | 4.38 | 12.9 |

Table 5: Fast adaptation to new task (desnowing) in half an hour.

|  | balanced | ranking /64 |
|---|---|---|
| Random | 0.54 | 27.1 |
| Pre-defined Order and Model | 3.82 | 9.2 |
| Human Expert | 3.91 | 8.5 |
| **RestoreAgent** | 4.23 | 4.3 |

Table 6: Step-wise planning.

|  | balanced | ranking /64 |
|---|---|---|
| Human Expert | 5.42 | 21.2 |
| **RestoreAgent** | 6.35 | 5.7 |
| **RestoreAgent + Step-wise** | 6.38 | 4.5 |

randomly decided execution order, and using single model to address all types of degradations. These limitations often result in incomplete restoration, as depicted in Figure 5. These results underscore the limitations of all-in-one models, validating our initial hypothesis.

### 4.4 Adapting to Different Optimization Objectives

As discussed in the method, our proposed method can adapt to various optimization objectives, enabling the decision-making results tailored to specific target criteria. To verify it, we present the results of models trained with different individual metrics as the optimization objective in Table 3. The results indicate that when a model is trained with a single metric, the performance of the corresponding metric can be significantly improved compared to the balanced model. This showcases the adaptability and effectiveness of our method in catering to specific optimization goals.

### 4.5 Extending for New Tasks and Models

The proposed RestoreAgent demonstrates remarkable adaptability and extensibility, allowing for swift fine-tuning to accommodate new task types and incorporate additional models. To validate this capability, we introduced a new task, desnowing, along with its corresponding model. Building upon the RestoreAgent previously trained on six tasks, we performed rapid fine-tuning by integrating the desnowing task. Within thirty minutes, our model achieved exceptional performance on the new task type. As shown in Table 5, our approach quickly surpassed human expert-level proficiency on the new task and model. This validation underscores the practical value of our method, allowing efficient integration of additional tasks with minimal resource expenditure.

### 4.6 Step-wise Re-planning and Rollback

As mentioned in Section 3.2, RestoreAgent supports iterative decision-making with historical context awareness. It dynamically adjusts strategies during image restoration, reassessing image state after each step and rolling back if needed. As demonstrated in Table 6, we conducted experiments on a complex dataset incorporating four distinct types of image degradation: Motion Blur, Rain, Noise, and JPEG compression. Results show that while the single prediction approach performs well, iterative step-wise replanning further enhances restoration outcomes, allowing for precise control and mid-course corrections. The initial decision's performance is already strong, step-wise replanning thus offers incremental yet valuable improvements to an already effective process.

## 5 Conclusion

Our research identifies key factors in processing multi-degraded images, including task execution order, model selection, and the limitations of the all-in-one approach. Based on these insights, we present RestoreAgent, an agent model that makes intelligent processing decisions based on image degradation and user objectives. Experiments show that our pipeline outperforms the all-in-one method and surpasses human experts in decision-making performance.

**Acknowledgments.** This work is supported by the Guangzhou-HKUST(GZ) Joint Funding Program (No. 2023A03J0671), the Guangzhou Municipal Science and Technology Project (Grant No. 2024A04J4230), Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things(No.2023B1212010007), and the National Natural Science Foundation of China (Project No. 61902275).

## References

[1] Yuang Ai, Huaibo Huang, Xiaoqiang Zhou, Jiexiang Wang, and Ran He. Multimodal prompt perceiver: Empower adaptiveness, generalizability and fidelity for all-in-one image restoration. *arXiv preprint arXiv:2312.02918*, 2023.

[2] Yuanhao Cai, Hao Bian, Jing Lin, Haoqian Wang, Radu Timofte, and Yulun Zhang. Retinexformer: One-stage retinex-based transformer for low-light image enhancement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12504–12513, 2023.

[3] Haoyu Chen, Jinjin Gu, Yihao Liu, Salma Abdel Magid, Chao Dong, Qiong Wang, Hanspeter Pfister, and Lei Zhu. Masked image training for generalizable deep image denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1692–1703, 2023.

