TIR-Agent: Training an Explorative and Efficient Agent for Image Restoration

Guoli Jia[1⋆], Yisheng Zhang[1⋆], Haote Hu[1⋆], Shanxu Zhao[1], Kaikai Zhao[1,2], Long Sun[3], Xinwei Long[1], Kai Tian[1,4], Che Jiang[1,4], Zhaoxiang Liu[2], Kai Wang[2], Shiguo Lian[2], Kaiyan Zhang[1,4†], and Bowen Zhou[1,5†]

[1] Tsinghua University, Beijing, China
[2] Data Science & Artificial Intelligence Research Institute, China Unicom, Beijing, China
[3] Hunan University, Changsha, China
[4] Frontis.AI, Beijing, China
[5] Shanghai Artificial Intelligence Laboratory, Shanghai, China

Abstract. Vision-language agents that orchestrate specialized tools for image restoration (IR) have emerged as a promising method, yet most existing frameworks operate in a training-free manner. They rely on heuristic task scheduling and exhaustive tool traversal, resulting in sub-optimal restoration paths and prohibitive computational cost. We argue that the core bottleneck lies in the absence of a learned policy to make decision, as a vision-language model cannot efficiently handle degradation-aware task ordering and tool composition. To this end, we propose TIR-Agent, a trainable image restoration agent that performs a direct tool-calling policy through a two-stage training pipeline of supervised fine-tuning (SFT) followed by reinforcement learning (RL). Two key designs underpin effective RL training: (i) a random perturbation strategy applied to the SFT data, which broadens the policy's exploration over task schedules and tool compositions, and (ii) a multi-dimensional adaptive reward mechanism that dynamically re-weights heterogeneous image quality metrics to mitigate reward hacking. To support high-throughput, asynchronous GPU-based tool invocation during training, we further develop a globally shared model-call pool. Experiments on both in-domain and out-of-domain degradations show that TIR-Agent outperforms 12 baselines, including 6 all-in-one models, 3 training-free agents, and 3 proprietary models, and achieves over 2.5× inference speedup by eliminating redundant tool executions.

1 Introduction

Images captured in real-world scenarios inevitably suffer from composite degradations, *i.e.*, mixtures of noise, low light, compression artifacts, and more [26,61].

⋆: Equal contribution. Guoli Jia (exped1230@gmail.com) leads the project. †: Corresponding authors.

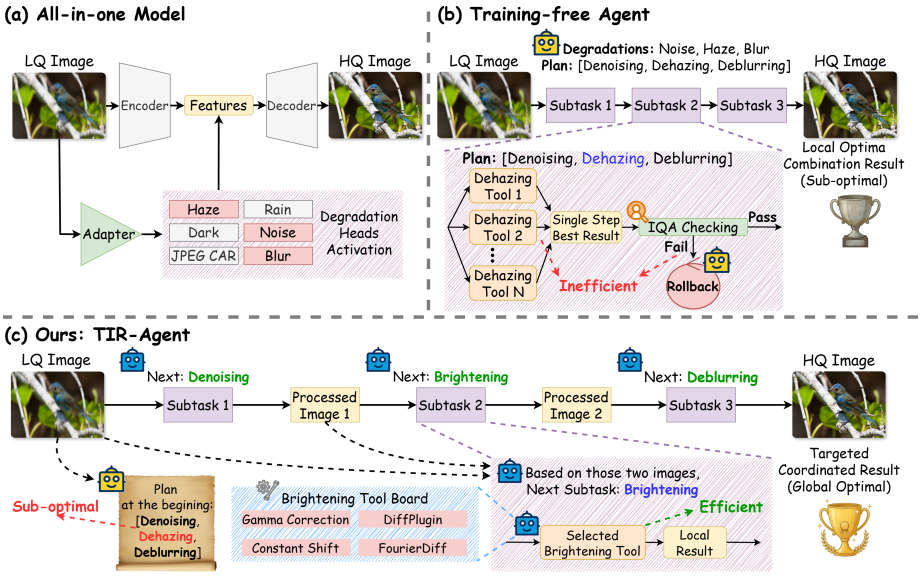


Fig. 1: Comparison of restoration paradigms. (a) All-in-one model: Uses a single network with specialized heads to inject various degradation features. **(b) Training-free Agent:** Uses VLMs for static task planning, executing sub-tasks sequentially via exhaustive tool traversal. **(c) TIR-Agent (Ours)**: A trainable agent that dynamically decides the next sub-task based on current results and learns to directly select the optimal tool, avoiding redundant execution.

This compositional nature makes the image restoration (IR) a challenging sequential decision-making problem that goes well beyond the scope of previous single degradation restoration network [7, 27].

Two dominant paradigms have been developed to address composite degradations. All-in-one models [22, 26, 38] employ a single network to jointly handle multiple degradations within one forward pass. While conceptually simple, sharing parameters across diverse degradations forces the model to trade off capacity, resulting in degraded performance on under-represented or long-tail degradations [16, 60]. Agent-based frameworks [61, 62] take a different approach. They treat pre-trained specialist restoration models as independent tools and leverage a vision-language model (VLM) [54, 61] to dynamically select and compose them. This modular design decouples capability from capacity, offering superior scalability and adaptability to unseen degradation combinations.

Despite their promise, current agent-based frameworks almost exclusively operate in a training-free setting [4, 6, 21, 25, 61, 62], which intrinsically faces three limitations: (i) **Sub-optimal task scheduling**. Without learning from experience, training-free agents rely on large language model (LLM)-based reasoning or fixed rules to determine the execution order of IR tasks [4, 21, 25]. Such static strategies fail to capture the complex, input-dependent interactions

between degradations. For example, the optimal ordering of denoising and dehazing may vary substantially across images. (ii) **Inefficient tool selection**. Given an IR task (*e.g.*, denoising), training-free agents typically exhaustively execute all candidate tools, such as X-Restormer [9], Restormer [56], and NAFNet [7], then select the best result [61, 62]. This enumeration scales linearly with the tool size, making inference prohibitively expensive. (iii) **Absence of reference-guided optimization**. Operating purely at inference time, training-free agents can only rely on no-reference (NR) image quality metrics [6, 34] to compare results. Without access to full-reference (FR) supervision during the learning phase, the fidelity of restored images is inherently limited.

Recent advances in agentic reinforcement learning (RL) have demonstrated that training LLM-based agents [32] to interact with tools, rather than relying on static prompting, leads to significantly more robust and efficient policies [5, 14]. A key insight from this emerging paradigm is that the agent's tool-use strategy should itself be a learned behavior, optimized through trial-and-error interaction based on the feedback. IR has well-defined reward signals that can be computed from FR and NR image quality metrics for training. Yet, to our knowledge, no prior work has systematically applied agentic RL to train a vision-language agent for IR.

We present TIR-Agent, a trainable image restoration agent that learns to directly invoke the suitable tool at each step, eliminating the need for exhaustive search. TIR-Agent is trained through a two-stage pipeline. In the supervised fine-tuning (SFT) stage, the VLM learns a base policy from demonstration data. To prevent the policy from converging to narrow, sub-optimal patterns, we apply a exploration-driven perturbation (EDP) strategy that injects controlled perturbation into the SFT trajectories, encouraging the model to explore a broader space of task schedules and tool compositions. In the subsequent RL stage, the agent refines its policy by interacting with the FR and NR feedback. We observe that naively combining multiple image quality metrics as a reward signal leads to reward hacking, *i.e.*, the agent exploits partial metrics at the expense of others. To address this issue, we propose a multi-dimensional adaptive reward (MAR) mechanism that dynamically adjusts the weight of each metric based on training progress, guiding the agent toward globally balanced optimization. To enable efficient RL training with GPU-intensive visual models, we further develop a distributed, asynchronous infrastructure centered on a globally shared model-call pool (MC-Pool), which supports high-frequency tool invocation across parallel training workers. Our contributions are summarized as follows:

- **Technical Contributions**. We introduce EDP that enhances exploration capability, enabling more effective policy refinement during RL. We further propose MAR that dynamically balances heterogeneous quality metrics, effectively mitigating reward hacking in multi-objective optimization.
- **Engineering Contribution**. We design a globally shared MC-Pool that supports distributed, high-frequency, and asynchronous GPU-based tool invocation, providing reliable infrastructure for training IR agents at scale.

