REGIONAGENT: TWO-STAGE PREFERENCE OPTIMIZATION FOR REGIONAL IMAGE RESTORATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Images under non-uniform coupled degradations require diverse restoration capabilities for different degraded regions. Existing methods employ globally uniform operations, potentially overlooking regional differences and causing incomplete restoration or even damage to image content. Moreover, a degraded region may require successive restoration actions, and earlier actions may alter residual degradation characteristics or introduce artifacts, potentially hindering subsequent restoration. To address these challenges, we propose RegionAgent, an adaptive restoration framework that trains a Vision-Language Model (VLM) through two-stage preference learning to generate a region-specific restoration action at each iteration. Specifically, we first train the VLM to select an action suited to the current region through step-level Direct Preference Optimization (DPO), using preferences derived from immediate restoration feedback to supervise only the differing component in each action pair. To account for the effects of current actions on subsequent restoration, we then apply trajectory-level DPO, using preferences derived from complete trajectory scores to supervise the sole differing action in each trajectory pair. Furthermore, as conventional image-pair datasets do not directly provide preferences over regional restoration decisions, we construct a regional restoration preference dataset comprising 10K action pairs and 20K trajectory pairs for two-stage VLM training. Experiments across multiple datasets demonstrate strong restoration performance under both mixed and single degradations. Ablation studies further validate the effectiveness of two-stage preference learning in improving restoration quality.

1 INTRODUCTION

Image restoration aims to recover clean images from observations affected by degradations such as blur, haze, rain, or low light. Deep learning-based methods have advanced this task through specialized models for individual degradation types (Ren et al., 2021; Zamir et al., 2021; Yan et al., 2025) and all-in-one networks for multiple restoration tasks (Li et al., 2022; Potlapalli et al., 2023) (Figure 1(a)). However, real-world images can contain coupled degradations that vary in type and severity across regions. Applying uniform restoration operations to the whole image may overlook regional differences and damage unaffected content. Within an affected region, removing one degradation can leave others unresolved, requiring successive restoration operations. The effectiveness of later operations then depends on how earlier ones change the image (Chen et al., 2024; Zhu et al., 2025; 2026).

To make restoration more adaptive, instruction-guided methods allow users to specify restoration goals (Conde et al., 2024) and target regions (Liu et al., 2025). Restoration agents automate tool selection and revise image-level plans using restoration feedback (Chen et al., 2024; Zhu et al., 2025; Lin et al., 2025) (Figure 1(b)). As an alternative to sequential tool invocation, Beyond Sequential Tools learns image-level expert weights through preference optimization for single-pass restoration (Xiong et al., 2026). However, these approaches focus on user-specified regions or image-level decisions, rather than learning to select regions and configure their restoration autonomously.

We therefore introduce *RegionAgent*, an adaptive restoration framework in which a Vision-Language Model (VLM) selects regions and configures their restoration from the current image state (Figure 1(c)). At each iteration, a localization network predicts a heatmap of residual degradations and

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.

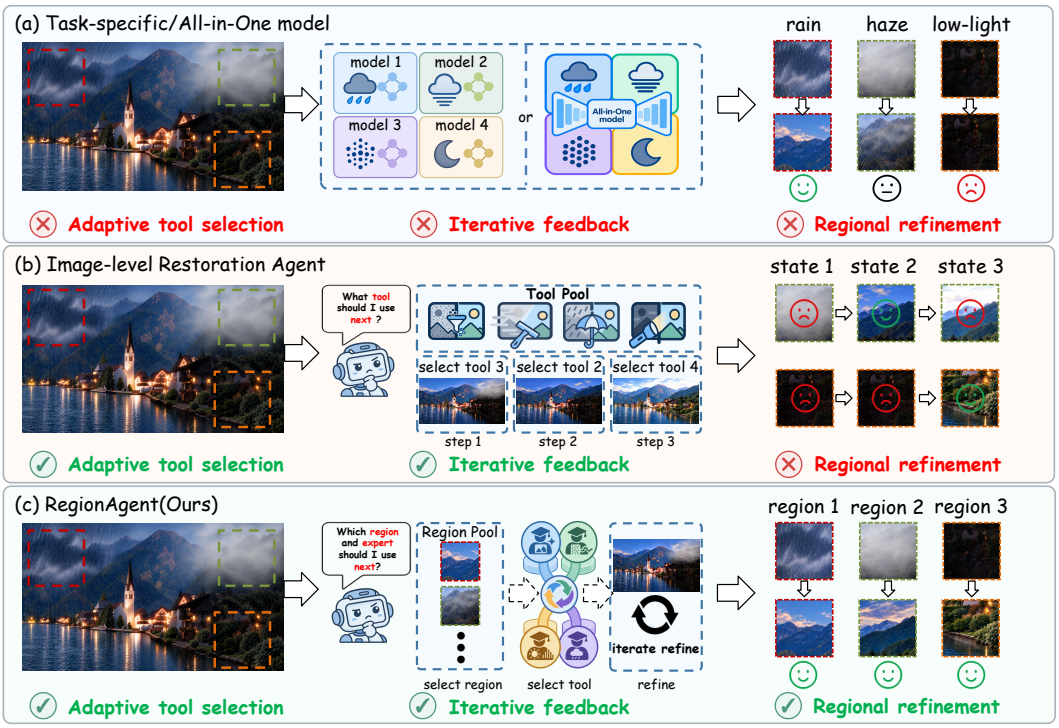

Figure 1: Conceptual comparison of restoration paradigms. (a) Task-specific and all-in-one models restore the whole image. (b) Image-level agents select tools using iterative feedback. (c) RegionAgent selects regions and expert configurations for adaptive local refinement.

uses it to propose candidate regions (Figure 2). The VLM then selects one of these regions and specifies its restoration configuration based on the current image.

Each refinement action specifies a target region, weights for combining restoration experts, and the strength of the local correction. We train the VLM through two stages of Direct Preference Optimization (DPO) (Rafailov et al., 2023): step-level DPO followed by trajectory-level DPO. Step-level DPO learns which configuration choices suit the current region by comparing their immediate restoration effects, applying the preference loss only to the component that differs between paired actions. Trajectory-level DPO then trains the VLM to account for the effects of current choices on subsequent restoration. It uses complete-trajectory scores to supervise the sole differing action between paired trajectories. Since conventional image-pair datasets lack preferences over regional restoration decisions, we construct a dataset of action and trajectory preference pairs for two-stage training.

Executing these actions produces restored crops whose direct writeback can introduce boundary artifacts and structural discontinuities. Our Degradation- and Structure-Guided Residual Fusion Module (DSRF) integrates the difference between each restored crop and its current counterpart as a residual correction. Spatial masks regulate this correction according to restoration needs and image structure, while Laplacian pyramid blending promotes smooth transitions across scales. The VLM then generates its next action from the fused image.