[4] Haoyu Chen, Wenbo Li, Jinjin Gu, Jingjing Ren, Haoze Sun, Xueyi Zou, Youliang Yan, Zhensong Zhang, and Lei Zhu. Low-res leads the way: Improving generalization for super-resolution by self-supervised learning. *CVPR*, 2024.

[5] Haoyu Chen, Jingjing Ren, Jinjin Gu, Hongtao Wu, Xuequan Lu, Haoming Cai, and Lei Zhu. Snow removal in video: A new dataset and a novel method. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13211–13222, October 2023.

[6] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*, 2023.

[7] Sixiang Chen, Tian Ye, Jinbin Bai, Erkang Chen, Jun Shi, and Lei Zhu. Sparse sampling transformer with uncertainty-driven ranking for unified removal of raindrops and rain streaks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13106–13117, 2023.

[8] Sixiang Chen, Tian Ye, Yun Liu, Jinbin Bai, Haoyu Chen, Yunlong Lin, Jun Shi, and Erkang Chen. Cplformer: Cross-scale prototype learning transformer for image snow removal. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, page 4228–4239, New York, NY, USA, 2023. Association for Computing Machinery.

[9] Sixiang Chen, Tian Ye, Yun Liu, and Erkang Chen. Snowformer: Context interaction transformer with scale-awareness for single image desnowing. *arXiv preprint arXiv:2208.09703*, 2022.

[10] Sixiang Chen, Tian Ye, Yun Liu, and Erkang Chen. Snowformer: Context interaction transformer with scale-awareness for single image desnowing. *arXiv preprint arXiv:2208.09703*, 2022.

[11] Xiang Chen, Hao Li, Mingqiang Li, and Jinshan Pan. Learning a sparse transformer network for effective image deraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5896–5905, 2023.

[12] Jun Cheng, Tao Liu, and Shan Tan. Score priors guided deep variational inference for unsupervised real-world single image denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12937–12948, 2023.

[13] Sung-Jin Cho, Seo-Won Ji, Jun-Pyo Hong, Seung-Won Jung, and Sung-Jea Ko. Rethinking coarse-to-fine approach in single image deblurring. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4641–4650, 2021.

[14] Marcos V Conde, Gregor Geigle, and Radu Timofte. High-quality image restoration following human instructions. *arXiv preprint arXiv:2401.16468*, 2024.

[15] XTuner Contributors. Xtuner: A toolkit for efficiently fine-tuning llm. `https://github.com/InternLM/xtuner`, 2023.

[16] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P. Simoncelli. Image quality assessment: Unifying structure and texture similarity. *CoRR*, abs/2004.07728, 2020.

[17] Jiangxin Dong, Jinshan Pan, Zhongbao Yang, and Jinhui Tang. Multi-scale residual low-pass filter network for image deblurring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12345–12354, 2023.

[18] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

[19] Xueyang Fu, Zheng-Jun Zha, Feng Wu, Xinghao Ding, and John Paisley. Jpeg artifacts reduction via deep convolutional sparse coding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[20] Jinjin Gu, Haoming Cai, Haoyu Chen, Xiaoxing Ye, Jimmy Ren, and Chao Dong. Image quality assessment for perceptual image restoration: A new dataset, benchmark and metric. *arXiv preprint arXiv:2011.15002*, 2020.

[21] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[22] Seo-Won Ji, Jeongmin Lee, Seung-Wook Kim, Jun-Pyo Hong, Seung-Jin Baek, Seung-Won Jung, and Sung-Jea Ko. Xydeblur: Divide and conquer for single image deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17421–17430, June 2022.

[23] Jiaxi Jiang, Kai Zhang, and Radu Timofte. Towards flexible blind jpeg artifacts removal. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4997–5006, 2021.

[24] Yitong Jiang, Zhaoyang Zhang, Tianfan Xue, and Jinwei Gu. Autodir: Automatic all-in-one image restoration with latent diffusion. *arXiv preprint arXiv:2310.10123*, 2023.