- **SOTA Performance**. TIR-Agent achieves over 2.5× inference speedup compared to training-free agents, and outperforms 12 baselines across both in-domain and out-of-domain settings.

2 Related Work

2.1 Image Restoration

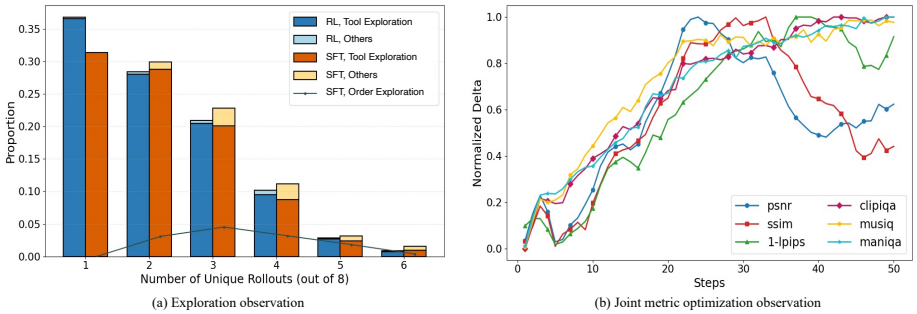
As a fundamental research in computer vision, IR has been widely deployed in professional software and mobile ecosystems, such as Adobe Photoshop's Neural Filters [20] and Topaz Photo AI [24]. The learning policy has evolved through three stages: *restoring specific degradations*, *unified models for unknown composite degradations*, *agents to organize specialized models*.

Early IR methods adopt a task-specific paradigm, where models are trained for a predefined degradation. Network architectures have evolved from CNN-based models [12, 28, 58] to GAN-based networks [49], and Transformer-based architectures such as SwinIR [27] and Uformer [50]. The models are supported by a unified framework BasicSR [48]. Later, motivated by the fact that most images often suffer from unknown and combined degradations, all-in-one restoration models are trained to handle multiple corruption types. Mainstream approaches adopt dedicated strategies, such as contrastive learning [26], degradation classification [38] for feature differentiation, learnable prompts [36], adapters [53], query injections [40], and data scheduling [8]. Besides, recent agent frameworks leverage VLM to assess degradation types [61] and organize specialized tools [4, 25]. These systems typically rely on planning mechanisms to determine the order of restoration subtasks, ranging from GPT-based selection [61], prior knowledge-based staging [21], to greedy strategies [60]. However, current tool selection is inefficient due to exhaustive traversal [6] and often yields suboptimal performance when relying solely on LLM-based reasoning [60].

To overcome these challenges, we investigate training the image restoration agent. A closely related work is SimpleCall [34], which introduces a lightweight multilayer perceptron (MLP)-based IR agent. However, its MLP-based framework lacks context-aware reasoning, *limiting its ability to determine whether and when to invoke super-resolution tools*. With improved tool invocation and training strategies, TIR-Agent achieves superior efficiency and multi-metric performance.

2.2 Vision-Language Agent

Recently, increasing efforts have focused on developing vision-language agent frameworks for diverse visual tasks through tool composition. Representative frameworks such as Visual Programming [18] and ViperGPT [44] translate queries into executable programs. Similarly, Visual ChatGPT [51] and HuggingGPT [42] utilize VLM-coordinated planning-execution pipelines to integrate external visual models and learned value functions [3] for grounded decision making.



(a) Exploration observation (b) Joint metric optimization observation

Fig. 2: Observation of trajectory diversity, tool selection distribution, and optimization trend. (a) Statistics of the 8 rollouts per sample, including trajectory diversity (distinct trajectories), order diversity (same tasks with different execution orders, line), and model (tool) diversity (identical task sequences, different tool sequences). (b) Convergence trends of evaluation metrics during 50 RL training steps.

To enhance multimodal reasoning capability [30, 33], recent studies employ thinking with image paradigm. These methods employ interleaved chain-of-thought [19,59], curiosity-driven reward designs [43], long-horizon reasoning [57], and adaptive tool invocation [47] for reasoning. Notably, recent studies have demonstrated that joint text-vision pre-training [46] significantly enhances multimodal reasoning capability. Besides, a wider range of tools is introduced to encourage explicit exploration of visual evidence [13, 17]. Considering that real images often suffer from diverse degradations [45], we investigate the agentic framework.

3 Methodology

We present TIR-Agent, a trainable image restoration agent built upon Qwen3-VL-8B-Instruct [2]. Given a degraded low-quality image I^L, TIR-Agent iteratively perceives the current image state sequence, selects a restoration task (*e.g.*, denoising, deblurring), and invokes a specialist tool to execute it. As illustrated in Fig. 3, the agent is trained in two stages, where SFT learns a base tool-calling policy from demonstration trajectories, followed by RL to refine this policy through trial-and-error interaction with the FR and NR feedback.

We formulate IR as a finite-horizon decision process. At each step k, the agent observes the image state sequence $\{s_0, s_1, \cdots, s_k\}$ ($s_0 = I^L$), and takes an action $a_k = (\tau_k, m_k)$, where τ_k denotes the selected restoration task and m_k the specialist model. Applying tool m_k yields the next state $s_{k+1} = m_k(s_k)$. The agent's policy $\pi_\theta(a_k \mid s_k, h_k)$ is parameterized by the VLM, where $h_k = \{(a_j, s_j)\}_{j<k}$ is the interaction history. The training objective is to learn π_θ that maximizes the expected image quality of the final restored output.

Since the pretrained VLM lacks image quality perception and restoration tool selection capabilities, we construct a demonstration dataset $\mathcal{S} = \{< I^L, I^H, \mathcal{T} >$

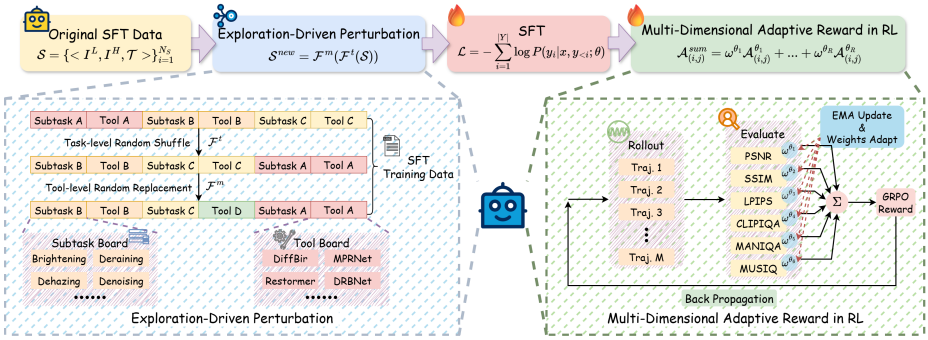


Fig. 3: Training pipeline of TIR-Agent. The process begins with **SFT Data Generation**, followed by **Exploration-Driven Perturbation** to enhance data diversity via shuffled task sequences and tool substitutions. Following **SFT**, the agent undergoes **RL with a Multi-Dimensional Adaptive Reward** mechanism, which dynamically balances metric weights based on their EMA trends to optimize the restoration policy.