Experiments across multiple datasets demonstrate strong restoration performance under both mixed and single degradations. Our contributions can be summarized as follows:

- We introduce RegionAgent, an adaptive restoration framework whose VLM generates regional actions from the current image state to address non-uniform coupled degradations.
- We develop two-stage DPO: step-level DPO adapts restoration configurations to current regional needs, while trajectory-level DPO uses complete-trajectory feedback to guide current restoration actions.

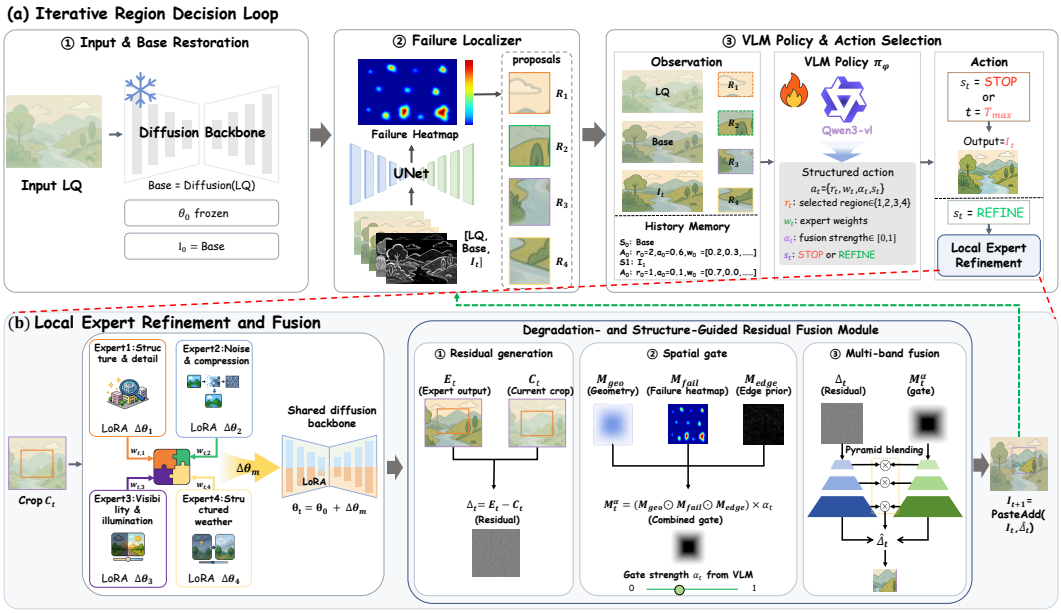

Figure 2: Overview of RegionAgent. (a) The Localizer proposes candidate regions, and the VLM selects restoration actions from the current image state. (b) The composed restorer processes the selected crop, and DSRF blends its residual correction into the image for subsequent refinement.

- We construct a regional restoration preference dataset comprising 10K action pairs and 20K trajectory pairs. It includes controlled comparisons that vary a single configuration component or a single trajectory action, providing targeted supervision for regional restoration decisions beyond conventional degraded–clean image pairs.

2 Related Work

Adaptive Image Restoration. All-in-one models condition shared restoration networks on degradation representations and learned prompts (Li et al., 2022; Potlapalli et al., 2023), while vision-language approaches incorporate visual-textual features for degradation diagnosis and restoration guidance (Luo et al., 2024; Ai et al., 2024; Jiang et al., 2024). Instruction-guided methods extend this control to user-defined goals (Conde et al., 2024) and region-specific restoration (Liu et al., 2025). Restoration agents automate tool selection and sequencing (Yu et al., 2018), using multimodal diagnosis and quality feedback to revise their plans (Chen et al., 2024; Zhu et al., 2025; Lin et al., 2025). Recent systems integrate restoration with assessment (Chen et al., 2026) or couple planners with LoRA-based tools (Li et al., 2026a; Liu et al., 2026), while OPERA jointly optimizes planning and tool execution using restoration rewards (Zhu et al., 2026). As an alternative to sequential execution, Beyond Sequential Tools learns image-level expert weights for single-pass restoration through a DPO-trained allocation branch (Xiong et al., 2026). RegionAgent instead selects regions and configures their restoration autonomously from the updated image at each iteration.

Preference Optimization. Preference optimization applies to restoration outputs and decision policies. DSPO aligns super-resolution generators with semantic-instance preferences (Cai et al., 2025), whereas TTPO guides diffusion sampling with test-time preferences without model training (Li et al., 2026b). Within policy outputs, fine-grained objectives weight tokens or target localized differences (Christopoulou et al., 2025; Hasan et al., 2025), and DiTTo separates preference supervision across degradation reasoning, restoration order, and tool calls (Choi & Oh, 2026). For sequential decisions, Hierarchical Preference Learning combines trajectory-, step-, and action-group objectives (Gao et al., 2026), while Critical Step Optimization uses successful policy continuations to verify alternative actions for action-level DPO (Li et al., 2026c). Our refinement pairs vary one configuration component from a shared image state; trajectory pairs vary one action while fixing the region and other actions. The losses target the differing component and full action, respectively.

3 METHOD

3.1 PROBLEM FORMULATION

Given a low-quality image I_{LQ} of height H and width W, a pretrained shared restorer E_{θ_0} with frozen parameters θ_0 produces the fixed reference $I_{\mathrm{Base}} = E_{\theta_0}(I_{\mathrm{LQ}})$ and initializes the current image as $I_0 = I_{\mathrm{Base}}$. At iteration t, the VLM selects an action $a_t \in \{\texttt{STOP}\} \cup \{\texttt{REFINE}(r_t, \mathbf{w}_t, \alpha_t)\}$ from the current restoration state. A refinement action specifies a region r_t from the candidate region set $\mathcal{R}_t$, expert combination weights $\mathbf{w}_t$, and fusion strength α_t, with the latter two taking discrete values. The process returns the current image upon `STOP` or at the iteration limit $T_{\max}$. Model details and action settings are provided in Appendices A.1 and A.4.

We first describe regional decisions and execution (Secs. 3.2–3.3), then preference construction and two-stage DPO for learning these decisions (Secs. 3.4–3.5).

3.2 ITERATIVE REGIONAL DECISION-MAKING

We train a lightweight U-Net Localizer f_ω, with parameters ω, to localize residual degradations. To cover different restoration stages, we derive training targets from discrepancies between degraded inputs or intermediate restoration results and their corresponding clean images.