[25] Xiaoyu Jin, Yuan Shi, Bin Xia, and Wenming Yang. Llmra: Multi-modal large language model based restoration assistant. *arXiv preprint arXiv:2401.11401*, 2024.

[26] Gu Jinjin, Cai Haoming, Chen Haoyu, Ye Xiaoxing, Jimmy S Ren, and Dong Chao. Pipal: a large-scale image quality assessment dataset for perceptual image restoration. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 633–651. Springer, 2020.

[27] 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.

[28] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8183–8192, 2018.

[29] Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[30] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. *arXiv preprint arXiv:1803.04189*, 2018.

[31] 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.

[32] Ji Li, Weixi Wang, Yuesong Nan, and Hui Ji. Self-supervised blind motion deblurring with deep expectation maximization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13986–13996, June 2023.

[33] Zilong Li, Yiming Lei, Chenglong Ma, Junping Zhang, and Hongming Shan. Prompt-in-prompt learning for universal image restoration. *arXiv preprint arXiv:2312.05038*, 2023.

[34] Xinqi Lin, Jingwen He, Ziyan Chen, Zhaoyang Lyu, Ben Fei, Bo Dai, Wanli Ouyang, Yu Qiao, and Chao Dong. Diffbir: Towards blind image restoration with generative diffusion prior. *arXiv preprint arXiv:2308.15070*, 2023.

[35] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

[36] Lin Liu, Lingxi Xie, Xiaopeng Zhang, Shanxin Yuan, Xiangyu Chen, Wengang Zhou, Houqiang Li, and Qi Tian. Tape: Task-agnostic prior embedding for image restoration. In *European Conference on Computer Vision*, pages 447–464. Springer, 2022.

[37] Ziwei Luo, Fredrik K Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B Schön. Controlling vision-language models for universal image restoration. *arXiv preprint arXiv:2310.01018*, 2023.

[38] Dongwon Park, Byung Hyun Lee, and Se Young Chun. All-in-one image restoration for unknown degradations using adaptive discriminative filters for specific degradations. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5815–5824. IEEE, 2023.

[39] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023.

[40] Vaishnav Potlapalli, Syed Waqas Zamir, Salman Khan, and Fahad Shahbaz Khan. Promptir: Prompting for all-in-one blind image restoration. *arXiv preprint arXiv:2306.13090*, 2023.

[41] Xu Qin, Zhilin Wang, Yuanchao Bai, Xiaodong Xie, and Huizhu Jia. Ffa-net: Feature fusion attention network for single image dehazing. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11908–11915, 2020.

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[43] Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Erix Xing, Ming-Hsuan Yang, and Fahad S Khan. Glamm: Pixel grounding large multimodal model. *arXiv preprint arXiv:2311.03356*, 2023.

[44] Mengwei Ren, Mauricio Delbracio, Hossein Talebi, Guido Gerig, and Peyman Milanfar. Multiscale structure guided diffusion for image deblurring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10721–10733, 2023.

[45] Haoze Sun, Wenbo Li, Jianzhuang Liu, Haoyu Chen, Renjing Pei, Xueyi Zou, Youliang Yan, and Yujiu Yang. Coser: Bridging image and language for cognitive super-resolution. *arXiv preprint arXiv:2311.16512*, 2023.

[46] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[47] Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin CK Chan, and Chen Change Loy. Exploiting diffusion prior for real-world image super-resolution. *arXiv preprint arXiv:2305.07015*, 2023.

[48] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1905–1914, 2021.

[49] Zejin Wang, Jiazheng Liu, Guoqing Li, and Hua Han. Blind2unblind: Self-supervised image denoising with visible blind spots. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2027–2036, June 2022.

[50] Yanyan Wei, Zhao Zhang, Jiahuan Ren, Xiaogang Xu, Richang Hong, Yi Yang, Shuicheng Yan, and Meng Wang. Clarity chatgpt: An interactive and adaptive processing system for image restoration and enhancement. *arXiv preprint arXiv:2311.11695*, 2023.

[51] Jay Whang, Mauricio Delbracio, Hossein Talebi, Chitwan Saharia, Alexandros G. Dimakis, and Peyman Milanfar. Deblurring via stochastic refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16293–16303, June 2022.