$\}_{i=1}^{N_S}$. Each trajectory $\mathcal{T}_i = \{(\tau_k, m_k)\}_{k=1}^{K_i}$ records a sequence of task-tool decisions. Trajectories are generated using training-free pipeline [62].

3.1 Observations

Before presenting our technical designs, we provide empirical observations from the vanilla SFT+RL setting that reveal two primary challenges in training an IR agent. These observations directly motivate our proposed exploration-driven perturbation (EDP, §3.2) and multi-dimensional adaptive reward (MAR, §3.3).
Observation 1: Limited exploration capability. As shown in Fig. 2, during RL training, more than 35% of the trajectories are identical, and a similar phenomenon is observed for SFT model. Moreover, as shown in Fig. 2(a), among different trajectories, only about 10% differ in the task execution order, while trajectories with the same tasks but different tool selections are even fewer.
Observation 2: Difficulty in joint multi-metric optimization. Image quality assessment inherently involves multiple complementary metrics (*e.g.*, PSNR, SSIM, LPIPS). When naively combining them as a scalar reward, we observe that the agent tends to exploit partial metrics at the expense of others. This occurs because heterogeneous metrics have different scales, sensitivities, and optimization landscapes, making uniform weighting unstable as training progresses.

3.2 Exploration-Driven Perturbation

To enhance exploration capability, we apply random perturbation to the SFT data to increase the diversity of task scheduling and tool invocation after fine-tuning. The function for task scheduling, denoted as $\mathcal{F}^t$, is defined as:

$$\mathcal{F}^t(\mathcal{S}) = \mathcal{S} \cup \mathcal{G}(\mathcal{S}'),\ \ \mathcal{S}' = \mathbb{I}^{\alpha^t}(\mathcal{S}), \tag{1}$$

where $\mathcal{S}' \subseteq \mathcal{S}$ is obtained by independently sampling elements in $\mathcal{S}$ with a probability $\alpha^t \in [0,1]$. For each sampled instance $\mathcal{S}'_i =< I_i^L, I_i^H, \mathcal{T}_i >$ in $\mathcal{S}'$, we apply perturbation operator $\mathcal{G}(\cdot)$ as:

$$\mathcal{G}(\mathcal{S}_i) =< I_i^L, I_i^H, \phi(\mathcal{T}_i) >, \tag{2}$$

where $\phi(\cdot)$ denotes a random permutation over the order of the trajectory. Next, we illustrate the tool-level perturbation. For a given restoration task t, let $\mathcal{P}(m|t)$ denote the original tool selection distribution over candidate tools m. To encourage exploration, we leverage function $\mathcal{F}^m$ to construct a perturbed distribution $\mathcal{P}^m(m|t)$ as:

$$\mathcal{P}^m(m|t) = (1-\alpha^m)\mathcal{P}(m|t) + \alpha^m U(m|t), \tag{3}$$

where $U(m|t)$ denotes a uniform distribution over all valid tools for task t, and $\alpha^m \in [0,1]$ is the perturbation probability. This formulation interpolates between the original tool preference and uniform exploration, preventing the policy from collapsing to deterministic tool selection. Finally, combining task-level and tool-level perturbations, the overall dataset $\mathcal{S}^{new}$ for SFT is defined as:

$$\mathcal{S}^{new} = \mathcal{F}^m(\mathcal{F}^t(\mathcal{S})). \tag{4}$$

3.3 Multi-Dimensional Adaptive Reward

Image quality assessment inherently involves multiple metrics, which are often difficult to optimize jointly [31]. To address this issue, we propose a multi-dimensional adaptive reward mechanism that dynamically adjusts the optimization weight of each metric, encouraging the model to focus on the most under-optimized objective during training.

Given a batch of samples, taking metric θ_1 as an example, we measure its relative performance change by comparing the current batch reward r^{θ_1} with its exponential moving average (EMA) $\overline{r^{\theta_1}}$. Specifically, we define the clipped deviation score as:

$$\hat{\omega}^{\theta_1} = 1 - \text{clip}\left(\frac{r^{\theta_1} - \overline{r^{\theta_1}}}{\overline{r^{\theta_1}}}, -\epsilon, \epsilon\right), \tag{5}$$

where the clipping operation limits extreme fluctuations and stabilizes training. When the current reward drops below its EMA, the weight increases, encouraging the policy to allocate more optimization effort to this metric. The EMA is updated via:

$$\overline{r^{\theta_1}} = (1-\beta)r^{\theta_1} + \beta\overline{r^{\theta_1}}, \tag{6}$$

which provides a smoothed historical performance and reduces sensitivity to batch noise. The final metric weights are normalized via the softmax function:

$$\omega^{\theta_1} = \text{softmax}(\mathcal{W}),\ \mathcal{W} = \{\omega^{\theta_1}, ..., \omega^{\theta_R}\}. \tag{7}$$

Table 1: Quantitative comparison of multiple-degradation IR tasks on three subsets (Group A, B, and C) from the MiO100 dataset against previous sota methods. The top three performances of each metric are marked in **bold**, underline, *italic*, respectively.

Dataset	Type	Method	PSNR↑	SSIM↑	LPIPS↓	MANIQA↑	CLIPIQA↑	MUSIQ↑	Time(s)↓
Group A	All-in-One	AirNet [26]	19.13	0.6019	0.4283	0.2581	0.3930	42.46	1.75
		PromptIR [39]	20.06	0.6088	0.4127	0.2633	0.4013	42.62	1.26
		MiOIR [23]	20.84	0.6558	0.3715	0.2451	0.3933	47.82	0.69
		DA-CLIP [35]	19.58	0.6032	0.4266	0.2418	0.4139	42.51	44.87
		InstructIR [10]	18.03	0.5751	0.4429	0.2660	0.3528	45.77	1.45
		AutoDIR [22]	19.64	0.6286	0.3967	0.2500	0.3767	47.01	128.64
	Agent	AgenticIR [61]	*21.04*	0.6818	0.3148	0.3071	0.4474	56.88	184.57
		MAIR [21]	21.02	0.6715	0.2963	*0.3330*	*0.4751*	*59.19*	-
		4KAgent [62]	21.48	*0.6720*	*0.3019*	0.3748	**0.5544**	63.19	333.14
		TIR-Agent	**22.07**	**0.6935**	**0.2874**	**0.3907**	0.5454	**63.69**	**61.58**
Group B	All-in-One	AirNet [26]	19.31	0.6567	0.3670	0.2882	0.4274	47.88	1.83
		PromptIR [39]	20.47	0.6704	0.3370	0.2893	0.4289	48.10	1.30
		MiOIR [23]	20.56	0.6905	0.3243	0.2638	0.4330	51.87	1.09
		DA-CLIP [35]	18.56	0.5946	0.4405	0.2435	0.4154	43.70	47.41
		InstructIR [10]	18.34	0.6235	0.4072	0.3022	0.3790	50.94	1.50
		AutoDIR [22]	19.90	0.6643	0.3542	0.2534	0.3986	49.64	174.48
	Agent	AgenticIR [61]	20.55	0.7009	0.3072	0.3204	0.4648	57.57	201.25
		MAIR [21]	*20.92*	*0.7004*	0.2788	*0.3544*	*0.5084*	*60.98*	-
		4KAgent [62]	20.95	0.6727	*0.3017*	0.3734	**0.5505**	62.69	409.49
		TIR-Agent	**22.80**	**0.7340**	**0.2725**	**0.3997**	0.5488	**65.24**	**71.09**
Group C	All-in-One	AirNet [26]	17.95	0.5145	0.5782	0.1854	0.3113	30.12	2.07
		PromptIR [39]	18.51	0.5166	0.5756	0.1906	0.3104	29.71	1.36
		MiOIR [23]	15.63	0.4896	0.5376	0.1717	0.2891	37.95	2.06
		DA-CLIP [35]	18.53	0.5320	0.5335	0.1916	0.3476	33.87	59.64
		InstructIR [10]	17.09	0.5135	0.5582	0.1732	0.2537	33.69	1.90
		AutoDIR [22]	18.61	0.5443	0.5019	0.2045	0.2939	37.86	240.34
	Agent	AgenticIR [61]	18.82	0.5474	0.4493	0.2698	0.3948	48.68	249.70
		MAIR [21]	*19.42*	*0.5544*	0.4142	*0.2798*	*0.4239*	*51.36*	-
		4KAgent [62]	**19.77**	0.5629	*0.4271*	0.3545	0.5233	55.56	555.12
		TIR-Agent	19.53	**0.5643**	**0.4120**	**0.3739**	**0.5495**	**56.93**	**98.92**

