FINE-GRAINED DEGRADATION REPRESENTATIONS FOR EFFICIENT AGENTIC IMAGE RESTORATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing agentic image restoration methods rely on coarse degradation labels, which cannot capture fine-grained characteristics within the same degradation type and may lead to inefficient tool selection. We propose Locality-prior Agentic Image Restoration (LAIR), an efficient framework that reuses successful restoration trajectories based on fine-grained degradation similarity. LAIR employs a Degradation-Query Momentum Contrastive Encoder (DQ-MCE), which combines learnable degradation queries, cross-content momentum contrastive learning, and parameter-aware regularization to learn content-robust yet degradation-sensitive representations. These representations enable the agent to retrieve, rank, execute, and verify relevant historical tool chains instead of repeatedly rolling back and rescheduling when restoration fails. Experiments on both synthetic and real-world degradations show that LAIR improves restoration fidelity while maintaining competitive perceptual quality. It also reduces runtime by up to 58% and requires fewer tool invocations than AgenticIR. These results demonstrate that fine-grained degradation representations provide an effective bridge between historical restoration trajectories and efficient agentic decision-making.

1 INTRODUCTION

Image Restoration (IR) is a fundamental and important task in computer vision, due to the fact that images are often corrupted by noise, blur, or other distortions during acquisition, transmission, and storage (Li et al., 2022). Image restoration has achieved remarkable progress with task-specific (Chen et al., 2023; Wang et al., 2018; Zamir et al., 2022; Liang et al., 2021) models and all-in-one approaches (Jiang et al., 2024; Li et al., 2022; Potlapalli et al., 2023; Zamfir et al., 2025). However, these methods still struggle with complex real-world degradations, motivating adaptive restoration systems that can dynamically select restoration strategies.

With the development of the reasoning and planning capabilities of LLMs (Achiam et al., 2023; Liu et al., 2024), an increasing number of researchers have begun to explore the use of agent systems for low-level vision tasks (Che et al., 2026), including image restoration (Zhu et al., 2025). Agent-based IR methods usually use pre-trained task-specific IR models as tools and leverage a VLM to perceive the input image and dynamically select the appropriate tools to restore the image (Jiang et al., 2025).

Existing agent-based IR methods can be broadly categorized into training-based and training-free paradigms. Training-based methods require expensive optimization over tool composition (Zhu et al., 2026; Zhang et al., 2026). Training-free methods usually rely on accumulated experiences, but existing memories are coarse-grained and fail to capture degradation-specific characteristics (Cui et al., 2026).

Our method is motivated by an empirical observation: images assigned the same highest haze-severity level by a perception model can favor different fidelity-optimal dehazing tools (Fig. 1). Thus, discrete degradation descriptions are insufficient for reliable tool selection, motivating continuous representations of fine-grained degradation characteristics.

To achieve this goal, we propose a **Degradation-Query Momentum Contrastive Encoder** (DQ-MCE). DQ-MCE introduces a set of learnable degradation queries to aggregate patch-level features associated with different degradation types, enabling each token to capture degradation-specific patterns rather than global semantic information. To further suppress content-dependent variations,

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

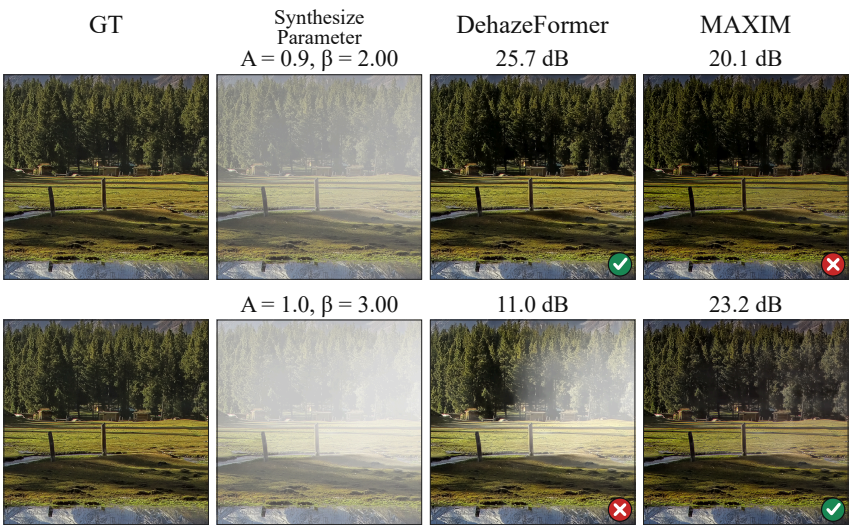


Figure 1: Images with the same MLLM-predicted haze-severity level can favor different fidelity-optimal dehazing tools as the degradation manifestation varies.

we adopt **cross-content momentum contrastive learning**, where images with different contents but identical degradation combinations are encouraged to have similar degradation representations. In addition, a **parameter-aware feature regularizer** encourages DQ-MCE to preserve the fine-grained characteristics. Together, these designs allow DQ-MCE to capture fine-grained degradation characteristics that are critical for restoration tool selection.

With the help of DQ-MCE, we propose an efficient agent-based image restoration framework, **Locality-prior Agentic Image Restoration** (LAIR). In this framework, we reuse the successful tool selection trajectories as effective candidates to restore the degraded image instead of inefficiently rolling back to the initial state when the restoration fails.

2 Related Work

2.1 All-in-One Image Restoration

The target of All-in-One (AiO) image restoration is to restore images with multiple types of degradations using a single model (Jiang et al., 2025). AirNet (Li et al., 2022) is the first AiO IR method, which uses a single model to model multiple types of degradation distributions. PromptIR (Potlapalli et al., 2023) uses a prompt-based approach to adapt a single model to different degradation types, while DA-CLIP (Luo et al., 2024) leverages both content and degradation information to improve restoration performance. DA-CLIP's use of degradation information is closely related to our method. Mixture-of-Experts has also been used to improve the performance of AiO IR methods (Zamfir et al., 2025). Although AiO IR methods can handle multiple types of degradations, they often struggle to generalize to unseen or complex degradations in real-world scenarios (Jiang et al., 2025).

2.2 Agent-based Image Restoration

Agent-based methods address the problem of generalization by employing a powerful vision-language model to dynamically select the appropriate restoration tools for each input image. RestoreAgent (Chen et al., 2024a) and AgenticIR (Zhu et al., 2025) first leverage in-context learning capabilities of large language models to select restoration tools while a finetuned VLM is used to perceive the input image and provide feedback on the restoration results. Multi-agent systems are also used to improve the performance of agent-based image restoration, such as HybridAgent (Li et al., 2026) and MAIR (Jiang et al., 2026). They utilize specialized sub-agents or expert agents to address complex restoration tasks, achieving better performance through collaboration and coor-

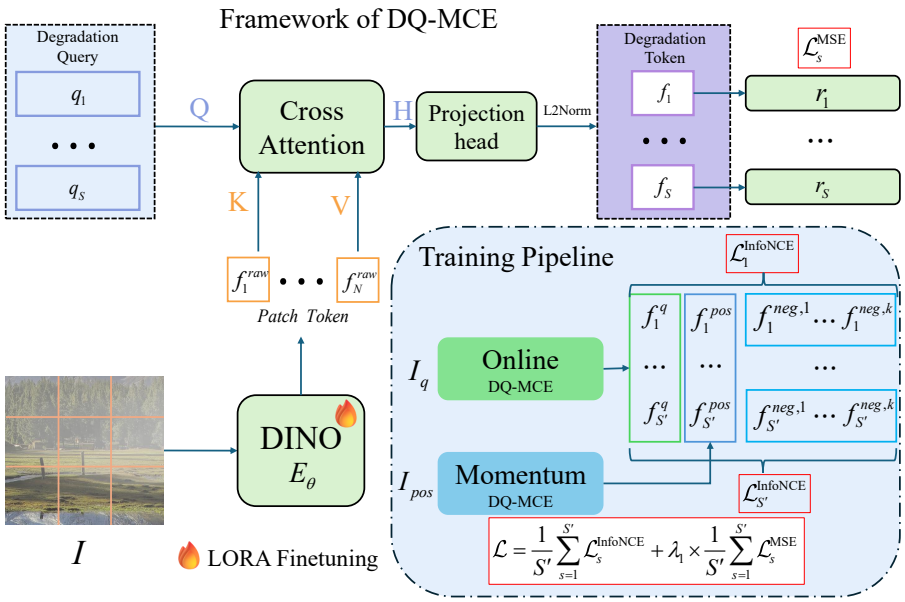


Figure 2: The architecture and training pipeline of the DQ-MCE. The loss labels indicate where the corresponding losses are computed.

dination among agents. Training-based methods (Lin et al., 2025; Zhu et al., 2026; Zhang et al., 2026) supervise the agent to learn a better tool selection strategy, requiring large amounts of training data and substantial computational resources. Our method is similar to SEAR (Cui et al., 2026), which also reuses successful tool selection trajectories to improve the agent's performance over time. However, SEAR only divides the degradation space into a few discrete levels, which is too coarse to reflect the real characteristics of degraded images.