After training, we freeze ω and predict the failure heatmap $F_t = f_\omega(X_t) \in [0,1]^{H\times W}$ at each iteration from the input tensor $X_t = \mathrm{cat}[I_{\mathrm{LQ}}, I_{\mathrm{Base}}, I_t, |I_{\mathrm{LQ}} - I_t|, |I_{\mathrm{Base}} - I_t|]$, where I_t is the current image and cat denotes channel-wise concatenation. Higher heatmap values indicate a greater need for further restoration. We rank square windows of different sizes by their mean heatmap values and apply non-maximum suppression to select four new proposals. The candidate region set $\mathcal{R}_t$ contains these proposals and, for $t > 0$, the previously selected region r_{t-1}, with duplicate boxes included once.

Given the observation $o_t = (I_{\mathrm{LQ}}, I_{\mathrm{Base}}, I_t, \mathcal{R}_t, \mathcal{H}_t)$, the VLM policy π_φ with parameters φ selects $a_t \sim \pi_\varphi(\cdot \mid o_t)$, where $\mathcal{H}_t$ is the structured-action history. After refinement produces I_{t+1}, we update the heatmap and proposals while retaining r_t, allowing the VLM to continue refining it or select another region (Figure 2(a)). Localizer and action details appear in Appendices A.2 and A.4.

3.3 REGIONAL REFINEMENT

To process candidate regions of different sizes at a common input resolution, we construct a crop C_t of size $s_p \times s_p$. The selected region r_t occupies a central core of size $s_c \times s_c$, while the surrounding halo provides structural and textural context and a transition area for fusion. An affine transform B_t maps r_t to this core. Using the same transform, the sampling operator Sample extracts the current-image crop $C_t = \mathrm{Sample}(I_t; B_t, s_p)$ and the spatially aligned failure map $\widetilde{F}_t = \mathrm{Sample}(F_t; B_t, s_p)$.

Each expert is a LoRA update $\Delta\theta_k$ to the shared backbone (Hu et al., 2022). The executor combines them as $\Delta\theta_m = \sum_k w_{t,k}\Delta\theta_k$ (Xiong et al., 2026), where k indexes experts, $w_{t,k}$ is the corresponding VLM-selected weight in $\mathbf{w}_t$, and m denotes the mixture. The composed parameters $\theta_t = \theta_0 + \Delta\theta_m$ produce the restored crop $E_t = E_{\theta_t}(C_t)$ in one forward pass. The residual $\Delta_t = E_t - C_t$ lets fusion control where and how strongly this correction changes the current image.

Directly adding the crop residual can unnecessarily alter well-restored pixels and create abrupt transitions to surrounding content. We therefore design the ***Degradation- and Structure-Guided Residual Fusion Module*** (DSRF), which combines three masks to regulate corrections according to spatial position, predicted restoration need, and local structural changes. The geometric mask M_{geo} equals one in the core and smoothly decays to zero through the halo, providing a gradual transition to surrounding content. The failure mask M_{fail} weights corrections by the restoration need predicted in $\widetilde{F}_t$, using C_t-guided filtering to align mask boundaries with image structure (He et al., 2013). The edge mask M_{edge} further reduces correction weights at halo pixels with large gradient-magnitude differences between E_t and C_t, while remaining one in the core to permit structural correction. Together, these masks define the spatial gate $M_t^\alpha = \alpha_t(M_{\mathrm{geo}} \odot M_{\mathrm{fail}} \odot M_{\mathrm{edge}})$, where α_t controls fusion strength, each mask lies in $[0,1]$, and $\odot$ denotes element-wise multiplication.

To blend appearance and detail across scales, DSRF uses this spatial gate to weight the Laplacian pyramid of the crop residual and reconstructs the fused residual $\widehat{\Delta}_t = \mathrm{Recon}\{G_\ell(M_t^\alpha) \odot L_\ell(\Delta_t)\}_{\ell=0}^{J-1}$. Here, L_ℓ and G_ℓ extract Laplacian and Gaussian pyramid levels indexed by ℓ, J is the number of levels, and Recon reconstructs the pyramid (Burt & Adelson, 1983). Residual writeback gives $I_{t+1} = \mathrm{PasteAdd}(I_t, \widehat{\Delta}_t) = I_t + \mathcal{W}_{B_t}^{-1}(\widehat{\Delta}_t)$, where $\mathcal{W}_{B_t}^{-1}$ maps the crop residual back to the image with zero support outside the sampled context. Sampling, mask, and pyramid details are given in Appendix A.3.

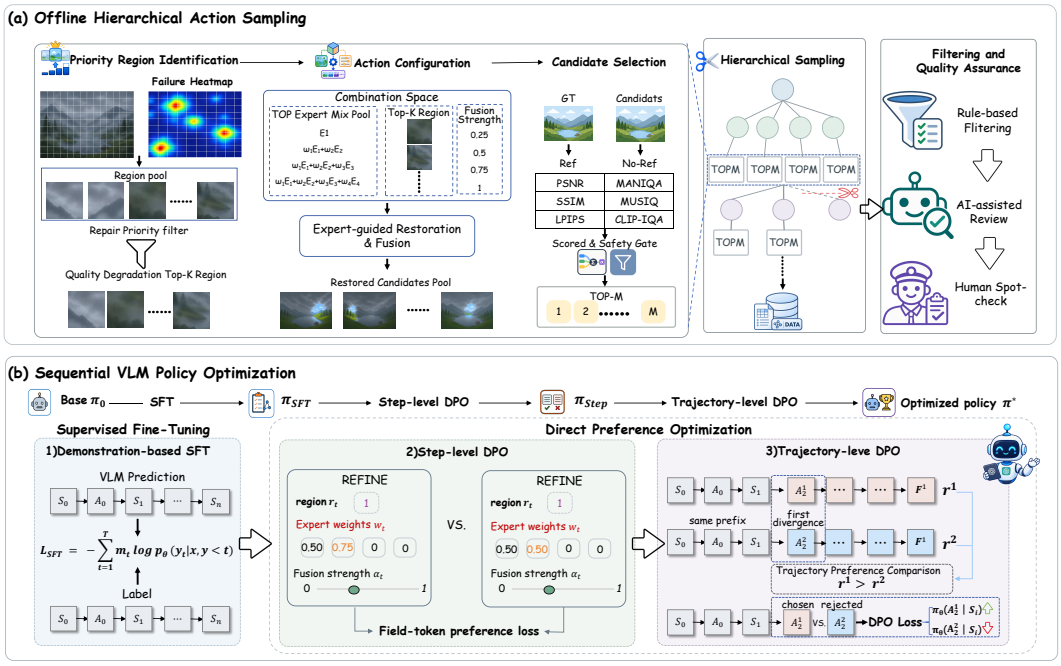

Figure 3: Supervision construction and policy optimization. (a) Action sampling, scoring, trajectory expansion, and filtering produce demonstrations and preference pairs. (b) Supervised fine-tuning and two-stage DPO train regional decisions.