Next, inspired by [29], we compute the advantage for each metric in a decoupled manner before aggregation:

$$\mathcal{A}^{\theta_1}_{(i,j)} = \frac{r^{\theta_1}_{(i,j)} - \text{mean}\{r^{\theta_1}_{(i,1)}, ..., r^{\theta_1}_{(i,g)}\}}{\text{std}\{r^{\theta_1}_{(i,1)}, ..., r^{\theta_1}_{(i,g)}\} + \epsilon}, \tag{8}$$

where i indexes the input samples and j indexes the j-th rollout generated, with g denoting the number of rollouts in each group. The decoupled advantages eliminate scale discrepancies, and the normalized advantages are subsequently combined via weighted summation as follows:

$$\mathcal{A}^{sum}_{(i,j)} = \omega^{\theta_1}\mathcal{A}^{\theta_1}_{(i,j)} + ... + \omega^{\theta_R}\mathcal{A}^{\theta_R}_{(i,j)}. \tag{9}$$

3.4 Globally Shared MC-Pool

Beyond algorithmic improvements, we develop a globally shared MC-Pool to support high-concurrency, high-frequency, GPU-based restoration model invocation during RL training. During agentic RL training, the sampling is asynchronous, each step may trigger up to $b \times g$ concurrent tool invocations, where b denotes

Table 2: Quantitative comparison of multiple-degradation IR tasks on three subsets (Group A, B, and C) from the MiO100 dataset against proprietary models.

Dataset	Method	PSNR↑	SSIM↑	LPIPS↓	MANIQA↑	CLIPIQA↑	MUSIQ↑	Time(s)↓
Group A	GPT-5.2 [37]	21.33	0.6620	0.3639	0.3385	0.4870	*57.45*	101.32
	Claude Opus 4.5 [1]	*21.47*	0.6999	*0.3264*	0.3073	0.4538	55.89	97.71
	Gemini 3.0 Pro [15]	**22.23**	**0.7085**	0.3151	*0.3149*	*0.4573*	57.95	99.59
	TIR-Agent	22.07	*0.6935*	**0.2874**	**0.3907**	**0.5454**	**63.69**	**61.58**
Group B	GPT-5.2 [37]	*20.84*	0.6627	0.3557	0.3369	0.4863	58.27	110.65
	Claude Opus 4.5 [1]	20.46	0.7051	0.3145	0.3189	0.4643	56.24	112.52
	Gemini 3.0 Pro [15]	21.03	*0.6927*	*0.3238*	*0.3225*	0.4764	*57.82*	109.83
	TIR-Agent	**22.80**	**0.7340**	**0.2725**	**0.3997**	**0.5488**	**65.24**	**71.09**
Group C	GPT-5.2 [37]	*19.53*	*0.5759*	0.4441	0.3218	0.4604	53.09	133.44
	Claude Opus 4.5 [1]	20.27	0.6043	*0.4228*	*0.2660*	*0.4132*	48.12	137.40
	Gemini 3.0 Pro [15]	**20.38**	**0.6118**	0.4150	0.2577	0.3973	*48.62*	148.51
	TIR-Agent	*19.53*	0.5643	**0.4120**	**0.3739**	**0.5495**	**56.93**	**98.92**

batch size and g is the number of a group of rollouts per sample. Such large-scale concurrent execution leads to severe GPU memory cost and unpredictable runtime failures if unmanaged.

To address this issue, we develop a globally-shared pool to manage the resource set $\mathcal{M} = \{M_1, M_2, ..., M_N\}$, where each M represents an independent MCP service instance bound to a specific GPU, which is deployed in a distributed manner. Tool invocation requests are assigned to available instances through a locking-based allocation operator, which guarantees mutual exclusion during execution and releases the resource upon completion. Further, we develop a retry mechanism (up to three attempts) to improve robustness against uncontrollable communication failures. Overall, the MC-Pool provides a stable and scalable execution infrastructure that enables controlled high-concurrency scheduling and GPU-based restoration during RL training.

4 Experiments

4.1 Implementation Details

We adopt Qwen3-VL-8B-Instruct [2] as the VLM and train it in two stages: SFT and RL. In the SFT stage, the model is fully fine-tuned for 3 epochs on 8 NVIDIA A100 GPUs, with a learning rate of 1e-5. The batch size is 1 with gradient accumulation of 4, and a cosine learning rate scheduler is applied. In the RL stage, we employ 8 NVIDIA A100 GPUs to train the VLM, with a batch size of 64, 8 rollouts per sample, and a learning rate of 5e-6. The maximum number of parallel rollouts is capped at 128 to limit memory overhead. In addition, we deploy 32 NVIDIA RTX 3090 GPUs as the MC-Pool to serve restoration model inference during rollout sampling.

4.2 Datasets and Evaluation

In the SFT stage, we construct 39,827 training samples containing trajectories generated by 4KAgent [62]. In the RL stage, we build 4,000 image pairs for train-

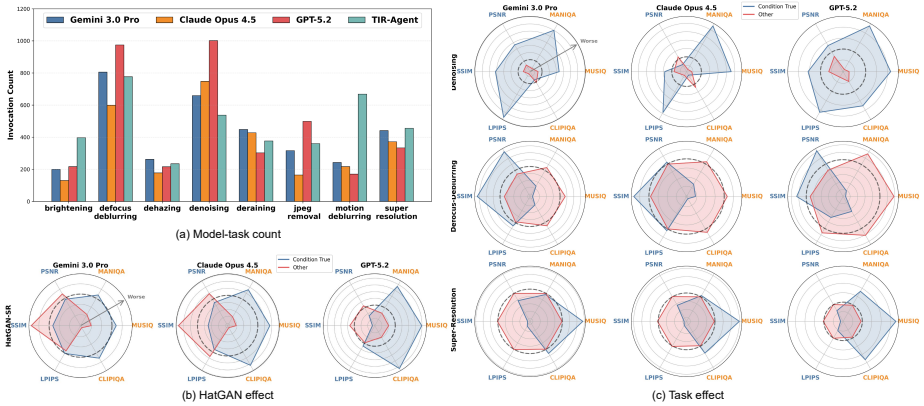


Fig. 4: Analysis of proprietary models. (a) presents the invocation count of each IR tasks on MiO100. (b) presents the effect of HatGAN. (c) presents the effect of IR tasks. Note that for denoising, defocus deblurring, and HatGAN, the condition is proprietary model invokes more times of such task / model, super-resolution is the opposite. Then, we standardize the differences between these models and TIR-Agent in each metric.

ing. The images are sampled from ImageNet [11] and are completely different from the test set to avoid data leakage. To obtain high-quality images, we filter 10k images based on several criteria, including image smoothness, aesthetic score [41], border presence, and overall image quality [55]. Following [61], we generate degraded images from these clean images. Notably, since the depth estimation used for adding haze is not publicly available, we replace it with Depth-Anything [52]. All evaluation experiments are conducted on MiO100 [61]. Following previous works [61, 62], we employ three FR metrics (PSNR, SSIM, LPIPS) and three NR metrics (MANIQA, CLIPIQA, MUSIQ) to evaluate restoration performance.