[52] Haiyan Wu, Yanyun Qu, Shaohui Lin, Jian Zhou, Ruizhi Qiao, Zhizhong Zhang, Yuan Xie, and Lizhuang Ma. Contrastive learning for compact single image dehazing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10551–10560, 2021.

[53] Rongyuan Wu, Tao Yang, Lingchen Sun, Zhengqiang Zhang, Shuai Li, and Lei Zhang. Seesr: Towards semantics-aware real-world image super-resolution. *arXiv preprint arXiv:2311.16518*, 2023.

[54] Ruiqi Wu, Zhengpeng Duan, Chunle Guo, Zhi Chai, and Chongyi Li. Ridcp: Revitalizing real image dehazing via high-quality codebook priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[55] Xiaohe Wu, Ming Liu, Yue Cao, Dongwei Ren, and Wangmeng Zuo. Unpaired learning of deep image denoising. In *European conference on computer vision*, pages 352–368. Springer, 2020.

[56] Fanghua Yu, Jinjin Gu, Zheyuan Li, Jinfan Hu, Xiangtao Kong, Xintao Wang, Jingwen He, Yu Qiao, and Chao Dong. Scaling up to excellence: Practicing model scaling for photo-realistic image restoration in the wild. *arXiv preprint arXiv:2401.13627*, 2024.

[57] 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.

[58] 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.

[59] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017.

[60] 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.

[61] Yi Zhang, Dasong Li, Ka Lung Law, Xiaogang Wang, Hongwei Qin, and Hongsheng Li. Idr: Self-supervised image denoising via iterative data refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2098–2107, June 2022.

[62] Yihua Zhang, Pingzhi Li, Junyuan Hong, Jiaxiang Li, Yimeng Zhang, Wenqing Zheng, Pin-Yu Chen, Jason D Lee, Wotao Yin, Mingyi Hong, et al. Revisiting zeroth-order optimization for memory-efficient llm fine-tuning: A benchmark. *arXiv preprint arXiv:2402.11592*, 2024.

[63] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual non-local attention networks for image restoration. *arXiv preprint arXiv:1903.10082*, 2019.

# A Appendix / supplemental material

## A.1 Model Tool Setings

As shown in Table 7, for the tasks of denoising and deJPEG, as well as deraining, we employ Restormer [58] as our model. For dehazing, we utilize RIDCP [54], while for motion deblurring, we use DeblurGANv2 [28]. For desnowing, we implement Snowformer [10]. For low-light enhancement, we use Retinexformer [2]. It is noteworthy that the models we are using are not the latest state-of-the-art models, indicating that there is significant room for improvement in our models.

A crucial consideration in image restoration is the limited generalization capability of many current models, which often fail to maintain performance when faced with subtle variations in image degradation. This necessitates the selection of more robust models. For example, in our approach to denoising, we enhance model generalization by incorporating not only Gaussian noise but also random blur and other noise types during training. This strategy enables the model to address more complex degradation scenarios effectively.