3.4 Supervision Construction

Conventional image-pair datasets provide degraded images and clean targets, but do not directly provide preferences over regional restoration decisions. We therefore construct a regional restoration preference dataset containing action and trajectory pairs to support two-stage DPO.

To control search cost, we first restore each newly proposed region with one expert at a time, with every trial starting from the same current image. We fuse each restored crop at the allowed fusion strengths and score the resulting images using the one-step utility U_{step}, which measures restoration gains relative to the current image after accounting for penalties. We rank regions by their highest utility across these trials and expand expert configurations in the top two regions. Sampling details are provided in Appendix B.2.

The one-step utility U_{step} uses three full-reference metrics (PSNR, SSIM, and LPIPS) and three no-reference metrics (MANIQA, MUSIQ, and CLIP-IQA). We normalize and clip each metric gain separately. For full-reference metrics, we equally combine full-image and selected-region gains, whereas no-reference metrics use full-image gains. We average the resulting gains across the six metrics and subtract penalties for halo degradation, boundary artifacts, action cost, repeated states, and degradation below Base. Complete scoring definitions are provided in Appendix B.3.

Beam search then builds complete restoration sequences with region coordinates held fixed, updating successor heatmaps and admitting successors only when their one-step utilities exceed a minimum threshold. This fixed-region constraint applies to offline trajectory construction, while inference selects regions dynamically (Sec. 3.2). For a trajectory τ_t ending at I_t after t refinement steps, we

define its normalized quality gain $Q(I_t)$ and penalized trajectory score $r(\tau_t)$ as

$$Q(I_t) = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \mathrm{clip}\left(\frac{d_m[m(I_t) - m(I_{\mathrm{Base}})]}{s_m + \epsilon}, -1, 1 \right),$$

$$r(\tau_t) = Q(I_t) - \lambda_{\mathrm{out}} \sum_{i<t} P_i^{\mathrm{out}} - \lambda_{\mathrm{bnd}} \sum_{i<t} P_i^{\mathrm{bnd}} - \lambda_{\mathrm{len}} t - \lambda_{\mathrm{rep}} N_{\mathrm{rep}} - \lambda_{\mathrm{base}} P_{\mathrm{base}}.$$

Here, $\mathcal{M}$ contains the six metrics above, and $m(I)$ is the full-image score for metric m, using the corresponding clean image for full-reference metrics. The sign $d_m \in \{-1, +1\}$ aligns improvement, s_m is the metric's normalization scale, and $\epsilon > 0$ ensures numerical stability.

At step i, P_i^{out} averages positive increases in absolute error to the clean image over the halo, the band inside the writeback support but outside the core. Similarly, P_i^{bnd} penalizes increased gradient error to the clean image near the core boundary. The term N_{rep} counts repeated states along the trajectory, and P_{base} penalizes degradation of the final image relative to Base. The λ terms weight these penalties and refinement count t. This score compares trajectories ending with `STOP` or reaching $T_{\max}$. Calibration and penalty settings are specified in Appendix B.3.

We form step-level pairs from same-state actions. To construct a trajectory pair, we replace one refinement action in a completed search trajectory while keeping its target region fixed, then execute the unchanged subsequent actions from the resulting image state. The original and modified trajectories therefore differ in exactly one action, while subsequent image states may differ. Filtering enforces pairing constraints and executability, removes duplicates, and balances degradation and decision coverage. Pairs meeting score-gap thresholds are selected automatically; low-margin or metric-conflicting cases undergo VLM-assisted screening and human verification. The best trajectories form the demonstration set $\mathcal{D}_{\mathrm{SFT}}$ for supervised fine-tuning (SFT), and retained action and trajectory pairs form $\mathcal{D}_{\mathrm{step}}$ and $\mathcal{D}_{\mathrm{traj}}$. Appendices B.1, B.2, and B.4 detail data sources and splits, search settings, and pair selection.

3.5 Preference Optimization

SFT first teaches the VLM to generate correctly formatted, executable actions from demonstrations. Step-level and trajectory-level DPO then train regional configuration choices and decisions that account for subsequent restoration, respectively.

Both DPO stages use a common objective with stage-specific labels and token masks. For action a under observation o, let y_j denote its j-th serialized token, $y_{<j}$ its full token prefix, and $\mathcal{T}(a)$ the supervised positions. The selected-token log-probability ratio $z_\pi^{\mathcal{T}}$ and DPO loss for preferred/rejected actions (a^+, a^-) are

$$z_\pi^{\mathcal{T}}(a \mid o) = \sum_{j \in \mathcal{T}(a)} \log \frac{\pi(y_j \mid o, y_{<j})}{\pi_{\mathrm{ref}}(y_j \mid o, y_{<j})},$$

$$\mathcal{L}_{\mathrm{DPO}}(\mathcal{T}) = -\mathbb{E} \log \sigma\big(\beta[z_\pi^{\mathcal{T}}(a^+ \mid o) - z_\pi^{\mathcal{T}}(a^- \mid o)]\big),$$

where $\pi = \pi_\varphi$ is the trainable VLM policy, π_{ref} is its frozen reference, β controls preference strength, and σ is the logistic function (Rafailov et al., 2023). We sum selected-token log-probability ratios without length normalization. The expectation averages retained pairs from the current stage's dataset; trajectory pairs supply the shared observation and alternative actions defined below.

Step-level DPO learns configurations suited to current regional needs from paired `REFINE` actions that share an observation and differ in one field: region, expert combination weights, or fusion strength. Their one-step utilities determine the preference, and $\mathcal{T}$ selects that field's value tokens, including the full expert-weight vector for weight comparisons. For `STOP`–`REFINE` pairs, complete-trajectory scores compare immediate termination with continued restoration, while $\mathcal{T}$ selects only the `action` value tokens. The frozen SFT checkpoint supplies the reference policy.

Trajectory-level DPO evaluates a current decision through its effects on subsequent restoration of the same region. For $(\tau^+, \tau^-) \in \mathcal{D}_{\mathrm{traj}}$ with $r(\tau^+) > r(\tau^-)$, let $t^\dagger$ be their sole differing iteration, so $a_t^+ = a_t^-$ for $t \neq t^\dagger$. The shared prefix supplies the same observation $o_{t^\dagger}$, and $\mathcal{T}$ selects every token of the alternative actions at $t^\dagger$. With the region and remaining action configurations held

Table 1: Quantitative comparison on the MiO100 mixed-degradation benchmark. Best and second-best results in each group are shown in **bold** and underlined, respectively.