4.3 Comparison with SOTA Methods

Baselines. We compare our TIR-Agent against both all-in-one and agentic methods. All-in-one methods include AirNet [26], PromptIR [39], MiOIR [23], DA-CLIP [35], InstructIR [10], and AutoDIR [22]. Agentic methods contain three training-free works, *i.e.*, AgenticIR [61], MAIR [21], 4KAgent [62], and another trainable agentic method SimpleCall [34]. Notably, we further compare our method with proprietary SOTA models, including GPT-5.2 [37], Claude Opus 4.5 [1], and Gemini 3.0 Pro [15].

Main comparison. Tab. 1 presents the comparison with 9 state-of-the-art methods under three settings on the MiO100 [61]. We have three observations. 1) ***Effectiveness***: Our TIR-Agent exhibits clear superiority, **achieving consistent improvements on both full-reference (FR) fidelity metrics and no-reference (NR) image quality metrics**. 2) ***Efficiency***: Compared with training-free agentic methods, TIR-Agent trains the VLM to directly select the

Table 4: Quantitative comparison on out-of-domain degradation combinations across two categories: $D = 2$ (mixtures of two degradations) and $D \geq 3$ (mixtures of three, four, five degradations). Details of degradations are provided in the supplementary material. The top three performances of each metric are marked in **bold**, underline, *italic* respectively.

Group	Type	Method	PSNR↑	SSIM↑	LPIPS↓	MANIQA↑	CLIPIQA↑	MUSIQ↑
$D = 2$	All-in-One	AirNet [26]	19.05	0.5892	0.4454	0.2384	0.3030	37.59
		PromptIR [39]	19.53	0.5947	0.4441	0.2445	0.3223	37.06
		MiOIR(R) [23]	18.93	0.6098	0.4131	0.2268	0.3336	39.73
		MiOIR(U) [23]	18.62	0.6092	0.4129	0.2309	0.3422	40.56
		DA-CLIP [35]	*20.20*	0.6266	0.4062	0.2436	0.3371	40.19
		InstructIR [10]	18.63	0.5622	0.4780	0.2622	0.3136	42.38
		AutoDIR [22]	19.18	0.6013	0.4046	0.2980	0.3646	50.31
	Agent	AgenticIR [61]	21.54	0.7006	0.3492	0.2762	*0.3833*	*52.57*
		4KAgent [62]	20.17	*0.6488*	*0.3519*	*0.2936*	**0.4341**	55.65
		TIR-Agent	**23.19**	**0.7526**	**0.3316**	**0.3103**	0.4097	**57.63**
$D \geq 3$	All-in-One	AirNet [26]	17.36	0.4387	0.6162	0.1477	0.2198	25.12
		PromptIR [39]	17.86	0.4502	0.6071	0.1553	0.2356	25.33
		MiOIR(R) [23]	18.11	0.4719	0.6087	0.1540	0.2573	26.76
		MiOIR(U) [23]	18.10	0.4700	0.6067	0.1564	0.2817	26.98
		DA-CLIP [35]	17.84	0.4536	0.6136	0.1729	*0.2843*	28.06
		InstructIR [10]	18.04	0.4664	0.6065	0.1509	0.2218	26.05
		AutoDIR [22]	18.12	0.4651	0.5668	0.1726	0.2429	30.13
	Agent	AgenticIR [61]	*18.13*	0.5237	*0.5429*	*0.1777*	0.2801	38.00
		4KAgent [62]	18.17	*0.5072*	0.5384	0.1909	0.2989	*37.58*
		TIR-Agent	**18.69**	**0.5287**	**0.5233**	**0.2079**	**0.3268**	**45.55**

optimal restoration model, leading to faster inference. In particular, TIR-Agent **achieves at least 2.5× and 5× speedup over AgenticIR and 4KAgent**, respectively. 3) ***IR Expertise***: Compared with leading proprietary models, TIR-Agent outperforms them on most metrics, with **especially notable advantages on NR image quality metrics**.

Next, we compare TIR-Agent with another trainable agentic method, SimpleCall [34], in Tab. 3. Since SimpleCall is not publicly available and the degradation combinations differ, we report results under the partial settings described in the paper. We observe: 1) ***Balance***: SimpleCall^{rr} drops significantly on FR metrics. This phenomenon is consistent with our motivation for MAR, which aims to address the challenge of jointly optimizing heterogeneous image quality metrics.

Table 3: Comparing TIR-Agent with SimpleCall on five degradation combinations in MiO100's GroupA. Only these five combinations have complete results of $SimpleCall^{fr}$ and $SimpleCall^{rr}$.

Method	PSNR↑	SSIM↑	LPIPS↓	MANIQA↑	CLIPIQA↑	MUSIQ↑	Time(s)↓
SimpleCall^{rr}	19.51	0.6474	0.3650	0.3492	0.5252	63.00	80.20
SimpleCall^{fr}	22.51	0.7212	0.2928	0.3346	0.4448	59.65	-
TIR-Agent	**23.01**	**0.7421**	**0.2800**	0.3448	0.4874	61.85	**54.38**

2) ***Superiority***: Compared with SimpleCall^{fr}, TIR-Agent achieves better performance on all metrics. 3) ***Task scalability***: SimpleCall trains an MLP to select restoration tasks and tools without contextual awareness, which prevents it from scheduling super-resolution tasks. In contrast, TIR-Agent can naturally extend to arbitrary restoration tasks. 4) ***Efficiency***: Both methods directly select restoration models, but TIR-Agent achieves higher efficiency, indicating that it can obtain performance with fewer rounds of tool invocations.

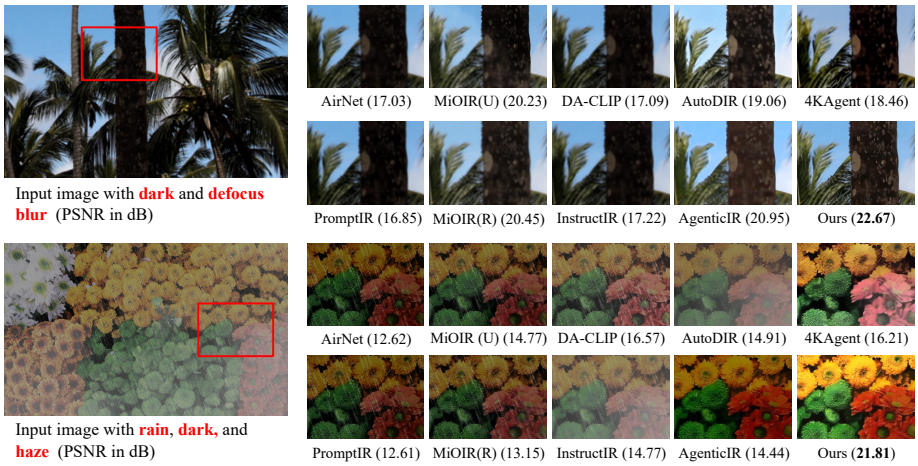


Fig. 5: Visual comparison examples on MiO100 dataset.