3 METHOD

3.1 DEGRADATION-QUERY MOMENTUM CONTRASTIVE ENCODER

Tool performance varies with fine-grained degradation attributes, such as atmospheric light A and scattering coefficient β in hazy images (Fig. 1). General-purpose encoders primarily capture semantic and structural information, potentially entangling degradation with image content. Reliable tool selection therefore requires representations that distinguish fine-grained degradation characteristics.

To address this limitation, we propose a **Degradation-Query Momentum Contrastive Encoder** (DQ-MCE), which is based on cross-content momentum contrastive learning, motivated by MoCo (He et al., 2020). As illustrated in Fig. 2, the training pipeline consists of an online branch and a momentum branch. The online encoder is updated by minimizing the training objective through gradient descent, while the momentum encoder is not optimized by back-propagation. Instead, its parameters are updated as an exponential moving average of the online encoder:

$$\theta_m \leftarrow m\theta_m + (1-m)\theta_o, \tag{1}$$

where θ_m and θ_o denote the parameters of the momentum and online encoders, respectively, and m is the momentum coefficient.

For either the online or momentum branch, DINO (Siméoni et al., 2025) E_θ maps a degraded image I into N patch-level feature tokens of dimension d. We introduce S learnable degradation queries $Q = [q_1, \ldots, q_S] \in \mathbb{R}^{S\times d}$, each associated with one predefined degradation type. Using Q as queries and F^{raw} as keys and values, multi-head cross-attention aggregates the patch tokens into degradation-specific hidden states. A shared projection head maps these states into a compact

embedding space, followed by row-wise ℓ_2 normalization:

$$\begin{aligned} F^{raw} &= E_\theta(I) = [f_1^{raw}, \ldots, f_N^{raw}] \in \mathbb{R}^{N\times d}, \\ H &= \mathrm{CrossAttn}(Q, F^{raw}, F^{raw}) \in \mathbb{R}^{S\times d}, \\ F &= \mathrm{L2Norm}(g_\phi(H)) = [f_1, \ldots, f_S]^\top \in \mathbb{R}^{S\times D}, \end{aligned} \tag{2}$$

where f_s denotes the degradation representation associated with the s-th degradation type.

3.2 Training pipeline

As shown in Fig. 2, we perform token-wise contrastive learning, where each degradation token is optimized independently to learn degradation-specific representations rather than a global semantic representation.

Given an anchor image I_q with degradation combination D, a token f_s is active when its corresponding degradation type is present in D. We select a positive image I_{pos} from images generated using the same degradation combination D but a different clean source image. Among these candidates, we choose the one with the smallest Euclidean distance to the anchor in the space of concatenated normalized degradation parameters which will be introduced later. For each active degradation token f_s, the online embedding f_s^q and the corresponding momentum embedding f_s^{pos} form a positive pair.

Negative samples are momentum embeddings of the same token slot s from images in which degradation type s is absent, drawn from both the current minibatch and a degradation-specific FIFO queue. These images may share other degradation types with the anchor image I_q, but the absence of degradation s ensures that in token s, the anchor and negative embeddings are not confused by the same degradation pattern. This objective encourages degradation-specific representations while reducing their sensitivity to image content.

Although contrastive learning encourages alignment between images with the same degradation combination, InfoNCE (van den Oord et al., 2019) does not explicitly preserve variations within a degradation type. Fine-grained characteristics may therefore be weakened or encoded arbitrarily.

To address this issue, we introduce a degradation-parameter prediction objective. For each active degradation token f_s, a parameter regressor r_s predicts its normalized degradation parameters from the projected token embedding,

$$\hat{p_s} = \sigma(r_s(f_s)), \tag{3}$$

where σ denotes the sigmoid function and the ground-truth parameter vector p_s is obtained from the synthesis process of degraded images which is also used in positive pairing for contrastive learning. Detailed implementation information can be found in the Appendix B.1.

Since the regressor receives only f_s, accurate prediction requires the token to retain parameter-dependent cues. The objective thus acts as **a parameter-aware feature regularizer** for within-type degradation variations. The loss terms and overall training objective are defined as:

$$\mathcal{L}_s^{\text{MSE}} = ||p_s - \hat{p_s}||^2, \tag{4}$$

$$\mathcal{L}_s^{\text{InfoNCE}} = -\log \frac{\exp(\mathrm{sim}(f_s^q, f_s^{pos})/\tau)}{\sum_{z\in\{f_s^{pos}, f_s^{neg,1}, \ldots\}} \exp(\mathrm{sim}(f_s^q, z)/\tau)}, \tag{5}$$

$$\mathcal{L} = \frac{1}{S'}\sum_{s=1}^{S'} \mathcal{L}_s^{\text{InfoNCE}} + \frac{\lambda}{S'}\sum_{s=1}^{S'} \mathcal{L}_s^{\text{MSE}}, \tag{6}$$

where τ is the temperature parameter in InfoNCE loss (van den Oord et al., 2019), S' is the number of active degradation tokens, and λ is the weight for the MSE loss.

Figure 3 shows smooth, ordered haze-token variations with physical degradation parameters across diverse clean source images. This coherence suggests that DQ-MCE preserves fine-grained degradation characteristics while reducing content-dependent variations.

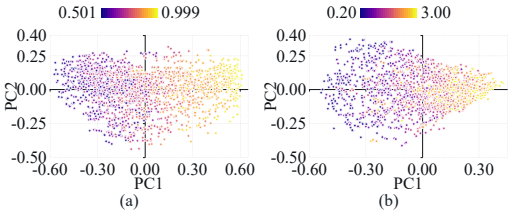


Figure 3: PCA visualization of DQ-MCE haze-token representations. Hazy inputs are synthesized from different GT images, with points colored by (a) atmospheric light A and (b) scattering coefficient β.

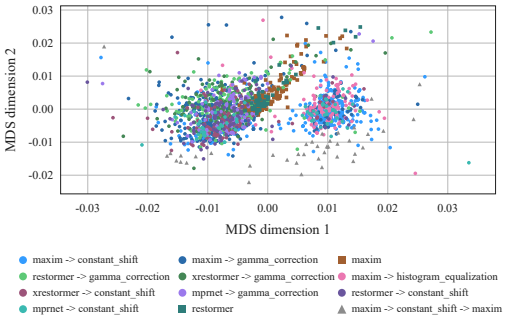


Figure 4: MDS visualization of the feature space. Each point represents a degraded input, and its color and marker denote the restoration tool chain that achieves the best performance for that input.

3.3 Tool-chain Locality Prior

Although DQ-MCE uses no tool-selection supervision, nearby representations tend to share the same optimal tool chain. The MDS visualization in Fig. 4 approximately preserves pairwise feature distances and illustrates this pattern through matching colors and markers. This motivates locality-based memory retrieval, using successful historical trajectories as candidates for similarly degraded images.

3.4 Locality-prior Agentic Image Restoration

Instead of rolling back when a restoration attempt fails, we propose a **Locality-prior Agentic Image Restoration framework (LAIR)**, which retrieves and reuses historical successful restoration trajectories as candidates for restoring new degraded images.

3.4.1 Locality-Prior-Guided Memory Retrieval

We construct a memory bank M to store previously validated restoration trajectories. Each memory entry associated with historical image j is represented as $m_j = (\hat{D}_j, F_j, P_j, T_j, G_j)$, where $\hat{D}_j$ denotes the degradation set of the corresponding input image predicted by the perception model, F_j is its structured degradation representation extracted by DQ-MCE, P_j and T_j denote the executed restoration plan and tool chain. G_j denotes the quality grade of the stored memory entry.

When processing a new degraded image I, a perception model first predicts the degradation set $\hat{D}_I$ of the image and DQ-MCE extracts its structured degradation representation $F_I = \{f_{I,1}, f_{I,2}, \ldots, f_{I,S}\}$, where each $f_{I,s}$ corresponds to one predefined degradation token. We compute the memory-retrieval score using the cosine similarity between the corresponding degradation tokens:

$$S_{\text{ret}}(I, m_j) = \sum_{d \in \hat{D}_I} \text{Sim}_{\text{token}}(f_{I,d}, f_{j,d}). \tag{7}$$

Here, $\text{Sim}_{\text{token}}$ denotes the cosine similarities of the degradation tokens relevant to the current input image.

The top-K memories are obtained as

$$\mathcal{R}_K(I) = \text{TopK}_{m_j \in M}\, S_{\text{ret}}(I, m_j). \tag{8}$$

A high retrieval score indicates that a historical memory is relevant under the proposed degradation prior. However, feature-level relevance does not necessarily imply that the stored tool chain covers every sub-degradation required by the current image. We select the set of memories $\mathcal{R}_{\text{exact}}(I)$ from the top-K retrieved memories:

$$\mathcal{R}_{\text{exact}}(I) = \left\{ m_j \in \mathcal{R}_K(I) \mid \hat{D}_j = \hat{D}_I \right\}, \tag{9}$$

which contains only the memories that match the degradation set of the current image.