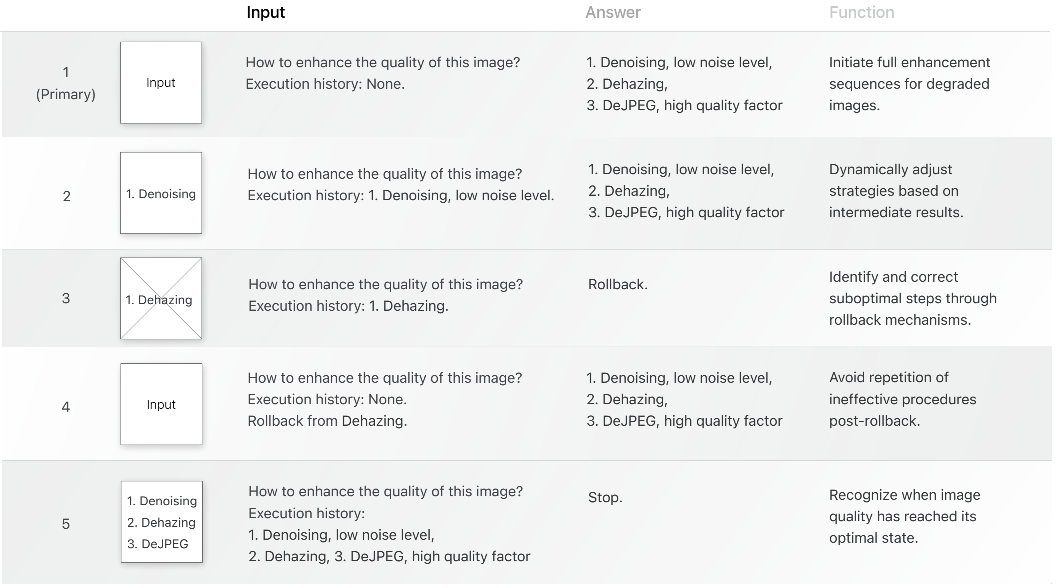
Table 7: Model tools for different restoration tasks.

| Task | Model Tools |
|---|---|
| Gaussian denosing | Restormer (trained on large noise level)<br>Restormer (trained on medium noise level)<br>Restormer (trained on small noise level) |
| DeJPEG | Restormer (trained on high quality factor)<br>Restormer (trained on low quality factor) |
| Dehazing | RIDCP [54] |
| Deraining | Restormer [58] |
| Motion deblurring | DeblurGANv2 [28] |
| Low-light enhancement | Retinexformer [2] |

Table 8: Testset details.

| Degradatio | Number of Images |
|---|---|
| Noise + JPEG | 50 |
| Noise + Low light | 30 |
| Motion Blur + Noise + JPEG | 30 |
| Rain + Noise + JPEG | 20 |
| Haze + Noise + JPEG | 30 |
| Haze + Rain + Noise + JPEG | 20 |
| Motion Blur + Rain + Noise + JPEG | 20 |
| **Total** | **200** |

## A.2 Dataset Construction Details



| | | Input | Answer | Function |
|---|---|---|---|---|
| 1<br>(Primary) | Input | How to enhance the quality of this image?<br>Execution history: None. | 1. Denoising, low noise level,<br>2. Dehazing,<br>3. DeJPEG, high quality factor | Initiate full enhancement sequences for degraded images. |
| 2 | 1. Denoising | How to enhance the quality of this image?<br>Execution history: 1. Denoising, low noise level. | 1. Denoising, low noise level,<br>2. Dehazing,<br>3. DeJPEG, high quality factor | Dynamically adjust strategies based on intermediate results. |
| 3 | 1. Dehazing | How to enhance the quality of this image?<br>Execution history: 1. Dehazing. | Rollback. | Identify and correct suboptimal steps through rollback mechanisms. |
| 4 | Input | How to enhance the quality of this image?<br>Execution history: None.<br>Rollback from Dehazing. | 1. Denoising, low noise level,<br>2. Dehazing,<br>3. DeJPEG, high quality factor | Avoid repetition of ineffective procedures post-rollback. |
| 5 | 1. Denoising<br>2. Dehazing<br>3. DeJPEG | How to enhance the quality of this image?<br>Execution history:<br>1. Denoising, low noise level,<br>2. Dehazing, 3. DeJPEG, high quality factor | Stop. | Recognize when image quality has reached its optimal state. |

Figure 6: Five scenarios for dataset construction and their corresponding examples.

Figure 6 illustrates 5 scenarios incorporated into our dataset, designed to enhance the versatility and robustness of the RestoreAgent model:

(1) Once we obtain a degraded image along with its corresponding optimal decision results, we can construct the primary part of our dataset. This part consists of degraded images in their original, unprocessed state. For these inputs, the RestoreAgent receives a prompt: "How to enhance the quality of this image? Execution history: None." This scenario trains the model to formulate comprehensive enhancement strategies from scratch, encompassing multiple restoration steps. This part of the data exceeds 23k pairs.