Comparison with proprietary models. In Tab. 2, we compare TIR-Agent with three proprietary models. We can observe that these models achieve competitive performance on FR metrics, but perform notably worse on NR metrics. To better understand the underlying factors behind this phenomenon, we conduct a detailed analysis as shown in Fig. 4. 1) ***Task bias.*** As shown in Fig. 4 (a), these models prefer to select denoising and defocus deblurring, while invoking fewer motion deblurring and brightening. It is reasonable, since real-world low-quality images usually lack professional annotations and are often directly attributed to noise. 2) ***Task effect.*** Fig. 4 (c) analyzes the effect of denoising, defocus deblurring, and super-resolution. Excessive denoising degrades MANIQA, LPIPS, and MUSIQ, more defocus deblurring benefits NR metrics but harms PSNR/SSIM. Additionally, the lack of super-resolution invocation improves FR metrics but noticeably reduces NR metrics, especially MUSIQ. Empirically, the low-resolution images may reduce the influence of high-frequency differences. 3) ***Tool effect.*** Without IR-specific training, proprietary models lack knowledge of tool differences for the same task. For example, they prefer HatGAN for super-resolution, which improves FR metrics but degrades NR metrics as demonstrated in Fig. 4 (b). More analysis about task and model is provided in the supplementary material.

Out-of-domain degradation combination validation. To evaluate generalization capability, we conduct experiments on *unseen degradation combinations* during training, with results reported in Tab. 4. We observe: 1) ***Superiority***: Under the settings of $D = 2$ and $D \geq 3$, TIR-Agent achieves the best performance on 11 out of 12 results compared with nine baselines. 2) ***Generalization capability***: As the number of degradations increases, TIR-Agent shows more pronounced advantages on most metrics, particularly NR metrics, demonstrating its strong generalization capability.

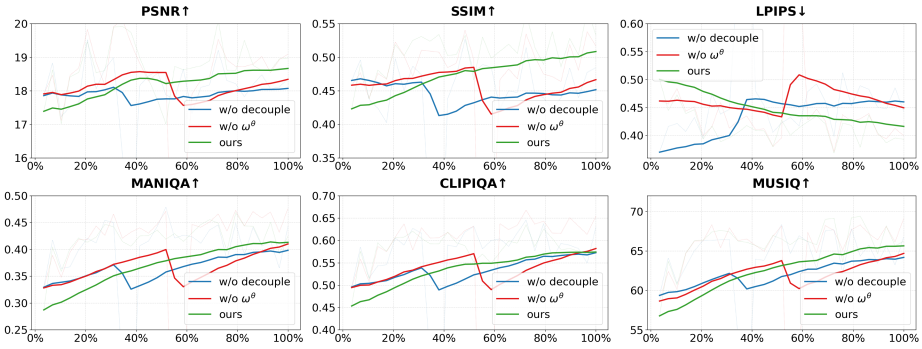


Fig. 6: Training curves under different reward settings: w/o decouple, w/o ω^θ and ours.

Qualitative comparison. According to the qualitative examples in Fig. 5, we observe that our method effectively addresses all degradations in the image and generates visually pleasing results. Besides, comparing all-in-one models and agentic methods, we find that all-in-one models tend to perform well on dehazing, but struggle to handle dark, revealing a notable task bias.

4.4 Ablation Study

Overall ablation. The overall ablation results of the training algorithm and optimizations are provided in Tab. 5. We have the following three observations: 1) ***Importance of SFT and RL***. Both SFT and RL are crucial. SFT enables the VLM with the ability to understand image degradations, select appropriate restoration tasks and models, and follow the required output format. Removing SFT severely degrades performance, especially on PSNR and SSIM. Further, RL encourages the model to jointly improve all metrics on top of the SFT initialization. 2) ***Effect of EDP***. Without EDP, training diversity quickly collapses, aggravating the imbalance among metrics. In later stages of training, this even leads to drops in PSNR and SSIM. 3) ***Effect of MAR***. Without MAR, the model achieves NR metrics close to TIR-Agent but still lags behind on FR metrics.

Table 5: Ablation results of training algorithm and optimization strategy in TIR-Agent.

Method	PSNR↑	SSIM↑	LPIPS↓	MANIQA↑	CLIPIQA↑	MUSIQ↑
TIR-Agent	22.07	0.6935	0.2874	0.3907	0.5454	63.69
Vanilla	20.71	0.6893	0.3501	0.3550	0.4481	56.62
w/o SFT	19.49	0.6097	0.3288	0.3706	0.5239	59.51
w/o RL	21.78	0.6812	0.3060	0.3319	0.5258	60.17
w/o EDP	19.97	0.6227	0.2989	0.3756	0.5518	62.54
w/o α^t	20.61	0.6501	0.2847	0.3745	0.5497	62.49
w/o α^m	21.95	0.6749	0.3256	0.3860	0.5125	60.94
w/o MAR	20.85	0.6792	0.2830	0.3797	0.5437	63.23
w/o decouple	21.86	0.6776	0.3098	0.3660	0.5335	61.17
w/o ω^θ	20.51	0.6573	0.2995	0.3808	0.5462	63.15

Analysis of EDP. We provide a detailed analysis of the effect of EDP on the order schedule and model (tool) selection in Fig. 7, 1) **fine-tuning with the EDP-adjusted SFT data significantly increases the proportion of trajectories that explore order scheduling**. This trend is consistently ob-

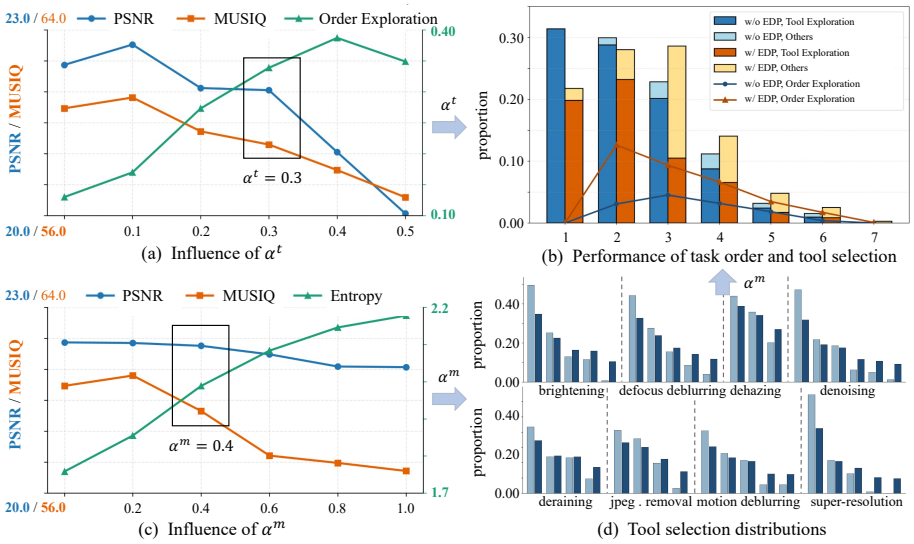


Fig. 7: Influence of EDP and MAR. (a) Influence of α^t on metrics and order exploration (proportion of samples with same tasks with different execution orders). (b) Performance of order and tool exploration. (c) Influence of α^m for metrics and tool selection (average entropy of tool distribution for given task.) (d) Tool selection distributions.

served in samples with 2–6 distinct rollouts among the 8 trajectories generated per sample. Moreover, 2) **the bar plots indicate a substantial increase in model (tool) diversity across different rollouts**. Specifically, Fig. 7 (d) compares the tool selection distributions before and after applying EDP. After EDP adjustment, the predicted distribution becomes smoother, and the usage probability of previously under-invoked tools is significantly increased.