3.4.2 VERIFIED MEMORY REUSE

If $\mathcal{R}_{\text{exact}}(I)$ is non-empty, its memory entries are first reordered according to their stored quality grades. Starting from the highest-grade entry, we execute its complete historical tool chain T_j to obtain $I_j^{\text{out}} = \mathcal{E}(I; T_j)$, where $\mathcal{E}$ denotes the restoration execution environment. A retrieved memory is therefore treated as a promising candidate rather than an unconditionally correct solution.

When the degraded image is processed, evaluation of the restoration outcome is performed. For each degradation d in the input image, the perception model gives a discrete label γ_d to evaluate the severity in the restored image. We define the global severity of a restored image,

$$\Gamma = \min_{d \in \hat{D}_I} \gamma_d, \tag{10}$$

where $\gamma_d \in \{1, 2, 3, 4, 5\}$, $\gamma = 1$ means the highest severity, and $\gamma = 5$ indicates that almost no degradation remains. The system will try all retrieved memory entries in a grade-based manner until one of them yields a satisfactory restoration outcome ($\Gamma \geq 4$) which is decided by the perception model. When all the memory candidates fail, the system will switch to the planning stage, following the setting of AgenticIR (Zhu et al., 2025). The detailed pipeline is provided in the Appendix B.2.

3.4.3 RESTORATION-FIRST MEMORY UPDATE

If a restoration result is satisfactory ($\Gamma \geq 4$), its trajectory is stored in the memory bank. We assign each validated trajectory a hierarchical grade:

$$G_{\text{perc}} = \frac{1}{2}\left(\text{CLIP-IQA} + \frac{\text{MUSIQ}}{100}\right), \qquad G_I = \Gamma + G_{\text{perc}}. \tag{11}$$

Here, Γ specifies the primary restoration level, while $G_{\text{perc}} \in [0, 1)$ provides a secondary ranking based on no-reference perceptual quality. Since adjacent restoration levels are separated, the resulting grade satisfies $G_I \in [\Gamma, \Gamma + 1)$. Consequently, memories are prioritized first by the completeness of degradation removal and then by perceptual quality within the same restoration level. This restoration-prioritized grading encourages the agent to reuse trajectories that effectively remove degradations while preserving image details and thus results in a high-fidelity restoration outcome.

4 EXPERIMENTS

4.1 IMAGE RESTORATION IMPLEMENTATION DETAILS

Following most agent-based image restoration work (Zhu et al., 2025; Cui et al., 2026), the pre-trained restoration models X-Restormer (Chen et al., 2024b), HAT (Chen et al., 2025), FBCNN (Jiang et al., 2021), SwinIR (Liang et al., 2021), DiffBIR (Lin et al., 2024), MPRNet (Zamir et al., 2021), DRBNet (Ruan et al., 2022), DehazeFormer (Song et al., 2023), MAXIM (Tu et al., 2022), RIDCP (Wu et al., 2023), Restormer (Zamir et al., 2022) are used as restoration tools. All experiments are conducted on a single NVIDIA RTX A6000 GPU.

We construct two different memory banks. The first one M_1 is constructed from the training set used by AgenticIR (Zhu et al., 2025), consisting of 160 images. The second memory bank, M_2, is used for ablation study, constructed by applying **rain+dark** degradation combinations from the LSDIR (Li et al., 2023) dataset, consisting of 2200 images. If not mentioned, the memory bank used in our experiments is M_1.

The "best tool" labels are obtained by ranking all candidate tool chains according to the aggregated z-score of PSNR, SSIM, LPIPS, CLIP-IQA, and MUSIQ, where all metrics are equally weighted. And the tool-chain in the above memory bank is initially constructed by the best tool for each degraded image. Following AgenticIR (Zhu et al., 2025), we use a finetuned DepictQA (You et al., 2024) as the perception model. In all experiments, we use top-$K = 5$ for memory retrieval. And the detailed implementation of DQ-MCE is provided in the Appendix B.3.

Datasets, baselines, and metrics. Following AgenticIR (Zhu et al., 2025), we evaluate on three groups of synthesized images: Group A contains known combinations of two degradation types,

Table 1: Quantitative comparison on different image groups. PSNR and SSIM are computed on the Y channel. All agent-based methods use the same unified restoration-tool library. The best and second-best results are shown in **boldface** and underlining, respectively.

Datasets	Method	PSNR↑	SSIM↑	LPIPS↓	MANIQA↑	CLIP-IQA↑	MUSIQ↑
Group A ($n = 640$)	PromptIR (NIPS′23)	21.4075	0.6701	0.4254	0.2693	0.3957	42.8456
	MiOIR	21.3563	0.6834	0.4074	0.2626	0.4120	43.8445
	InstructIR (ECCV′24)	20.2364	0.6409	0.4626	0.2630	0.3781	42.9907
	AgenticIR (ICLR′25)	22.0847	0.7097	0.3027	0.3392	0.4874	60.9504
	4KAgent (NIPS′25)	21.6468	0.6936	0.3082	0.3403	0.5017	61.1519
	SEAR	21.8042	0.6961	0.3019	**0.3411**	**0.5039**	**61.2882**
	Ours	**22.4440**	**0.7157**	**0.3016**	0.3335	0.4826	60.3328
Group B ($n = 400$)	PromptIR (NIPS′23)	20.3621	0.7146	0.3550	0.2961	0.4208	47.5886
	MiOIR	20.6956	0.7244	0.3477	0.2830	0.4325	48.5851
	InstructIR (ECCV′24)	20.8317	0.6761	0.4133	0.3016	0.4006	48.9953
	AgenticIR (ICLR′25)	22.2607	**0.7348**	0.2949	0.3321	0.4951	60.3346
	4KAgent (NIPS′25)	21.5416	0.7175	0.2958	0.3434	**0.5226**	**61.4795**
	SEAR (ECCV′26)	22.1332	0.7251	**0.2890**	**0.3447**	0.5134	60.6520
	Ours	**22.4498**	0.7325	0.2939	0.3427	0.4981	61.0305
Group C ($n = 400$)	PromptIR (NIPS′23)	20.6046	0.6012	0.5584	0.2059	0.3189	29.8850
	MiOIR	20.5308	**0.6015**	0.5725	0.1962	0.3171	30.1058
	InstructIR (ECCV′24)	19.4498	0.5781	0.5756	0.1890	0.3103	29.2901
	AgenticIR (ICLR′25)	20.6268	0.5964	0.4286	**0.3000**	0.4458	**53.4485**
	4KAgent (NIPS′25)	20.3078	0.5798	0.4223	0.2929	**0.4737**	52.6504
	SEAR (ECCV′26)	20.6121	0.5927	0.4173	0.2742	0.4420	52.7338
	Ours	**20.9178**	0.6010	**0.4147**	0.2991	0.4612	53.0653

while Groups B and C contain unseen combinations of two and three types, respectively. We compare against the All-in-One methods PromptIR (Potlapalli et al., 2023), MiOIR (Kong et al., 2024), and InstructIR (Conde et al., 2024), and the training-free agent methods AgenticIR (Zhu et al., 2025), 4KAgent (Zuo et al., 2025), and SEAR (Cui et al., 2026). SEAR is evaluated using the metrics in its original paper; 4KAgent uses the same tool set as the other agent-based methods except in the tool-library extension experiments. Evaluation metrics are PSNR and SSIM (Wang et al., 2004) (both on the Y channel), LPIPS (Zhang et al., 2018), MANIQA (Yang et al., 2022), CLIP-IQA (Wang et al., 2023), and MUSIQ (Ke et al., 2021).

4.2 Comparison with State-of-the-Art Methods

As shown in Table 1, our method achieves the highest PSNR on all three groups, outperforming AgenticIR by 0.3593, 0.1891, and 0.2910 dB on Groups A, B, and C, respectively. It also obtains the best LPIPS on Groups A and C while maintaining competitive SSIM. Notably, on the more challenging Group C, our method ranks first in PSNR and LPIPS and second in all three no-reference quality metrics, with SSIM only 0.0005 below the best result. These results validate that historical tool chains retrieved in the DQ-MCE space can serve as effective restoration candidates for high-fidelity restoration, even under complex three-degradation inputs. And there is qualitative analysis of the restoration results in the Appendix A.1.

4.2.1 Time Efficiency Comparison

Table 2 compares LAIR with AgenticIR (Zhu et al., 2025). Replacing rollback and rescheduling with memory reuse maintains comparable restoration quality while reducing average tool invocations from 7.31 to 4.64 and runtime by 58.00%.

We also integrate memory reuse into 4KAgent (Zuo et al., 2025), retaining its original setting and larger tool library. Table 2 shows unchanged PSNR with lower SSIM, approximately half as many tool invocations, and a 24.58% runtime reduction. These results demonstrate applicability to a larger tool library; detailed settings are provided in the Appendix B.4.

Table 2: Time efficiency (per-image averages). T.I.: tool invocations. Compare within each block: unified (upper) or extended 4KAgent (lower) tools.