(2) To foster dynamic decision-making capabilities, we introduce a second category of training instances. Here, the input comprises partially processed images (e.g., after denoising) along with their execution history. This approach enables the RestoreAgent to adapt its predictions based on intermediate results, promoting a more flexible and context-aware enhancement process.

(3) The third scenario addresses situations where the model identifies suboptimal outcomes from a particular enhancement step. In such cases, the RestoreAgent is trained to output "Rollback," indicating the need to revert to a previous state and recalibrate its strategy. This feature is crucial for maintaining high-quality outputs and avoiding the propagation of errors through the enhancement pipeline. We select from erroneous paths (the decisions with the worst metric results) to construct this portion of the paired data, as the worst paths require a rollback.

(4) Following a rollback event, our fourth data category provides the model with information about the specific step that triggered the rollback. This guidance is essential in preventing the model from repeating ineffective procedures, thus streamlining the enhancement process and improving efficiency.

(5) The final scenario in our training regime represents fully processed images that require no further enhancement. In these instances, the RestoreAgent is trained to recognize optimal image quality and output "Stop", effectively terminating the enhancement sequence.

By incorporating these diverse scenarios, we aim to develop a highly adaptive and efficient image restoration system capable of addressing a wide array of real-world image degradation challenges. For computational efficiency, unless specifically mentioned otherwise, our default experiments are based on a single planning for the initial image rather than using iterative step-wise replanning.

### A.3 Testset details

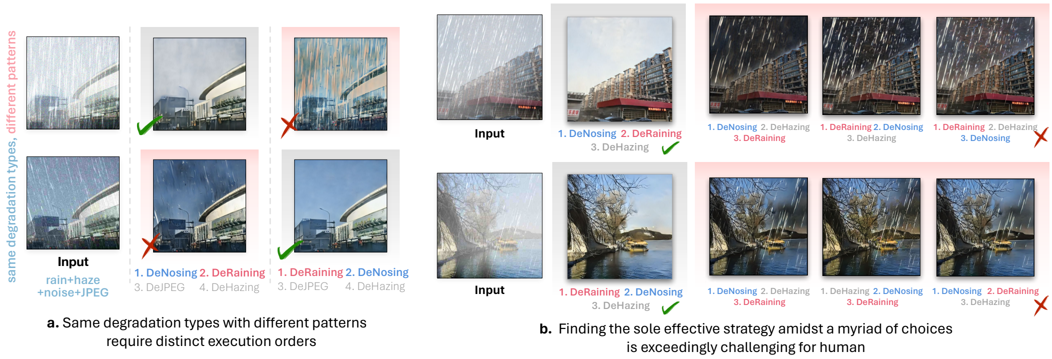
The specific details of our test set are presented in Table 8, which demonstrates our construction of various combinations of degradation types. Each image in the set contains a minimum of one and a maximum of four types of degradation, with the entire set comprising 200 images.

### A.4 Training Setups

In this study, we incorporate the CLIP pre-trained Vision Transformer (ViT-L/14) [42] as the image encoder to convert input images into visual tokens. For the language model, we utilize the Llama3-7B [46]. Despite their capabilities, pre-trained LLMs fail to provide accurate responses without dataset-specific fine-tuning. To address this, we adopt LoRA [21], a fine-tuning technique that efficiently modifies a limited number of parameters within the model. Following [21], we apply LoRA to adjust the projection layers in all self-attention modules of both the vision encoder and the LLM, thereby generating our RestoreAgent. We employ the Xtuner framework [15] to facilitate the training process. For our experimental setup, we configure the LoRA rank to 16. The RestoreAgent undergoes training across ten epochs on 4 NVIDIA RTX A100 GPUs, with a batch size of 32. We employ the Adam optimizer and a learning rate of 0.00002. The total duration of the training process approximates ten hours.