Analysis of MAR. As shown in Fig. 6, using both ω^θ and the decoupling strategy **enables the 6 metrics to converge in a coordinated manner during training, achieving the best performance on five of the metrics.** Without decoupling, the model shows limited improvement on the 3 FR metrics. This is likely because, without normalization, the NR metrics have larger numerical scales and dominate the reward signal, thereby limiting the contribution of the FR metrics. When ω^θ is removed, the model performs relatively well on the three NR metrics, but a noticeable performance drop occurs midway through training. Although the metrics eventually recover, the FR metrics only return to their pre-drop levels, due to the lack of dynamic weight adaptation, which makes it difficult for the model to simultaneously optimize multiple equally weighted objectives. Overall, these results highlight the importance of both decoupling and dynamic weighting in achieving stable and balanced multi-metric optimization.

4.5 Hyperparameter Analysis

We analyze the effect and sensitivity of two hyperparameters in TIR-Agent, α^t and α^m. As shown in Fig. 7(a), increasing α^t continuously improves the order diversity, with a slight drop at 0.5. However, performance begins to degrade significantly when $\alpha^t > 0.3$, especially for PSNR. We attribute this to the fact that after shuffling the order, the model may encounter examples where low-quality images appear before high-quality ones, which makes the model confusing. Therefore, we set $\alpha^t = 0.3$ to balance order diversity and performance.

As shown in Fig. 7 (c), increasing α^m significantly improves tool diversity (larger entropy, *i.e.*, smoother distribution). Meanwhile, α^m has minimal impact on PSNR, indicating that different tools for the same restoration task mainly affect perceptual quality rather than fidelity. We set $\alpha^m = 0.4$ to balance tool diversity and performance.

5 Conclusion

In this paper, we introduce TIR-Agent, a trainable vision-language agent that addresses the fundamental limitations of existing training-free IR agent frameworks. Rather than relying on heuristic planning and exhaustive tool traversal, TIR-Agent learns a direct tool-calling policy that dynamically selects restoration tasks and models based on the current image state, through a two-stage training pipeline of SFT and RL. To enable effective policy learning, we identify two key challenges in agentic RL for IR: limited exploration in restoration trajectories and unstable optimization across heterogeneous image quality metrics. To address these issues, we introduce an EDP strategy that expands the diversity of task ordering and tool selection during SFT, and an MAR mechanism that dynamically balances multiple image quality metrics to prevent reward hacking and stabilize multi-objective optimization. Extensive experiments demonstrate that TIR-Agent consistently outperforms existing all-in-one models and agentic frameworks on both in-domain and out-of-domain composite degradations. Meanwhile, by directly selecting the restoration tool, TIR-Agent achieves over 2.5× faster inference speed. These results highlight the effectiveness of learning-based tool orchestration for IR, and suggest that agentic RL provides a promising direction for building scalable and adaptive IR systems.

Acknowledgements

This work is supported by the National Science and Technology Major Project (2023ZD0121403).

References

1. Anthropic: Claude opus 4.5 (2025), https://www.anthropic.com/news/claude-opus-4-5
2. Bai, S., Cai, Y., Chen, R., et al.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025)
3. Brohan, A., Chebotar, Y., Finn, C., Hausman, K., Herzog, A., Ho, D., Ibarz, J., Irpan, A., Jang, E., Julian, R., et al.: Do as i can, not as i say: Grounding language in robotic affordances. In: CoRL. pp. 287–318. PMLR (2023)
4. Cao, J., Meng, D., Cao, X.: Chain-of-restoration: Multi-task image restoration models are zero-shot step-by-step universal image restorers. arXiv preprint arXiv:2410.08688 (2024)
5. Chan, J.S., Chowdhury, N., Jaffe, O., Aung, J., Sherburn, D., Mays, E., Starace, G., Liu, K., Maksin, L., Patwardhan, T., Weng, L., Mądry, A.: Mle-bench: Evaluating machine learning agents on machine learning engineering. arXiv preprint arXiv:2410.07095 (2024)
6. Chen, H., Li, W., Gu, J., Ren, J., Chen, S., Ye, T., Pei, R., Zhou, K., Song, F., Zhu, L.: Restoreagent: Autonomous image restoration agent via multimodal large language models. Advances in Neural Information Processing Systems **37**, 110643–110666 (2024)
7. Chen, L., Chu, X., Zhang, X., Sun, J.: Simple baselines for image restoration. In: ECCV. pp. 17–33. Springer (2022)
8. Chen, X., Pan, J., Dong, J., Yang, J., Tang, J.: Foundir-v2: Optimizing pre-training data mixtures for image restoration foundation model. arXiv preprint arXiv:2512.09282 (2025)
9. Chen, X., Li, Z., Pu, Y., Liu, Y., Zhou, J., Qiao, Y., Dong, C.: A comparative study of image restoration networks for general backbone network design. In: ECCV. pp. 74–91 (2024)
10. Conde, M.V., Geigle, G., Timofte, R.: Instructir: High-quality image restoration following human instructions. In: European Conference on Computer Vision. pp. 1–21. Springer (2024)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255. IEEE (2009)
12. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence **38**(2), 295–307 (2015)
13. Fan, S., Cui, J., Guo, M.H., Yang, S.: Tool-augmented spatiotemporal reasoning for streamlining video question answering task. arXiv preprint arXiv:2512.10359 (2025)
14. Ferrag, M.A., Tihanyi, N., Debbah, M.: From llm reasoning to autonomous ai agents: A comprehensive review. arXiv preprint arXiv:2504.19678 (2025)
15. Google: Gemini 3 pro: the frontier of vision ai (2025), https://blog.google/innovation-and-ai/technology/developers-tools/gemini-3-pro-vision/
16. Guo, H., Dai, T., Bai, Y., Chen, B., Ren, X., Zhu, Z., Xia, S.: Parameter efficient adaptation for image restoration with heterogeneous mixture-of-experts. In: NeurIPS. vol. 37, pp. 13522–13547 (2024)
17. Guo, Y., Xu, Z., Yao, Z., Lu, Y., Lin, J., Hu, S., Tang, Z., Wang, H., Chen, R.: Octopus: Agentic multimodal reasoning with six-capability orchestration. arXiv preprint arXiv:2511.15351 (2025)