Degradations	Method	Venue	PSNR↑	SSIM↑	LPIPS↓	MANIQA↑	CLIP-IQA↑	MUSIQ↑
Group A	AirNet (2022)	CVPR	19.13	0.6019	0.4283	0.2581	0.3930	42.46
	PromptIR (2023)	NeurIPS	20.06	0.6088	0.4127	0.2633	0.4013	42.62
	MiOIR (2024)	arXiv	20.84	0.6558	0.3715	0.2451	0.3933	47.82
	DA-CLIP (2024)	ICLR	19.58	0.6032	0.4266	0.2418	0.4139	42.51
	InstructIR (2024)	ECCV	18.03	0.5751	0.4429	0.2660	0.3528	45.77
	AdaIR (2025)	ICLR	19.06	0.6061	0.4170	0.2718	0.4117	44.66
	DA-RCOT (2025)	TPAMI	18.82	0.5960	0.4303	0.2641	0.4058	43.42
	AutoDIR (2024)	ECCV	19.64	0.6286	0.3967	0.2500	0.3767	47.01
	AgenticIR (2025)	ICLR	21.04	0.6818	0.3148	0.3071	0.4474	56.88
	MAIR (2026)	IJCV	21.02	0.6715	0.2963	0.3330	0.4751	59.19
	4KAgent (2025)	NeurIPS	21.48	0.6720	0.3019	0.3748	0.5544	63.19
	RAR (2026)	CVPR	20.46	0.7144	**0.1299**	0.4659	**0.6566**	57.19
	RegionAgent (Ours)	Ours	**22.11**	**0.7150**	0.2377	**0.5834**	0.5652	**67.41**
Group B	AirNet (2022)	CVPR	19.31	0.6567	0.3670	0.2882	0.4274	47.88
	PromptIR (2023)	NeurIPS	20.47	0.6704	0.3370	0.2893	0.4289	48.10
	MiOIR (2024)	arXiv	20.56	0.6905	0.3243	0.2638	0.4330	51.87
	DA-CLIP (2024)	ICLR	18.56	0.5946	0.4405	0.2435	0.4154	43.70
	InstructIR (2024)	ECCV	18.34	0.6235	0.4072	0.3022	0.3790	50.94
	AdaIR (2025)	ICLR	19.24	0.6637	0.3467	0.3001	0.4380	49.16
	DA-RCOT (2025)	TPAMI	19.08	0.6372	0.3910	0.2845	0.4321	46.37
	AutoDIR (2024)	ECCV	19.90	0.6643	0.3542	0.2534	0.3986	49.64
	AgenticIR (2025)	ICLR	20.55	0.7009	0.3072	0.3204	0.4648	57.57
	MAIR (2026)	IJCV	20.92	0.7004	0.2788	0.3544	0.5084	60.98
	4KAgent (2025)	NeurIPS	20.95	0.6727	0.3017	0.3734	0.5505	62.69
	RAR (2026)	CVPR	21.04	0.7326	**0.1269**	0.4582	**0.6483**	56.91
	RegionAgent (Ours)	Ours	**21.61**	**0.7496**	0.2324	**0.5790**	0.5609	**67.26**
Group C	AirNet (2022)	CVPR	17.95	0.5145	0.5782	0.1854	0.3113	30.12
	PromptIR (2023)	NeurIPS	18.51	0.5166	0.5756	0.1906	0.3104	29.71
	MiOIR (2024)	arXiv	15.63	0.4896	0.5376	0.1717	0.2891	37.95
	DA-CLIP (2024)	ICLR	18.53	0.5320	0.5335	0.1916	0.3476	33.87
	InstructIR (2024)	ECCV	17.09	0.5135	0.5582	0.1732	0.2537	33.69
	AdaIR (2025)	ICLR	17.88	0.5209	0.5802	0.2036	0.3276	30.83
	DA-RCOT (2025)	TPAMI	17.45	0.4993	0.6097	0.1937	0.3172	29.50
	AutoDIR (2024)	ECCV	18.61	0.5443	0.5019	0.2045	0.2939	37.86
	AgenticIR (2025)	ICLR	18.82	0.5474	0.4493	0.2698	0.3948	48.68
	MAIR (2026)	IJCV	19.42	0.5544	0.4142	0.2798	0.4239	51.36
	4KAgent (2025)	NeurIPS	19.77	0.5629	0.4271	0.3545	0.5233	55.56
	RAR (2026)	CVPR	19.33	0.6579	**0.1489**	0.4653	**0.6554**	56.56
	Xiong et al. (2026)	CVPR	19.85	0.5765	0.3392	0.3298	0.4742	58.78
	RegionAgent (Ours)	Ours	**20.90**	**0.6591**	0.2976	**0.5385**	0.5222	**64.29**

fixed, complete-trajectory scores provide supervision for the differing action's effect on subsequent refinement. The trainable and frozen reference policies both start from the step-level checkpoint. Retained pairs receive equal weight. Training and DPO details appear in Appendices C.1 and C.2.

4 Experiments

4.1 Experimental Settings

Evaluation benchmarks. We evaluate mixed degradations on AgenticIR's MiO100-derived benchmark (Zhu et al., 2025; Kong et al., 2024), which applies degradation operators sequentially to each source image, and single degradations on RAR's eight-dataset suite (Chen et al., 2026) (Appendix D.1). Tables 1 and 2 compare RegionAgent with all-in-one models, diffusion-based restorers, and restoration agents.

Evaluation metrics. PSNR, SSIM, and LPIPS measure fidelity to the clean image, while CLIP-IQA, MUSIQ, and MANIQA provide no-reference quality estimates (Wang et al., 2004; Zhang et al., 2018; Wang et al., 2023; Ke et al., 2021; Yang et al., 2022). Higher values are better except for LPIPS.

Implementation details. We train the VLM Qwen3-VL-4B-Instruct (Bai et al., 2025) with SFT followed by step-level and trajectory-level DPO. The framework combines an OSEDiff-based shared

restorer (Wu et al., 2024), four LoRA experts, and a frozen U-Net Localizer (Ronneberger et al., 2015). Inference permits at most $T_{\max} = 8$ `REFINE` actions, stopping earlier on `STOP`. Experiments run on NVIDIA A100 GPUs; Appendices A–C give implementation, supervision, and training details.

4.2 Main Results

Quantitative comparison. RegionAgent achieves the highest PSNR and SSIM across all three mixed-degradation groups, together with the highest MANIQA and MUSIQ scores (Table 1). Its reconstruction advantage persists under three coupled degradations: on Group C (Zhu et al., 2025), it reaches 20.90 dB, exceeding the strongest PSNR baseline from Xiong et al. (2026) by 1.05 dB.