### A.5 Analysis

Figure 4 and 8 illustrate RestoreAgent's decision-making process and the importance of model selection, respectively. Figure 6 further demonstrates why human decision-making often yields suboptimal results in image restoration tasks. Figure 8a exemplifies the nuanced challenges in degradation assessment. Despite identical backgrounds and degradation types, subtle variations in degradation features lead to divergent optimal restoration sequences. For instance, the sequence



**a.** Same degradation types with different patterns require distinct execution orders

**b.** Finding the sole effective strategy amidst a myriad of choices is exceedingly challenging for human

Figure 7: Challenges in human expert decision-making. This figure illustrates the difficulty faced by human experts in discerning minute differences between degradation patterns, leading to suboptimal restoration strategies.

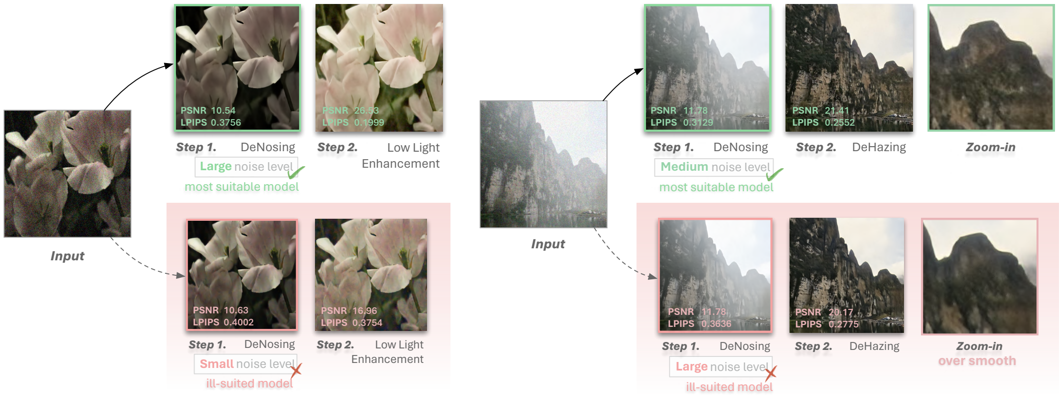


Figure 8: Examples of model decisions made by RestoreAgent. This figure demonstrates how choosing the appropriate model for a specific restoration task significantly affects the outcome quality. We present PSNR and LPIPS metrics for each image. Images with a pink background indicate examples of inappropriate decisions (zoom-in for better view).

"DeNoising → DeRaining → DeJPEG → DeHazing" proves effective for the upper row images but fails for the lower row. Conversely, the sequence "DeRaining → DeNoising → DeJPEG → DeHazing" yields optimal results for the lower row but is suboptimal for the upper row. This dichotomy underscores the difficulty human experts face in discerning minute degradation differences, thereby compromising effective decision-making.

The complexity of optimal restoration sequencing is further accentuated in Figure 8b. Here, we demonstrate scenarios where only one specific sequence among numerous permutations yields satisfactory results. This observation highlights the formidable challenge posed to human decision-makers in identifying the singular effective restoration pathway amidst a multitude of possibilities.

These findings collectively emphasize the superiority of automated, data-driven approaches in navigating the intricate landscape of image restoration. The RestoreAgent's ability to discern and adapt to subtle degradation variations surpasses human capabilities, particularly in scenarios where the optimal restoration sequence is non-intuitive and highly specific to individual image characteristics.

### A.6 Discussion

**Comparison with assistants with all-in-one models.** Assistants that employ unified models, such as LLMRA [25] and AutoDIR [24], attempt to handle diverse tasks, degradation patterns, and intensities using a single model. As discussed in Section 1.1, these all-in-one models face significant challenges, including restricted task scope and compromised performance, which greatly limit their effectiveness in real-world applications. Conversely, our method leverages various model experts to address specific situations, the upper bound of our pipeline is determined by the latest SOTA models, allowing us to maximally leverage the latest advancements in the field without being constrained by the limitations of an all-in-one model. Furthermore, as detailed in Section 4.4, our RestoreAgent exhibits high efficiency in incorporating new tasks and models, showcasing greater flexibility.