Method	Time (s) ↓	PSNR ↑	SSIM ↑	T.I. ↓
Unified tool library				
Our Method	**65.90**	**22.45**	0.7325	**4.64**
AgenticIR	156.91	22.26	**0.7348**	7.31
Extended 4KAgent tool library				
M.R. 4KAgent	**193.19**	21.01	0.6616	**5.70**
4KAgent	256.15	21.01	**0.6767**	11.22

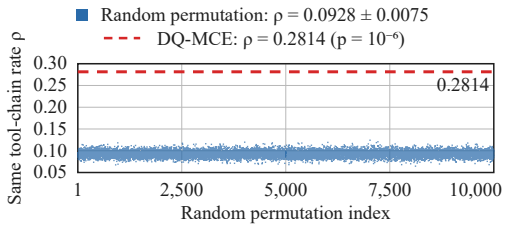


Figure 5: The same tool-chain rate of the first 10,000 permutations of the best tool-chain labels in the feature space.

Table 3: Comparison of three agent-based methods on 80 real-world degraded images from four datasets.

Metric	AgenticIR	4KAgent	Ours
PSNR↑	19.6698	**20.0533**	19.6934
SSIM↑	0.5823	0.6192	**0.6609**
LPIPS↓	0.5358	0.5013	**0.4929**
MANIQA↑	0.3417	0.3516	**0.3576**
CLIP-IQA↑	0.4447	0.4438	**0.4448**
MUSIQ↑	52.0525	**53.9053**	53.2327
Avg. Tool Invoc.↓	15.05	12.15	**9.84**
Time (s)↓	252.95	286.20	**182.13**

Table 4: Retrieval-only feature ablation on 100 **rain+dark** images from Group B.

Extractor	PSNR↑	SSIM↑	Recall↑	Rank↓
DQ-MCE	**24.60**	**0.8885**	**29**	**6.42**
DQ-MCE w/o $\mathcal{L}^{\text{MSE}}$	23.93	0.8703	<u>25</u>	8.54
severity-L2	23.98	0.8783	24	<u>8.16</u>
mean patch	23.68	0.8757	20	8.42
CLS token	23.45	0.8677	22	9.37

4.3 Real-World Generalization

To evaluate generalization beyond synthetic degradations, we select 80 images from four real-world restoration datasets: DRealSR (Wei et al., 2020), NH-HAZE (Ancuti et al., 2020), LHP-RAIN (Guo et al., 2023), and SIDD (Abdelhamed et al., 2018), with 20 images randomly sampled from each dataset.

The memory bank is initialized with synthetic restoration experiences, as in the previous experiments. In Table 3, LAIR requires 9.84 tool invocations on average, 34.6% fewer than AgenticIR and 19.0% fewer than 4KAgent, with a runtime of 182.13 seconds. It leads on four quality metrics, while 4KAgent achieves the highest PSNR and MUSIQ. These results support transferring synthetic restoration experiences to real-world images with competitive quality and lower execution cost.

4.4 Ablation Study

4.4.1 Analysis of Tool-Selection Locality

For each sample in M_2, we use its optimal tool chain as a label in the feature space in our DQ-MCE, then we retrieve its nearest neighbor according to the degradation representation and measure whether the retrieved image shares the same optimal restoration tool chain, obtaining a same tool-chain rate of $\rho = 0.2814$. Then we randomly permute these best tool-chain labels $1,000,000$ times and keep the position in the feature space, getting the average same tool-chain rate only $\rho = 0.0928 \pm 0.0075$, 32.98% of the observed value. None of the permutations reaches the observed value ($p = 10^{-6}$), demonstrating significant tool-chain locality despite the absence of tool-selection supervision, where the first 10,000 permutations are shown in Fig. 5.

4.4.2 Ablation Study on DQ-MCE

To isolate the effect of retrieval representations, we retrieve the top-5 experiences from a fixed memory bank, that is, disabling memory updates, just retrieving from M_2. For Severity-L2, the similarity

Table 5: Ablation study of the proposed memory-reuse strategy on Groups A, B, and C. The best and second-best results among the three configurations are highlighted in bold and underlined, respectively.

Configuration	PSNR↑	SSIM↑	LPIPS↓	MANIQA↑	CLIP-IQA↑	MUSIQ↑
Group A ($n = 640$)						
Base	22.1086	0.7127	<u>0.3034</u>	<u>0.3327</u>	**0.4879**	<u>60.2879</u>
w/o Grading	<u>22.3253</u>	<u>0.7146</u>	0.3036	0.3320	0.4806	60.2005
Ours	**22.4440**	**0.7157**	**0.3016**	**0.3335**	<u>0.4826</u>	**60.3328**
Group B ($n = 400$)						
Base	22.3098	0.7302	**0.2911**	<u>0.3311</u>	<u>0.5001</u>	<u>60.5037</u>
w/o Grading	<u>22.3216</u>	<u>0.7311</u>	<u>0.2939</u>	0.3308	**0.5009**	60.2300
Ours	**22.4498**	**0.7325**	<u>0.2939</u>	**0.3427**	0.4981	**61.0305**
Group C ($n = 400$)						
Base	<u>20.7111</u>	0.5989	<u>0.4201</u>	<u>0.2973</u>	**0.4627**	52.7779
w/o Grading	20.6361	**0.6021**	0.4220	0.2967	0.4601	<u>52.8110</u>
Ours	**20.9178**	<u>0.6010</u>	**0.4147**	**0.2991**	<u>0.4612</u>	**53.0653**

is computed using the L2 distance between the discrete degradation severity levels predicted by DepictQA, similar to the approach in SEAR (Cui et al., 2026). For the DINOv3 (Siméoni et al., 2025) baselines, we use either the mean-pooled patch tokens or the CLS token as the global image representation and compute cosine similarity for retrieval.

In Table 4, "Recall" counts images whose top-5 retrieved experiences contain a best tool chain; "Rank" is its average rank in the **rain+dark** tool-chain search space. DQ-MCE achieves the best fidelity and retrieval results. Removing $\mathcal{L}^{\mathrm{MSE}}$ degrades all four metrics, supporting parameter-level supervision. DQ-MCE also outperforms Severity-L2 and both DINOv3 features, supporting fine-grained tokens for restoration-experience retrieval.

4.4.3 Image Restoration Pipeline Ablation

As shown in Table 5, "Base" plans and executes restoration using static scheduling experiences **without rollback or rescheduling**. "w/o Grading" adds memory reuse without sorting retrieved candidates by quality; "Ours" further enables quality-based ordering. Table 5 shows that the full pipeline improves PSNR over Base by 0.3354, 0.1400, and 0.2067 dB on Groups A, B, and C, respectively. It achieves the best results on five metrics in Group A and four in both Groups B and C. Base retains higher CLIP-IQA in every group and lower LPIPS on Group B. Overall, these results suggest that memory reuse with quality-based ordering provides effective candidate trajectories for restoration.

Compared with "w/o Grading", quality-based ordering improves PSNR across all groups and all six metrics on Group A. It also improves most metrics on Groups B and C, except for tied LPIPS and lower CLIP-IQA on Group B, and lower SSIM on Group C. Grading thus yields consistent gains in PSNR, MANIQA, and MUSIQ across all three groups, despite metric-dependent trade-offs.

5 Conclusion

We introduced LAIR, an efficient agentic image restoration framework with memory reuse to leverage historical restoration trajectories. To make prior restoration trajectories reusable at the appropriate granularity, we developed DQ-MCE, which combines degradation queries and cross-content momentum contrastive learning to learn content-robust yet manifestation-sensitive degradation representations. The resulting feature space exhibits significant tool-chain locality without tool-selection supervision. Both on synthetic and real-world degraded images, LAIR achieves competitive restoration quality with fewer tool invocations and reduced runtime compared to state-of-the-art methods. These results suggest that fine-grained degradation locality can turn successful restorations into an effective prior for more efficient agentic image restoration.

AI USE STATEMENT

We used AI tools to assist with debugging experimental code and monitoring experiment runs. Generative AI tools were also used for translation, language editing, LaTeX formatting.

The authors reviewed the AI-assisted text, code changes, and result interpretations and take responsibility for the final content and reported results.

REFERENCES

Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. A high-quality denoising dataset for smartphone cameras. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1692–1700, 2018. doi: 10.1109/CVPR.2018.00182.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Codruta O. Ancuti, Cosmin Ancuti, and Radu Timofte. Nh-haze: An image dehazing benchmark with non-homogeneous hazy and haze-free images. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1798–1805, 2020. doi: 10.1109/CVPRW50498.2020.00230.

Lirong Che, Zhenfeng Gan, Yanbo Chen, Junbo Tan, and Xueqian Wang. Photoagent: A robotic photographer with spatial and aesthetic understanding. *arXiv preprint arXiv:2603.22796*, 2026.

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*, pp. 1692–1703, 2023.