On the single-degradation suite, RegionAgent achieves an average PSNR of 28.71 dB, exceeding the strongest PSNR baseline, AutoDIR, by 0.90 dB, together with the highest average SSIM, MANIQA, and MUSIQ (Table 2).

Table 2: Average performance on single-degradation benchmarks. Best and second-best results are shown in **bold** and underlined, respectively. LD denotes latent diffusion.

Method	PSNR↑	SSIM↑	LPIPS↓	CLIP-IQA↑	MUSIQ↑	MANIQA↑
AirNet	25.65	0.8230	0.1820	0.4516	48.69	0.3163
PromptIR	27.04	0.8445	0.1515	0.4196	48.77	0.3169
AdaIR	26.48	0.7825	0.2604	0.4368	53.51	0.3456
DA-RCOT	26.08	0.7772	0.2579	0.4409	54.34	0.3461
LD	22.17	0.6922	0.1935	0.4253	54.94	0.3438
AutoDIR	27.81	0.8703	0.1283	0.4119	54.37	0.3053
AgenticIR	24.41	0.7794	0.1392	0.3763	55.95	0.2978
RAR	25.88	0.8378	**0.0699**	**0.5566**	56.00	0.4125
RegionAgent	**28.71**	**0.8861**	0.1269	0.5313	**58.35**	**0.6830**

Inference cost. RegionAgent takes 18.60 seconds per image on average and uses 9.28 GB of peak GPU memory. End-to-end latency and memory comparisons are reported in Appendix D.3.

Qualitative comparison. Figure 4 shows that RegionAgent suppresses residual degradations while producing clear local structures. In the landscape, it reduces rain streaks that remain in PromptIR, AdaIR, and RAR. In the building facade, RegionAgent produces straight boundaries and distinct cladding lines, whereas RAR introduces visibly wavy edges. Additional full-image and patch-level comparisons are provided in Appendix D.4.

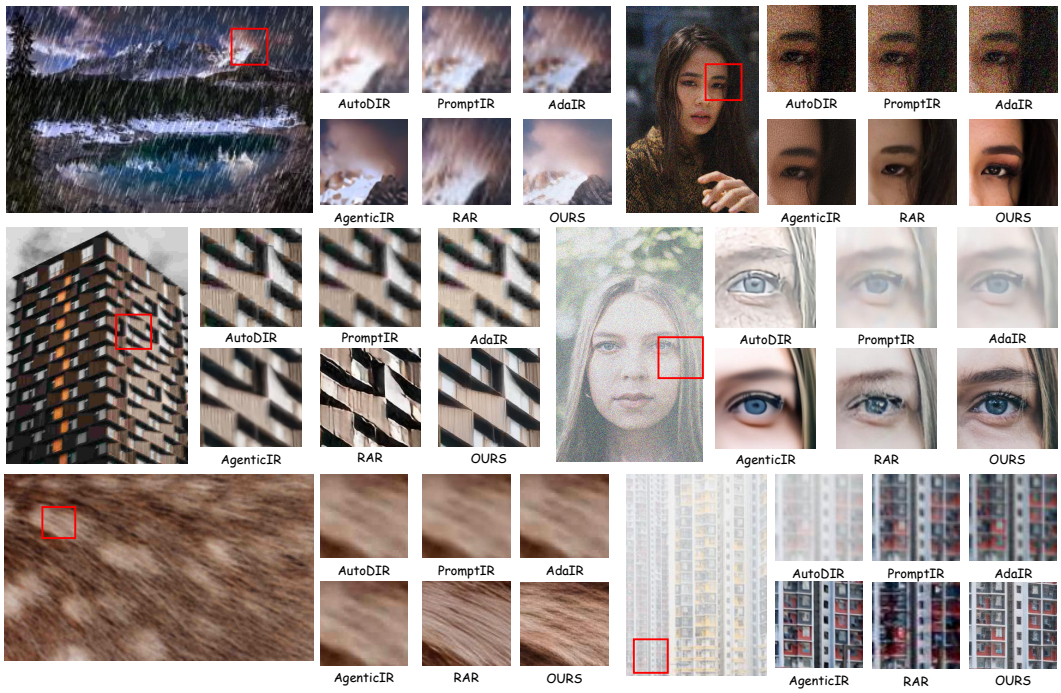

Figure 4: Qualitative comparison on representative degraded images. Red boxes on the LQ inputs identify the enlarged regions shown for each method.

4.3 Ablation Study

Preference optimization. We evaluate the effects of two-stage DPO and supervision scope on restoration quality, keeping the shared restorer, expert checkpoints, Localizer, fusion module, and inference budget fixed. A uses SFT alone. B and C add step-level DPO with whole-action and differing-component supervision, respectively. D applies trajectory-level DPO directly after SFT. E and F start from C and add trajectory-level DPO: E supervises the suffix beginning at the differing action, whereas F supervises that action alone and constitutes the full method.

Table 3: Ablation studies on MiO100. We compare policy training strategies on Group C and expert composition, spatial masks, and Laplacian fusion on Group A.

Variant	Restoration quality				N_{ref}
	PSNR↑	**SSIM↑**	**LPIPS↓**	**MUSIQ↑**	
A	20.01	0.6264	0.3341	59.32	7.79
B	20.25	0.6389	0.3260	60.66	7.63
C	20.47	0.6469	0.3138	62.81	7.61
D	20.37	0.6437	0.3149	62.47	6.48
E	20.69	0.6544	0.3078	63.24	7.34
F	**20.90**	**0.6591**	**0.2976**	**64.29**	7.12

(a) Preference optimization. N_{ref} denotes the mean refinement count.

Variant	Final image		Fusion error	
	PSNR↑	**LPIPS↓**	$E_{\mathrm{keep}}\downarrow$	$E_{\mathrm{bnd}}\downarrow$
Single expert	21.59	0.2537	0.61	4.12
w/o M_{geo}	21.91	0.2433	1.38	4.69
w/o M_{fail}	21.82	0.2450	2.43	4.40
w/o M_{edge}	22.02	0.2432	1.02	6.28
w/o Laplacian pyramid	21.83	0.2445	1.31	4.98
Full method	**22.11**	**0.2377**	0.85	3.28

(b) Local fusion. E_{keep} and E_{bnd} are reported in units of 10^{-3}.

F improves PSNR over SFT alone (A) by 0.89 dB and over the single-stage variants C and D by 0.43 and 0.53 dB, respectively, achieving the best restoration quality among the evaluated variants (Table 3(a)). F also reduces the mean `REFINE` count from C's 7.61 to 7.12.