18. Gupta, T., Kembhavi, A.: Visual programming: Compositional visual reasoning without training. In: CVPR. pp. 14953–14962 (2023)
19. Hong, J., Zhao, C., Zhu, C., Lu, W., Xu, G., Yu, X.: Deepeyesv2: Toward agentic multimodal model. arXiv preprint arXiv:2511.05271 (2025)
20. Inc., A.: Photoshop neural filters. https://helpx.adobe.com/photoshop/using/neural-filters.html (2024)
21. Jiang, X., Li, G., Chen, B., Zhang, J.: Multi-agent image restoration. arXiv preprint arXiv:2503.09403 (2025)
22. Jiang, Y., Zhang, Z., Xue, T., Gu, J.: Autodir: Automatic all-in-one image restoration with latent diffusion. In: ECCV. pp. 340–359 (2024)
23. Kong, X., Dong, C., Zhang, L.: Towards effective multiple-in-one image restoration: A sequential and prompt learning strategy. arXiv preprint arXiv:2401.03379 (2024)
24. Labs, T.: Topaz photo ai. https://www.topazlabs.com/topaz-photo-ai (2024)
25. Li, B., Li, X., Lu, Y., Chen, Z.: Hybrid agents for image restoration. arXiv preprint arXiv:2503.10120 (2025)
26. Li, B., Liu, X., Hu, P., Wu, Z., Lv, J., Peng, X.: All-in-one image restoration for unknown corruption. In: CVPR. pp. 17452–17462 (2022)
27. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., Timofte, R.: Swinir: Image restoration using swin transformer. In: ICCV. pp. 1833–1844 (2021)
28. Lim, B., Son, S., Kim, H., Nah, S., Mu Lee, K.: Enhanced deep residual networks for single image super-resolution. In: CVPR-Workshop. pp. 136–144 (2017)
29. Liu, S.Y., Dong, X., Lu, X., Diao, S., Belcak, P., Liu, M., Chen, M.H., Yin, H., Wang, Y.C.F., Cheng, K.T., et al.: Gdpo: Group reward-decoupled normalization policy optimization for multi-reward rl optimization. arXiv preprint arXiv:2601.05242 (2026)
30. Long, X., Ma, Z., Hua, E., Zhang, K., Qi, B., Zhou, B.: Retrieval-augmented visual question answering via built-in autoregressive search engines. In: AAAI. vol. 39, pp. 24723–24731 (2025)
31. Long, X., Tian, K., Xu, P., Jia, G., Li, J., Yang, S., Shao, Y., Zhang, K., Jiang, C., Xu, H., et al.: Adsqa: Towards advertisement video understanding. In: ICCV. pp. 23396–23407 (2025)
32. Long, X., Zeng, J., Meng, F., Ma, Z., Zhang, K., Zhou, B., Zhou, J.: Generative multi-modal knowledge retrieval with large language models. In: AAAI. vol. 38, pp. 18733–18741 (2024)
33. Long, X., Zeng, J., Meng, F., Zhou, J., Zhou, B.: Trust in internal or external knowledge? generative multi-modal entity linking with knowledge retriever. In: Findings of ACL. pp. 7559–7569 (2024)
34. Lu, J., Wu, Y., Zhao, Z., Wang, H., Jimenez, F., Majeedi, A., Fu, Y.: Simplecall: A lightweight image restoration agent in label-free environments with mllm perceptual feedback. arXiv preprint arXiv:2512.18599 (2025)
35. Luo, Z., Gustafsson, F.K., Zhao, Z., Sjölund, J., Schön, T.B.: Controlling vision-language models for multi-task image restoration. arXiv preprint arXiv:2310.01018 (2023)
36. Ma, J., Cheng, T., Wang, G., Zhang, Q., Wang, X., Zhang, L.: Prores: Exploring degradation-aware visual prompt for universal image restoration. arXiv preprint arXiv:2306.13653 (2023)
37. OpenAI: Introducing gpt-5.2 (2025), https://openai.com/index/introducing-gpt-5-2/
38. Park, D., Lee, B.H., Chun, S.Y.: All-in-one image restoration for unknown degradations using adaptive discriminative filters for specific degradations. In: CVPR. pp. 5815–5824. IEEE (2023)

39. Potlapalli, V., Zamir, S.W., Khan, S.H., Shahbaz Khan, F.: Promptir: Prompting for all-in-one image restoration. Advances in neural information processing systems **36**, 71275–71293 (2023)
40. Pu, Y., Zhuo, L., Zhu, K., Xie, L., Zhang, W., Chen, X., Gao, P., Qiao, Y., Dong, C., Liu, Y.: Lumina-omnilv: A unified multimodal framework for general low-level vision. arXiv preprint arXiv:2504.04903 (2025)
41. Schuhmann, C.: Improved aesthetic predictor (2022), `https://github.com/christophschuhmann/improved-aesthetic-predictor`
42. Shen, Y., Song, K., Tan, X., Li, D., Lu, W., Zhuang, Y.: Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. Advances in Neural Information Processing Systems **36**, 38154–38180 (2023)
43. Su, A., Wang, H., Ren, W., Lin, F., Chen, W.: Pixel reasoner: Incentivizing pixel space reasoning via curiosity-driven reinforcement learning. In: NeurIPS (2025)
44. Surís, D., Menon, S., Vondrick, C.: Vipergpt: Visual inference via python execution for reasoning. In: ICCV. pp. 11888–11898 (2023)
45. Tang, J., Chen, J., Wei, W., Xu, X., Liu, R., Wu, X., Xie, Q., Wu, J., Zhang, L., Chen, Q.: Robust-r1: Degradation-aware reasoning for robust visual understanding. arXiv preprint arXiv:2512.17532 (2025)
46. Team, K., Bai, T., Bai, Y., Bao, Y., Cai, S., Cao, Y., Charles, Y., Che, H., Chen, C., Chen, G., et al.: Kimi k2. 5: Visual agentic intelligence. arXiv preprint arXiv:2602.02276 (2026)
47. Wang, C., Feng, K., Chen, D., Wang, Z., Li, Z., Gao, S., Meng, M., Zhou, X., Zhang, M., Shang, Y., et al.: Adatooler-v: Adaptive tool-use for images and videos. arXiv preprint arXiv:2512.16918 (2025)
48. Wang, X., Xie, L., Yu, K., Chan, K.C., Loy, C.C., Dong, C.: Basicsr: Open source image and video restoration toolbox. `https://github.com/XPixelGroup/BasicSR` (2022)
49. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Change Loy, C.: Esrgan: Enhanced super-resolution generative adversarial networks. In: ECCV-Workshop. pp. 0–0 (2018)
50. Wang, Z., Cun, X., Bao, J., Zhou, W., Liu, J., Li, H.: Uformer: A general u-shaped transformer for image restoration. In: CVPR. pp. 17683–17693 (2022)
51. Wu, C., Yin, S., Qi, W., Wang, X., Tang, Z., Duan, N.: Visual chatgpt: Talking, drawing and editing with visual foundation models. arXiv preprint arXiv:2303.04671 (2023)
52. Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H.: Depth anything v2. In: NeurIPS. pp. 21875–21911 (2024)
53. Yao, M., Xu, R., Guan, Y., Huang, J., Xiong, Z.: Neural degradation representation learning for all-in-one image restoration. IEEE transactions on image processing **33**, 5408–5423 (2024)
54. You, Z., Gu, J., Cai, X., Li, Z., Zhu, K., Dong, C., Xue, T.: Enhancing descriptive image quality assessment with a large-scale multi-modal dataset. IEEE Transactions on Image Processing (2025)
55. You, Z., Li, Z., Gu, J., Yin, Z., Xue, T., Dong, C.: Depicting beyond scores: Advancing image quality assessment through multi-modal language models. In: ECCV. pp. 259–276. Springer (2024)
56. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H.: Restormer: Efficient transformer for high-resolution image restoration. In: CVPR. pp. 5728–5739 (2022)

57. Zhang, H., Gu, X., Li, J., Ma, C., Bai, S., Zhang, C., Zhang, B., Zhou, Z., He, D., Tang, Y.: Thinking with videos: Multimodal tool-augmented reinforcement learning for long video reasoning. arXiv preprint arXiv:2508.04416 (2025)
58. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE transactions on image processing **26**(7), 3142–3155 (2017)
59. Zheng, Z., Yang, M., Hong, J., Zhao, C., Xu, G., Yang, L., Shen, C., Yu, X.: Deepeyes: Incentivizing" thinking with images" via reinforcement learning. arXiv preprint arXiv:2505.14362 (2025)
60. Zhou, Y., Cao, J., Zhang, Z., Wen, F., Jiang, Y., Jia, J., Liu, X., Min, X., Zhai, G.: Q-agent: Quality-driven chain-of-thought image restoration agent through robust multimodal large language model. arXiv preprint arXiv:2504.07148 (2025)
61. Zhu, K., Gu, J., You, Z., Qiao, Y., Dong, C.: An intelligent agentic system for complex image restoration problems. arXiv preprint arXiv:2410.17809 (2024)
62. Zuo, Y., Zheng, Q., Wu, M., Jiang, X., Li, R., Wang, J.: 4kagent: Agentic any image to 4k super-resolution. In: NeurIPS (2025)