**Comparison with assistants with tool use.** Image restoration assistants that utilize tool libraries, such as Clarity ChatGPT [50] and RL-Restore [57]. Clarity ChatGPT merely identifies the degradation in images, follows a rigid execution strategy, lacking the ability to make dynamic decisions on task execution order and select the best model. As discussed in Section 1.2.1 and 1.2.2, an inappropriate task execution sequence and model selection can leading to lower performance in subsequent operations. RL-Restore, on the other hand, uses reinforcement learning for sequence decision-making and model selection. However, its task definition is overly simplistic, limited to three degradation types (noise, blur, and JPEG) with a narrow degradation range. Also, training reinforcement learning-based methods is more challenging and may result in lower precision, making it difficult to achieve high performance in complex and varied scenarios. Conversely, the integration of a comprehensive task definition with advanced multimodal models allows our method to effectively manage various degradation types and intensities. This adaptability enhances its efficacy, positioning our approach as a promising solution for image restoration tasks.

### A.7 Alation Study

**Training data amount.** To investigate the effect of training data volume on our method, we evaluated the performance of the RestoreAgent model trained on datasets consisting of 7,000, 14,000, and 23,000 data pairs; see Table 4 The results demonstrate that even with the smallest dataset of 7k pairs, our RestoreAgent achieves superior performance over both random and human expert benchmarks. More notably, the training data volume increasing from 7k to 14k incurs a substantial performance improvement with the ranking percentage decreasing from 16.2% to 13.6%. With 23k data pairs, the performance further improves, achieving a ranking percentage of 12.9%. This indicates that using more training data boosts our RestoreAgent model. These findings emphasize the robustness of our approach, demonstrating that while larger datasets do enhance performance, our model already provides significant benefits even with relatively smaller datasets.

### A.8 Limitation and Future Work

The primary limitation of our study is the confined scope of models and tasks examined. While our research offers valuable insights into RestoreAgent's performance across several degradation scenarios, it does not encompass the full spectrum of restoration models or image degradation tasks currently available.

Another limitation pertains to the limited generalization capability of current image restoration models. These models often exhibit a notable decrease in performance or fail to respond adequately when faced with even minor variations in image degradation patterns. This limitation greatly narrows our selection of model tools, requiring us to choose more robust and generalizable model tools. The challenge underscores a critical need in the field of image restoration: future models must go beyond simply overfitting training data. Rather, they should exhibit better generalization and increased efficiency in handling real-world degradation cases.

Our future work will focus on significantly expanding the range of image restoration models incorporated into our multimodal large language model. This expansion aims to enhance RestoreAgent's capabilities across a broader scope of restoration tasks and degradation types. By integrating a more diverse set of state-of-the-art models, we seek to create a more comprehensive and versatile restoration framework.

## 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: Yes, the main claims made in the abstract and introduction accurately reflect the paper's contributions

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitations of the work in the supplemental material section.

   Guidelines:

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

3. **Theory Assumptions and Proofs**

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

   Answer: [Yes]

Justification: Yes, for each theoretical result, we provide the full set of assumptions and a complete proof.

Guidelines:

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

4. **Experimental Result Reproducibility**

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

   Answer: [Yes]

   Justification: Yes, our method is easy to be reproduced, and we provide all information.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We use public code and data.

Guidelines:

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

6. **Experimental Setting/Details**

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

   Answer: [Yes]

   Justification: Yes, we give all the details.

   Guidelines:

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

7. **Experiment Statistical Significance**

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

   Answer: [Yes]

   Justification: See Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

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

   Answer: [Yes]

   Justification: See Section 4.

   Guidelines:

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

9. **Code Of Ethics**

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

   Answer: [Yes]

   Justification: We conform, in every respect, to the NeurIPS Code of Ethics as outlined at the provided https://neurips.cc/public/EthicsGuidelines.

   Guidelines:

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

10. **Broader Impacts**

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

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The creators or original owners of assets used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected.

    Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

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

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

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

14. **Crowdsourcing and Research with Human Subjects**

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

    Answer: [Yes]

    Justification: See supplemental materia.

    Guidelines:

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

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

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

    Answer: [Yes]

    Justification: There are no potential risks.

    Guidelines:

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