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

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

Xiangyu Chen, Xintao Wang, Wenlong Zhang, Xiangtao Kong, Yu Qiao, Jiantao Zhou, and Chao Dong. Hat: Hybrid attention transformer for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Marcos V. Conde, Gregor Geigle, and Radu Timofte. Instructir: High-quality image restoration following human instructions, 2024. URL https://arxiv.org/abs/2401.16468.

Shuang Cui, Fan Ji, Guanglong Sun, Yufei Guo, Xiongxin Tang, Jiangmeng Li, and Fanjiang Xu. Self-evolving agentic image restoration via deliberate planning and intuitive execution. *arXiv preprint arXiv:2606.28971*, 2026.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026. doi: 10.48550/arXiv.2606.19348. URL https://arxiv.org/abs/2606.19348.

Yun Guo, Xueyao Xiao, Yi Chang, Shumin Deng, and Luxin Yan. From sky to the ground: A large-scale benchmark and simple baseline towards real rain removal, 2023. URL https://arxiv.org/abs/2308.03867.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

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

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

Xu Jiang, Gehui Li, Bin Chen, and Jian Zhang. Multi-agent image restoration. *International Journal of Computer Vision*, 134(5):205, 2026.

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

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

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.

Bingchen Li, Xin Li, Yiting Lu, and Zhibo Chen. Hybrid agents for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22636–22647, 2026.

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*, pp. 17452–17462, 2022.

Yawei Li, Kai Zhang, Jingyun Liang, Jiezhang Cao, Ce Liu, Rui Gong, Yulun Zhang, Hao Tang, Yun Liu, Denis Demandolx, et al. Lsdir: A large scale dataset for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1775–1787, 2023.

Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1833–1844, 2021.

Xinqi Lin, Jingwen He, Ziyan Chen, Zhaoyang Lyu, Bo Dai, Fanghua Yu, Yu Qiao, Wanli Ouyang, and Chao Dong. Diffbir: Toward blind image restoration with generative diffusion prior. In *European conference on computer vision*, pp. 430–448. Springer, 2024.

Yunlong Lin, Zixu Lin, Haoyu Chen, Panwang Pan, Chenxin Li, Sixiang Chen, Kairun Wen, Yeying Jin, Wenbo Li, and Xinghao Ding. Jarvisir: Elevating autonomous driving perception with intelligent image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22369–22380, 2025.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Ziwei Luo, Fredrik K Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas Schön. Controlling vision-language models for multi-task image restoration. In *International Conference on Learning Representations*, volume 2024, pp. 16226–16246, 2024.

Vaishnav Potlapalli, Syed Waqas Zamir, Salman H Khan, and Fahad Shahbaz Khan. Promptir: Prompting for all-in-one image restoration. *Advances in neural information processing systems*, 36:71275–71293, 2023.

Lingyan Ruan, Bin Chen, Jizhou Li, and Miuling Lam. Learning to deblur using light field generated and real defocus images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16304–16313, 2022.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Yuda Song, Zhuqing He, Hui Qian, and Xin Du. Vision transformers for single image dehazing. *IEEE Transactions on Image Processing*, 32:1927–1941, 2023.

Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxim: Multi-axis mlp for image processing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5769–5780, 2022.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019. URL https://arxiv.org/abs/1807.03748.

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

Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pp. 0–0, 2018.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

Pengxu Wei, Ziwei Xie, Hannan Lu, Zongyuan Zhan, Qixiang Ye, Wangmeng Zuo, and Liang Lin. Component divide-and-conquer for real-world image super-resolution, 2020. URL https://arxiv.org/abs/2008.01928.

Rui-Qi Wu, Zheng-Peng Duan, Chun-Le 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*, pp. 22282–22291, 2023.

Sidi Yang, Tianhe Wu, Shuwei Shi, Shanshan Lao, Yuan Gong, Mingdeng Cao, Jiahao Wang, and Yujiu Yang. Maniqa: Multi-dimension attention network for no-reference image quality assessment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1191–1200, 2022.

Zhiyuan You, Zheyuan Li, Jinjin Gu, Zhenfei Yin, Tianfan Xue, and Chao Dong. Depicting beyond scores: Advancing image quality assessment through multi-modal language models. In *European Conference on Computer Vision*, pp. 259–276, 2024.

Eduard Zamfir, Zongwei Wu, Nancy Mehta, Yuedong Tan, Danda Pani Paudel, Yulun Zhang, and Radu Timofte. Complexity experts are task-discriminative learners for any image restoration. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12753–12763, 2025.

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 14816–14826. IEEE, 2021.

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*, pp. 5728–5739, 2022.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. URL https://arxiv.org/abs/1801.03924.

Yisheng Zhang, Guoli Jia, Haote Hu, Shanxu Zhao, Kaikai Zhao, Long Sun, Xinwei Long, Kai Tian, Che Jiang, Zhaoxiang Liu, et al. Tir-agent: Training an explorative and efficient agent for image restoration. *arXiv preprint arXiv:2603.27742*, 2026.

Feng Zhu, Shuyang Xie, Yihan Zeng, Ming Liu, and Wangmeng Zuo. Opera: An agent for image restoration with end-to-end joint planning-execution optimization. *arXiv preprint arXiv:2605.22104*, 2026.

Kaiwen Zhu, Jinjin Gu, Zhiyuan You, Yu Qiao, and Chao Dong. An intelligent agentic system for complex image restoration problems. In *International Conference on Learning Representations*, volume 2025, pp. 57985–58013, 2025.

Yushen Zuo, Qi Zheng, Mingyang Wu, Xinrui Jiang, Renjie Li, Jian Wang, Yide Zhang, Gengchen Mai, Lihong V Wang, James Zou, et al. 4kagent: agentic any image to 4k super-resolution. *arXiv preprint arXiv:2507.07105*, 2025.

A Supplementary Experiments

A.1 Qualitative Analysis of LAIR compared to other methods

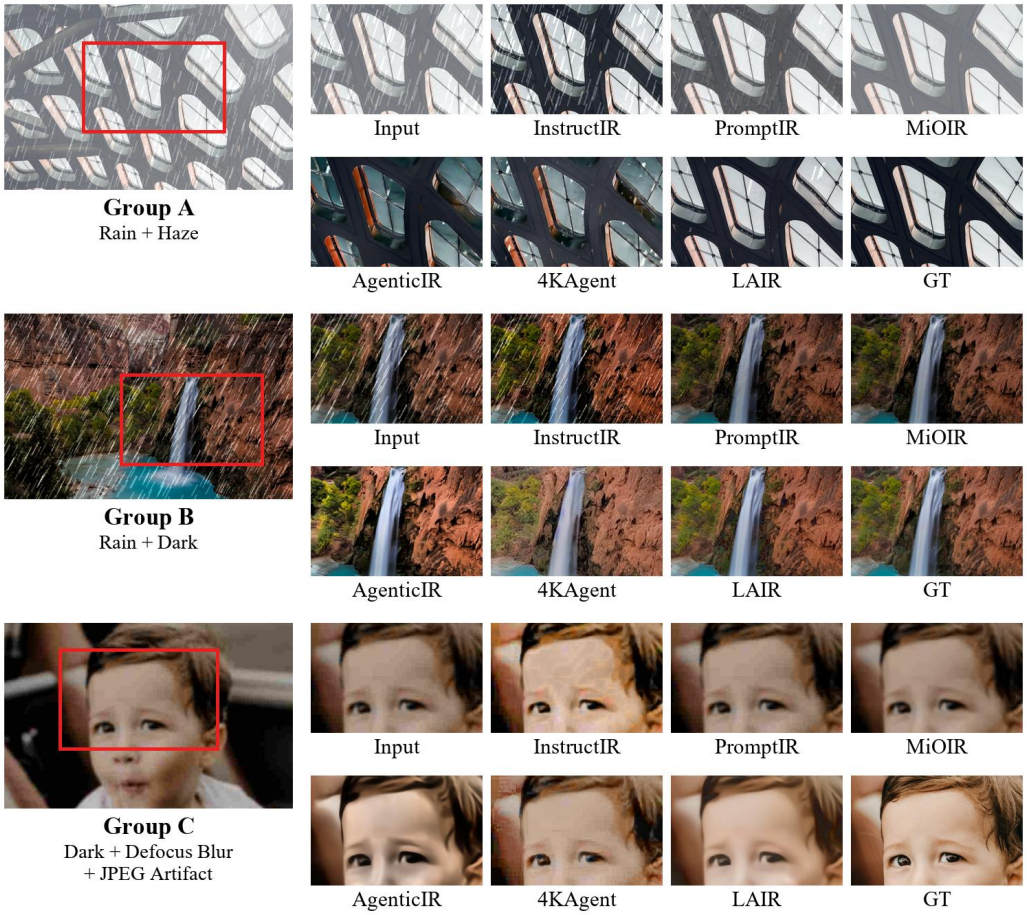


Figure 6: Qualitative comparison on representative samples from Groups A, B, and C. The red boxes indicate the enlarged regions. LAIR consistently handles compound degradations while preserving natural colors and structural fidelity, with fewer residual degradations and restoration artifacts than competing methods.