With the same action pairs and update budget, C exceeds B by 0.22 dB and reduces LPIPS from 0.3260 to 0.3138, supporting supervision focused on the differing configuration component. Starting from the same step-level checkpoint and using the same trajectory pairs and update budget, F exceeds E by 0.21 dB. This favors assigning trajectory preferences to the sole differing action over the complete suffix.

Local fusion. In Table 3(b), the full method improves PSNR by 0.52 dB over single-expert execution and achieves lower LPIPS. To evaluate the masks and multiscale blending, we compare different fusion variants using the same current image, selected region, restored crop, and fusion strength. Removing M_{fail} increases the added error in pixels well restored by Base (E_{keep}) from 0.85 to 2.43. Without M_{edge}, boundary gradient error (E_{bnd}) rises from 3.28 to 6.28; it also increases to 4.69 without M_{geo} and 4.98 without the Laplacian pyramid. The evaluation protocol is detailed in Appendix D.2.

5 Conclusion

We present RegionAgent, an adaptive restoration framework that uses a VLM to generate region-specific restoration actions for images with non-uniform coupled degradations. To address the lack of preferences over regional restoration decisions in conventional image-pair datasets, we construct a dataset of action and trajectory preference pairs. Using these pairs, step-level DPO first trains the VLM to select configurations suited to current regional needs. Trajectory-level DPO then uses complete-trajectory feedback to train the VLM to account for the effects of current actions on subsequent restoration. Experiments across multiple datasets demonstrate strong restoration performance under both mixed and single degradations, while ablation studies validate the effectiveness of two-stage preference learning.

AI use statement

We used OpenAI Codex to assist with manuscript drafting, translation, language editing, literature search, and LaTeX preparation. It provided feedback on methodological formulations, experimental design and parameter settings, and the interpretation of results. We also used AI tools to discuss

the design and assist in preparing the Localizer architecture illustration. Author feedback guided these revisions, and the drafted technical passages and schematic labels were checked against the supplied code and project documentation. VLM-based restoration decisions and preference-pair screening are described in Section 3 and Appendix B.4. The authors take responsibility for the final manuscript, including its text, figures, methodological specifications, and experimental claims.

Reproducibility statement

Appendix A provides implementation and inference details for the restoration models, Localizer, action interface, and regional fusion procedure. Appendix B describes the data sources, partitioning, and preference construction, while Appendix C specifies the training configurations, objectives, and supervision masks. Evaluation protocols, baseline provenance, ablation settings, and efficiency measurements are detailed in Appendix D.

References

Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1692–1700, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Abdelhamed_A_High-Quality_Denoising_CVPR_2018_paper.html.

Abdullah Abuolaim and Michael S. Brown. Defocus deblurring using dual-pixel data. In *European Conference on Computer Vision*, pp. 111–126. Springer, 2020. URL https://www.ecva.net/papers/eccv_2020/papers_ECCV/html/1003_ECCV_2020_paper.php.

Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 126–135, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017_workshops/w12/html/Agustsson_NTIRE_2017_Challenge_CVPR_2017_paper.html.

Yuang Ai, Huaibo Huang, Xiaoqiang Zhou, Jiexiang Wang, and Ran He. Multimodal prompt perceiver: Empower adaptiveness generalizability and fidelity for all-in-one image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25432–25444, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Ai_Multimodal_Prompt_Perceiver_Empower_Adaptiveness_Generalizability_and_Fidelity_for_All-in-One_CVPR_2024_paper.html.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report, 2025. URL https://arxiv.org/abs/2511.21631.

Peter J. Burt and Edward H. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2(4):217–236, 1983. doi: 10.1145/245.247. URL https://doi.org/10.1145/245.247.

Miaomiao Cai, Simiao Li, Wei Li, Xudong Huang, Hanting Chen, Jie Hu, and Yunhe Wang. DSPO: Direct semantic preference optimization for real-world image super-resolution. *arXiv preprint arXiv:2504.15176*, 2025. doi: 10.48550/arXiv.2504.15176. URL https://arxiv.org/abs/2504.15176.

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. In *Advances in Neural Information Processing Systems*, volume 37, pp. 110643–110666, 2024. doi: 10.52202/079017-3512. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/c78f639424b8d89ceb4f2efbb4dfe4f4-Paper-Conference.pdf.

I-Hsiang Chen, Isma Hadji, Enrique Sanchez, Adrian Bulat, Sy-Yen Kuo, Radu Timofte, Georgios Tzimiropoulos, and Brais Martinez. Restore, assess, repeat: A unified framework for iterative image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15506–15515, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Chen_Restore_Assess_Repeat_A_Unified_Framework_for_Iterative_Image_Restoration_CVPR_2026_paper.html.

Wei-Ting Chen, Hao-Yu Fang, Cheng-Lin Hsieh, Cheng-Che Tsai, I-Hsiang Chen, Jian-Jiun Ding, and Sy-Yen Kuo. ALL snow removed: Single image desnowing algorithm using hierarchical dual-tree complex wavelet representation and contradict channel loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. doi: 10.1109/ICCV48922.2021.00416. URL https://openaccess.thecvf.com/content/ICCV2021/html/Chen_ALL_Snow_Removed_Single_Image_Desnowing_Algorithm_Using_Hierarchical_Dual-Tree_ICCV_2021_paper.html.

Seungho Choi and Jihyong Oh. DiTTo: Scalable order-aware all-in-one image restoration agent. *arXiv preprint arXiv:2605.30915*, 2026. doi: 10.48550/arXiv.2605.30915. URL https://arxiv.org/abs/2605.30915.

Fenia Christopoulou, Ronald Cardenas, Gerasimos Lampouras, Haitham Bou Ammar, and Jun Wang. SparsePO: Controlling preference alignment of LLMs via sparse token masks. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 25477–25503. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.1389. URL https://aclanthology.org/2025.findings-emnlp.1389/.

Marcos V. Conde, Gregor Geigle, and Radu Timofte. InstructIR: High-quality image restoration following human instructions. In *Computer Vision – ECCV 2024*, pp. 1–21. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-72764-1_1. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/papers/05238.pdf.

Yuning Cui, Syed Waqas Zamir, Salman Khan, Alois Knoll, Mubarak Shah, and Fahad Shahbaz Khan. AdaIR: Adaptive all-in-one image restoration via frequency mining and modulation. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/fb44a668c2d4bc984e9d6ca261262cbb-Abstract-Conference.html.

Eastman Kodak Company. Kodak lossless true color image suite (PhotoCD PCD0992). Online image suite, 1993. URL https://r0k.us/graphics/kodak/.