Fig. 6 presents qualitative comparisons on representative samples with compound degradations. For the rain-and-haze case in Group A, conventional all-in-one restoration methods (Potlapalli et al., 2023; Conde et al., 2024; Kong et al., 2024) leave visible rain streaks or residual haze, whereas AgenticIR and 4KAgent introduce noticeable color and contrast shifts. In contrast, LAIR effectively removes both degradations while recovering clean structural boundaries and maintaining an appearance close to the ground truth. For the rain-and-dark case in Group B, LAIR suppresses dense rain streaks and restores scene illumination without the over-enhancement of AgenticIR or the washed-out appearance of 4KAgent. In the more challenging Group C example involving darkness, defocus blur, and JPEG artifacts, several competing methods either retain substantial blur, introduce chromatic or blocking artifacts, or produce excessive smoothing. LAIR produces a cleaner and more natural facial appearance while better preserving the underlying structure.

Notably, all LAIR results shown in Fig. 6 are obtained using retrieved tool chains that satisfy the memory-reuse criterion, without invoking the fallback branch. Their consistent restoration quality across different compound degradations supports the practical utility of locality-aware memory

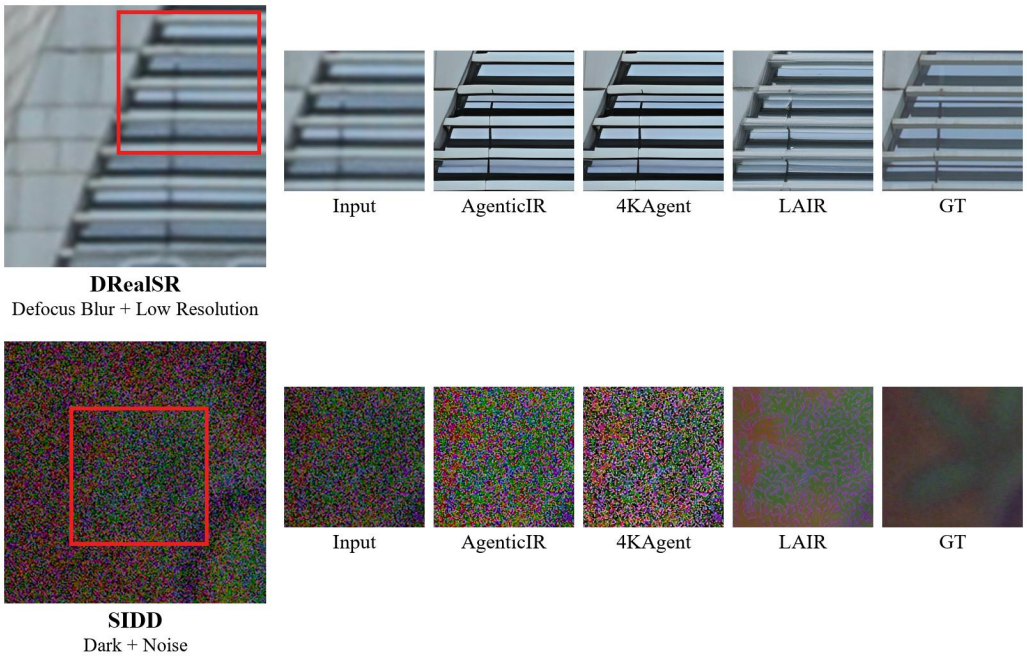


Figure 7: Qualitative comparison on real-world images from DRealSR and SIDD. The red boxes indicate the enlarged regions. LAIR provides a more balanced restoration by avoiding aggressive over-enhancement on DRealSR and substantially suppressing severe chromatic noise on SIDD.

reuse: restoration trajectories retrieved from similar inputs provide compatible tool-chain candidates that achieve a favorable balance between degradation removal and content fidelity.

We further evaluate the generalization ability of LAIR on real-world images from DRealSR (Wei et al., 2020) and SIDD (Abdelhamed et al., 2018). As shown in Fig. 7, AgenticIR and 4KAgent aggressively sharpen the DRealSR example, producing overly dark and high-contrast edges along the railings. LAIR yields a more conservative reconstruction whose overall color and contrast are closer to the ground truth. For the challenging low-light SIDD example, the competing agent-based methods retain or even amplify severe chromatic noise, whereas LAIR substantially suppresses the noise and recovers the underlying low-frequency appearance. These results indicate that the restoration trajectories retrieved by LAIR remain effective under complex real-world degradations, demonstrating the generalizability of its locality-aware memory-reuse strategy beyond synthetic degradation mixtures.

A.2 Memory Usage Analysis of LAIR

The qualitative examples above are produced through the memory-reuse branch. To quantify how frequently such retrieved memory entities can be directly reused, Table 6 reports the memory-hit and acceptance statistics over the complete test set.

Across the three synthetic splits, LAIR retrieves at least one degradation-compatible memory entity among the Top-5 memories for 64.24% of the queries. More importantly, 71.57% of these memory hits yield an accepted restoration tool chain, including 48.22% under the strict criterion and an additional 23.35% under the relaxed criterion. Consequently, memory reuse directly handles 662 out of $1{,}440$ images (45.97%) without invoking the fallback branch. These results indicate that the retrieved memories are not merely matched in terms of degradation labels, but frequently provide tool-chain candidates that are effective for the current input.

The relaxed criterion recovers an additional 216 cases after all strict candidates fail, increasing the end-to-end acceptance rate from 30.97% to 45.97% while still requiring a low residual degradation severity. Moreover, even for the more challenging three-degradation cases in Group C, 63.90% of

Table 6: Memory hit and acceptance statistics for the full method. A memory hit means that at least one of the retrieved Top-5 memories has the same degradation set as the query. Strict acceptance requires the selected memory chain to reach very-low severity. Relaxed acceptance selects the best low-severity memory chain after all strict candidates have been exhausted. Acceptance rates are conditioned on memory hits, while end-to-end acceptance rates use all images in each split.

Split	Memory Hit	Strict Accept	Relaxed Accept	Accept (incl. relaxed)	End-to-End Accept
Group A	447/640 (69.84%)	207/447 (46.31%)	114/447 (25.50%)	321/447 (71.81%)	321/640 (50.16%)
Group B	237/400 (59.25%)	141/237 (59.49%)	46/237 (19.41%)	187/237 (78.90%)	187/400 (46.75%)
Group C	241/400 (60.25%)	98/241 (40.66%)	56/241 (23.24%)	154/241 (63.90%)	154/400 (38.50%)
Overall	925/1440 (64.24%)	446/925 (48.22%)	216/925 (23.35%)	662/925 (71.57%)	662/1440 (45.97%)
Real-World	41/80 (51.25%)	17/41 (41.46%)	9/41 (21.95%)	26/41 (63.41%)	26/80 (32.50%)

the memory hits are successfully accepted, covering 38.50% of all Group C samples. On real-world images, LAIR retains a conditional acceptance rate of 63.41% and directly handles 32.50% of the inputs through memory reuse. Together, these results support the tool-effect locality prior underlying LAIR: memory entity retrieved from similar inputs often provide compatible and immediately reusable restoration tool chains.

B Detailed Implementation

B.1 Degradation-Parameter Prediction Objective

For each degradation type s, we define a degradation-parameter prediction objective to encourage the model to learn degradation-specific features. The regressor produces a $D_{\max} = 4$ dimensional output for every degradation token, while only the first d_s valid dimensions are supervised for token s. The ground-truth parameters are obtained either directly from the degradation synthesis process or from parameters manually specified during data generation.

For a raw parameter v with normalization interval $[l, u]$, its target is computed as

$$p = \frac{v - l}{u - l}. \tag{12}$$

The degradation-specific parameterizations and normalization intervals are summarized in Table 7. For Noise and Dark degradations, the parameters are defined in a unified manner to accommodate multiple noise and darkening operators, respectively.

B.1.1 Mapping of Noise Operators

Gaussian and Poisson noise are represented using a unified heteroscedastic noise model:

$$y = x + n, \qquad n \sim \mathcal{N}\left(0, \alpha x + \sigma_g^2\right), \tag{13}$$

where α controls the signal-dependent noise variance and σ_g denotes the standard deviation of the signal-independent Gaussian component.

For Gaussian noise with standard deviation σ, we set

$$(\alpha, \sigma_g) = (0, \sigma). \tag{14}$$

For Poisson noise with scale parameter s, we set

$$(\alpha, \sigma_g) = (1/s, 0). \tag{15}$$

Table 7: Degradation-specific parameterization used by the parameter-prediction objective. The intervals denote the raw ranges used for normalization. The regressor produces a four-dimensional output for every token, while only the first d_s dimensions are supervised.

Degradation	d_s	Parameters and normalization ranges	Description
Noise	2	$\alpha \in [0, 2]$, $\sigma_g \in [0, 80]$	α controls signal-dependent noise variance, while σ_g controls the standard deviation of signal-independent Gaussian noise. The unified noise model is $n(x) \sim \mathcal{N}(0, \alpha x + \sigma_g^2)$.
Motion blur	4	$c \in \{0, 1, 2\}$, $r \in [10, 15]$, $\sigma_m \in [3, 8]$, $\theta \in [-90^\circ, 90^\circ]$	c is the discrete severity level, r is the motion-kernel radius, σ_m is the standard deviation of the Gaussian weights along the motion trajectory, and θ is the motion direction.
Defocus blur	3	$c \in \{0, 1, 2\}$, $r \in [3, 6]$, $\sigma_a \in [0.1, 0.5]$	c is the discrete severity level, r is the radius of the disk kernel, and σ_a is the Gaussian smoothing parameter used to anti-alias the disk kernel.
Haze	2	$A \in [0.5, 1.0]$, $\beta \in [0.2, 3.0]$	A is the global atmospheric-light coefficient and β is the scattering coefficient. Given normalized depth d, the transmission is $t(x) = \exp(-\beta d(x))$.
Rain	3	$v \in [50, 100]$, $\ell \in [20, 40]$, $\theta \in [-30^\circ, 30^\circ]$, $w = 3$	v controls the density/intensity of the generated rain streaks, ℓ is the streak length, θ is the streak direction, and w is the streak width.
Dark	4	$a \in [0.1, 1.0]$, $\gamma \in [0.2, 1.0]$, $b \in [0, 0.5]$, $d_{\max} \in [0.1, 1.0]$	a is the multiplicative intensity factor, γ controls the nonlinear transformation, b is the subtractive offset, and $d_{\max}$ is the maximum output intensity. The mappings of the three operators are given in Equations 17–20.
JPEG Compression	1	$q \in [10, 30]$	q is the JPEG quality factor. A lower value corresponds to stronger compression and more severe compression artifacts.

Thus, Gaussian and Poisson noise correspond to the signal-independent and signal-dependent components of the unified noise model, respectively.

B.1.2 Mapping of Darkening Operators

The three darkening operators are represented using a unified parameter vector $(a, \gamma, b, d_{\max})$. Let $x \in [0, 1]$ denote the normalized value channel. The unified darkening model is

$$y = \mathrm{clip}\left(a\,\bar{x}^{\,1/\gamma} - b,\ 0,\ d_{\max}\right), \tag{16}$$

where a is a multiplicative intensity factor, γ controls the nonlinear intensity transformation, b is a subtractive offset, and $d_{\max}$ is the maximum output intensity.

For constant-shift and gamma-correction darkening, $\bar{x} = x$. For linear mapping, the value channel is first normalized as

$$\bar{x} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}. \tag{17}$$

The constant-shift operator is mapped as

$$(a, \gamma, b, d_{\max}) = (1,\ 1,\ \delta/255,\ 1). \tag{18}$$

The gamma-correction operator is mapped as

$$(a, \gamma, b, d_{\max}) = (1,\ \gamma,\ 0,\ 1). \tag{19}$$

The linear-mapping operator is mapped as

$$(a, \gamma, b, d_{\max}) = (m/255,\ 1,\ 0,\ m/255). \tag{20}$$

Here, δ denotes the intensity shift in the 8-bit value domain, and m denotes the maximum 8-bit output intensity.

B.2 Detailed pipeline of LAIR

This section describes the complete inference and online-update pipeline of the proposed **Locality-prior Agentic Image Restoration framework (LAIR)**. The procedure consists of degradation perception, locality-prior memory retrieval, verified trajectory reuse, planning fallback, and online memory update. In all experiments, we set $K = 5$.

B.2.1 Degradation perception and query construction

Given a degraded input image I, the perception model first estimates the severity of each predefined degradation type. Degradations whose predicted severity is at least medium ($\Gamma \leq 3$) are included in the active degradation set $\hat{D}_I$. For low-resolution inputs, super-resolution is additionally activated according to the image size.

DQ-MCE then extracts a structured representation

$$F_I = \{f_{I,1}, \ldots, f_{I,S}\}, \tag{21}$$

where each token encodes the degradation characteristics associated with one predefined degradation. The current implementation uses separate slots for noise, motion blur, defocus blur, haze, rain, darkness, and JPEG compression artifacts. As for low-resolution inputs, we use a separate super-resolution token. The super-resolution token is determined by the image size, consistent with the super-resolution activation in the active degradation set.

B.2.2 Top-5 locality-prior retrieval

For a query image, only the tokens associated with its active degradation set are used to compute memory relevance. Specifically, the retrieval score of a memory entry m_j is

$$S_{\mathrm{ret}}(I, m_j) = \frac{1}{|\mathcal{S}(\hat{D}_I)|} \sum_{s \in \mathcal{S}(\hat{D}_I)} \frac{f_{I,s}^{\mathsf{T}} f_{j,s}}{\|f_{I,s}\|_2 \|f_{j,s}\|_2}, \tag{22}$$

where $\mathcal{S}(\hat{D}_I)$ denotes the degradation-token slots activated by $\hat{D}_I$. Using a sum instead of the mean produces the same ranking because the number of active slots is fixed for a given query.

The five highest-scoring entries form the initial retrieval pool

$$\mathcal{R}_5(I) = \mathrm{Top5}_{m_j \in M}\, S_{\mathrm{ret}}(I, m_j). \tag{23}$$

B.2.3 Exact-degradation screening and grade-based ordering

A high feature similarity alone does not guarantee that the stored trajectory contains all restoration operations required by the query. We therefore retain only memories whose degradation set exactly matches that of the current image:

$$\mathcal{R}_{\mathrm{exact}}(I) = \left\{ m_j \in \mathcal{R}_5(I) \mid \hat{D}_j = \hat{D}_I \right\}. \tag{24}$$

Non-exact memories contribute to retrieval analysis but their tool chains are not directly executed in the reported top-5 setting.

The remaining candidates are sorted in descending order of their validated quality grades. Retrieval similarity determines which memories enter the top-5 pool, whereas the quality grade determines their execution order.

B.2.4 Independent execution of memory trajectories

For each candidate $m_j \in \mathcal{R}_{\mathrm{exact}}(I)$, LAIR restores the original input using its complete stored plan and tool chain:

$$I_j^{\mathrm{out}} = \mathcal{E}(I; P_j, T_j). \tag{25}$$

Every memory attempt starts from the original degraded image. Moreover, the order of both restoration subtasks and tools is preserved exactly as recorded in the historical trajectory.

The complete chain is executed before making the final acceptance decision. The perception model then evaluates the residual severity of every active degradation. Let

$$\Gamma_j = \min_{d \in \hat{D}_I} \gamma_{j,d}, \qquad \gamma_{j,d} \in \{1, 2, 3, 4, 5\}, \tag{26}$$

where larger values indicate less residual degradation. In particular, $\gamma = 5$ and $\gamma = 4$ correspond to very-low and low residual severity, respectively.

LAIR follows a strict-first verification strategy:

1. If a candidate reaches $\Gamma_j = 5$, it is accepted immediately and the remaining memories are not executed.
2. If $\Gamma_j = 4$, the candidate is retained as a relaxed fallback, but LAIR continues testing the remaining memories in search of a $\Gamma = 5$ result.
3. If $\Gamma_j < 4$, the candidate is rejected and the restoration state is reset to the original input.

If none of the five candidates reaches $\Gamma = 5$, all completed candidates with $\Gamma = 4$ are compared using the perception model's pairwise perceptual quality comparison, and the best one is returned. Consequently, LAIR does not simply accept the first satisfactory trajectory: it prioritizes complete degradation removal while retaining the best low-severity result as a safe fallback.

B.2.5 Planning fallback

If $\mathcal{R}_{\mathrm{exact}}(I)$ is empty, or if all exact-degradation memory candidates fail to reach $\Gamma \geq 4$, LAIR discards all intermediate states produced by memory attempts and returns to the original degraded input. It then invokes the planning and tool-selection procedure of AgenticIR (Zhu et al., 2025).

Specifically, the detected degradation set $\hat{D}_I$ is first converted into a set of restoration subtasks. An LLM determines their execution order using a static scheduling Experience:

$$P_I = \mathrm{LLM}\left(\hat{D}_I, \mathcal{X}_{\mathrm{static}}\right), \tag{27}$$

where $\mathcal{X}_{\text{static}}$ denotes the static **Experience** prompt inherited from AgenticIR (Zhu et al., 2025). Deepseek-v4-pro (DeepSeek-AI, 2026) is used as LLM here. This Experience summarizes general relationships among restoration subtasks, such as which subtask should be performed first when multiple degradations coexist.

The static Experience is extracted exclusively from the training set. For each selected multi-degradation training image, AgenticIR exhaustively executes all permutations of the required restoration subtasks and all combinations of their corresponding tools. The terminal results are evaluated by the perception model, and the restoration failure rates are aggregated according to the subtask order, marginalizing over individual tool choices and training images. These statistics are subsequently summarized by an LLM into general task-ordering principles, which form $\mathcal{X}_{\text{static}}$. Once extracted, this prompt is fixed and is not updated using validation or test images. **Please note that** this process has the same offline search complexity as the construction of our initial memory bank, since both exhaustively enumerate all task permutations and their associated tool combinations on the training set.

Conditioned on the current degradation set and the static scheduling Experience, the planner produces an ordered restoration plan

$$P_I = [p_{I,1}, p_{I,2}, \ldots, p_{I,L}]\,. \tag{28}$$

The plan is then executed sequentially. For each subtask $p_{I,l}$, the original AgenticIR tool-selection policy evaluates the corresponding restoration tools and selects an appropriate tool $t_{I,l}$. This produces the tool chain

$$T_I = [t_{I,1}, t_{I,2}, \ldots, t_{I,L}]\,, \tag{29}$$

and the final output is obtained as

$$I^{\text{out}} = \mathcal{E}\left(I; P_I, T_I\right). \tag{30}$$

Online memory update After either memory reuse or planning fallback produces the final output, LAIR validates the complete execution trajectory before storage. A trajectory is eligible only if its plan and tool chain are complete and aligned, its worst residual severity satisfies $\Gamma \geq 4$, and its no-reference quality scores are valid. In the reported experiment, we additionally require $\text{MUSIQ} \geq 35$ and $\text{CLIP-IQA} \geq 0.35$.

For an accepted trajectory, its perceptual quality component is

$$G_{\text{perc}} = \frac{1}{2}\left(\text{CLIP-IQA} + \frac{\text{MUSIQ}}{100}\right). \tag{31}$$

The trajectory is hierarchically ranked first by Γ and then by G_{perc}. The validated entry stores the input degradation set, the DQ-MCE representation of the original degraded image, the executed plan, the complete tool chain, its quality grade, and the input severity evaluation:

$$m_I = \left(\hat{D}_I, F_I, P_I, T_I, G_I\right). \tag{32}$$

B.3 DQ-MCE Training Details

B.3.1 Training data and pair construction

We use seven degradation queries, corresponding to noise, motion blur, defocus blur, haze, rain, low light, and JPEG compression. We use the train set of the MiOIR dataset (Kong et al., 2024) as the source of clean images to ensure that the training data is not contaminated by any of the test images used in our restoration evaluation. The training set contains 3,450 clean source images, from which 21,387 non-overlapping 512×512 crops are extracted. We synthesize eight parameter variants for each clean-crop–degradation-sequence pair, resulting in 101,241 groups and 809,928 degraded images covering 113 ordered degradation sequences. The order is retained because different orders of applying multiple degradations can produce different image distributions.

Each degraded image is used once as an anchor. Its positive is selected from images generated with the same ordered degradation sequence but from a different clean source image. Among all candidates satisfying these two conditions, we choose the one with **the smallest Euclidean distance between the concatenated normalized degradation-parameter vectors** described above. This

nearest- parameter rule avoids treating two substantially different manifestations of the same categorical degradation combination as a positive pair. If an image contains multiple degradation types, the same paired image provides a positive for every active query token.

B.3.2 Encoder and prediction heads

We initialize both branches with a DINOv3 ViT-L/16 (Siméoni et al., 2025) encoder pretrained on LVD-1689M. For a 512×512 input, the encoder yields a 32×32 grid of 1,024 patch tokens after removing the class token and four register tokens. Each token has dimension $d = 1{,}024$. The seven learnable degradation queries are ℓ_2-normalized before being passed to an eight-head cross-attention layer. The shared projection head is a two-layer MLP, $1{,}024 \to 512 \to 512$, with a ReLU between the two linear layers. Its output is normalized row-wise, giving a $D = 512$ embedding for each degradation query.

Parameter prediction uses a MLP, $512 \to 128 \to 4$, with a ReLU hidden activation and a sigmoid output. Although the regressor has four outputs, only the first d_s entries are supervised for degradation type s, where $(d_s)_{s=1}^{7} = (2, 4, 3, 2, 3, 4, 1)$. Thus, padded entries do not contribute to the parameter loss.

All scalar parameters are linearly mapped to $[0, 1]$ before regression. The parameterization and normalization intervals are summarized in Table 7. We unify the two noise types using the heteroscedastic noise model and the three low-light operators using the unified darkening model described above.

Token-wise negatives and objectives For an active token s, negatives comprise both the momentum embeddings of in-batch images in which degradation s is absent and historical momentum embeddings from a slot-specific FIFO queue. A negative may share other degradation types with the anchor; it is required only not to contain the degradation associated with token s. Each distributed worker maintains one queue of $K = 8,192$ keys for each degradation type. Queues are initially empty and are updated after each forward pass using detached momentum-branch embeddings from inactive samples. Consequently, no gradient is propagated through either the momentum keys or the queued negatives.

Let $\mathcal{A}$ denote the set of active sample–token pairs in a minibatch, where i indexes samples and s indexes degradation tokens. For each $(i, s) \in \mathcal{A}$, $f^q_{i,s}$ is the online embedding, $f^{pos}_{i,s}$ is the corresponding positive momentum embedding, and $\mathcal{N}_{i,s}$ contains the in-batch and queued negative momentum embeddings.

Following the notation in the main text, the token-wise contrastive loss is

$$\mathcal{L}^{\text{InfoNCE}}_{i,s} = -\log \frac{\exp\left(\text{sim}(f^q_{i,s}, f^{pos}_{i,s})/\tau\right)}{\exp\left(\text{sim}(f^q_{i,s}, f^{pos}_{i,s})/\tau\right) + \sum_{z \in \mathcal{N}_{i,s}} \exp\left(\text{sim}(f^q_{i,s}, z)/\tau\right)}. \tag{33}$$

Let $p_{i,s}$ and $\hat{p}_{i,s}$ denote the normalized ground-truth and predicted parameter vectors, respectively, restricted to the d_s valid dimensions for degradation type s. The parameter-prediction loss is

$$\mathcal{L}^{\text{MSE}}_{i,s} = \|p_{i,s} - \hat{p}_{i,s}\|_2^2 . \tag{34}$$

The complete training objective averages both loss terms over all active sample–token pairs:

$$\mathcal{L} = \frac{1}{|\mathcal{A}|} \sum_{(i,s) \in \mathcal{A}} \mathcal{L}^{\text{InfoNCE}}_{i,s} + \frac{\lambda}{|\mathcal{A}|} \sum_{(i,s) \in \mathcal{A}} \mathcal{L}^{\text{MSE}}_{i,s}, \tag{35}$$

where $\lambda = 50$ balances the contrastive and parameter-prediction objectives, and $\tau = 0.2$ is the temperature parameter. The contrastive objective is evaluated only in the online-to-momentum direction.

Optimization. The pretrained DINOv3 weights are frozen, and rank-8 LoRA adapters with scaling factor $\alpha_{\text{LoRA}} = 16$ and zero dropout are inserted into the query, key, value, and output projections of all 24 transformer blocks. The LoRA parameters and the complete query/projection/regression head of the online branch are optimized with AdamW. We use learning rates of 5×10^{-4} for LoRA and 10^{-3} for the heads, weight decay 10^{-4}, a linear warm-up over

the first 0.05 epoch, and cosine learning rate decay over the train stage. Training uses bfloat16 mixed precision on eight NVIDIA RTX A6000 GPUs with 16 images per GPU, giving an effective batch size of 128. The momentum coefficient is $m = 0.999$. The momentum branch is initialized as an exact copy of the online branch and is updated after every optimizer step. We use random seed 42 and the checkpoint at 12,000 optimization steps for the reported experiments.

B.4 Memory-Reuse Settings for 4KAgent

We integrate LAIR into 4KAgent (Zuo et al., 2025) as a plug-in memory-reuse branch between its proposal and tool-execution stages. The memory representation, Top-K retrieval, exact-degradation screening, and grade-based candidate ordering remain identical to those described in Section B.2, with $K = 5$. Specifically, after 4KAgent produces its degradation assessment and initial restoration plan, we map the planned subtasks to the degradation categories used by LAIR and retrieve candidate trajectories from the same online memory bank. The stored task order and tool chain of each retained candidate are then executed from the original degraded input using the 4KAgent toolbox. Under the adopted profile, this toolbox contains 43 task-specific tool candidates across eight restoration subtasks, compared with 35 candidates in our unified toolbox.

The main adaptation concerns trajectory verification and planning fallback. Instead of applying the residual-severity verification rule in Section B.2.4, we use the native 4KAgent reflection criterion. Every intermediate output of a retrieved trajectory is re-evaluated using HPSv2+metric, and a step is considered successful when its reflection score is at least 0.5. A retrieved trajectory is accepted only if all of its steps satisfy this criterion. No stored quality score or cached success decision is used for acceptance.

If no exact-degradation trajectory is retrieved, or any restoration step fails verification, all outputs from the memory branch are discarded. We then restore the 4KAgent state recorded immediately after its proposal stage, including its random-number-generator state, and resume the original execution procedure with its initial plan, rollback, and rescheduling mechanisms unchanged. Thus, memory reuse serves only as an optional shortcut and does not modify the behavior of the native fallback. After either branch finishes, a trajectory accepted by the native reflection criterion is stored online using the same memory format as LAIR, with the minimum reflection score along the trajectory used as its quality grade.