Heyang Gao, Zexu Sun, Erxue Min, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Xu Chen. Solving the granularity mismatch: Hierarchical preference learning for long-horizon LLM agents. In *International Conference on Learning Representations*, pp. 67081–67110, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/6dcf277ea32ce3288914faf369fe6de0-Paper-Conference.pdf.

Mohammad Saqib Hasan, Saikat Chakraborty, Santu Karmaker, and Niranjan Balasubramanian. Teaching an old LLM secure coding: Localized preference optimization on distilled preferences. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 26039–26057. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1263. URL https://aclanthology.org/2025.acl-long.1263/.

Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, 2013. doi: 10.1109/TPAMI.2012.213. URL https://doi.org/10.1109/TPAMI.2012.213.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Kui Jiang, Zhongyuan Wang, Peng Yi, Chen Chen, Baojin Huang, Yimin Luo, Jiayi Ma, and Junjun Jiang. Multi-scale progressive fusion network for single image deraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8346–8355, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Jiang_Multi-Scale_Progressive_Fusion_Network_for_Single_Image_Deraining_CVPR_2020_paper.html.

Xu Jiang, Gehui Li, Bin Chen, and Jian Zhang. Multi-agent image restoration. *International Journal of Computer Vision*, 134(5):205, April 2026. doi: 10.1007/s11263-026-02792-5. URL https://doi.org/10.1007/s11263-026-02792-5.

Yitong Jiang, Zhaoyang Zhang, Tianfan Xue, and Jinwei Gu. AutoDIR: Automatic all-in-one image restoration with latent diffusion. In *Computer Vision – ECCV 2024*, pp. 340–359. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-73661-2_19. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/html/5684_ECCV_2024_paper.php.

Yeying Jin, Xin Li, Jiadong Wang, Yan Zhang, and Malu Zhang. Raindrop clarity: A dual-focused dataset for day and night raindrop removal. In *European Conference on Computer Vision*, pp. 1–17. Springer, 2024. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/papers/00841.pdf.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019. doi: 10.1109/CVPR.2019.00453. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Karras_A_Style-Based_Generator_Architecture_for_Generative_Adversarial_Networks_CVPR_2019_paper.html.

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. doi: 10.1109/ICCV48922.2021.00510. URL https://openaccess.thecvf.com/content/ICCV2021/html/Ke_MUSIQ_Multi-Scale_Image_Quality_Transformer_ICCV_2021_paper.html.

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. doi: 10.48550/arXiv.2401.03379. URL https://arxiv.org/abs/2401.03379.

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, 2026a. URL https://openaccess.thecvf.com/content/CVPR2026/html/Li_Hybrid_Agents_for_Image_Restoration_CVPR_2026_paper.html.

Bingchen Li, Xin Li, Jiaqi Xu, Jiaming Guo, Wenbo Li, Renjing Pei, and Zhibo Chen. Test-time preference optimization for image restoration. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(8):5973–5981, 2026b. doi: 10.1609/aaai.v40i8.37520. URL https://ojs.aaai.org/index.php/AAAI/article/view/37520.

Boyi Li, Wenqi Ren, Dengpan Fu, Dacheng Tao, Dan Feng, Wenjun Zeng, and Zhangyang Wang. Benchmarking single-image dehazing and beyond. *IEEE Transactions on Image Processing*, 28(1):492–505, 2019. doi: 10.1109/TIP.2018.2867951. URL https://doi.org/10.1109/TIP.2018.2867951.

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. doi: 10.1109/CVPR52688.2022.01693. URL https://openaccess.thecvf.com/content/CVPR2022/html/Li_

All-in-One_Image_Restoration_for_Unknown_Corruption_CVPR_2022_paper.html.

Mukai Li, Qingcheng Zeng, Tianqing Fang, Zhenwen Liang, Linfeng Song, Qi Liu, Haitao Mi, and Dong Yu. Verified critical step optimization for LLM agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 39627–39639. Association for Computational Linguistics, 2026c. doi: 10.18653/v1/2026.findings-acl.1974. URL https://aclanthology.org/2026.findings-acl.1974/.

Yawei Li, Kai Zhang, Jingyun Liang, Jiezhang Cao, Ce Liu, Rui Gong, Yulun Zhang, Hao Tang, Yun Liu, Denis Demandolx, Rakesh Ranjan, Radu Timofte, and Luc Van Gool. LSDIR: A large scale dataset for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1775–1787, 2023. URL https://openaccess.thecvf.com/content/CVPR2023W/NTIRE/html/Li_LSDIR_A_Large_Scale_Dataset_for_Image_Restoration_CVPRW_2023_paper.html.

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. doi: 10.1109/CVPR52734.2025.02084. URL https://openaccess.thecvf.com/content/CVPR2025/html/Lin_JarvisIR_Elevating_Autonomous_Driving_Perception_with_Intelligent_Image_Restoration_CVPR_2025_paper.html.

Jingren Liu, Shuning Xu, Qirui Yang, Yun Wang, Xiangyu Chen, and Zhong Ji. FAPE-IR: Frequency-aware planning and execution framework for all-in-one image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15561–15573, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Liu_FAPE-IR_Frequency-Aware_Planning_and_Execution_Framework_for_All-in-One_Image_Restoration_CVPR_2026_paper.html.

Shuaizheng Liu, Jianqi Ma, Lingchen Sun, Xiangtao Kong, and Lei Zhang. InstructRestore: Region-customized image restoration with human instructions. In *Advances in Neural Information Processing Systems*, volume 38, pp. 44948–44968, 2025. doi: 10.52202/085713-1497. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/3fb97beb678a1c46bf25f40b3a8c6f98-Paper-Conference.pdf.

Ziwei Luo, Fredrik K. Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B. Schön. Controlling vision-language models for multi-task image restoration. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/467e109270678b3e2b3c2e7fa6b3d575-Abstract-Conference.html.

Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3883–3891, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Nah_Deep_Multi-Scale_Convolutional_CVPR_2017_paper.html.

Vaishnav Potlapalli, Syed Waqas Zamir, Salman Khan, and Fahad Shahbaz Khan. PromptIR: Prompting for all-in-one image restoration. In *Advances in Neural Information Processing Systems*, volume 36, pp. 71275–71293, 2023. doi: 10.52202/075280-3121. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/e187897ed7780a579a0d76fd4a35d107-Paper-Conference.pdf.

Rui Qian, Robby T. Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2482–2491, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Qian_Attentive_Generative_Adversarial_CVPR_2018_paper.html.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741, 2023. doi: 10.52202/075280-2338. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a85b405ed65c6477a4fe8302b5e06ce7-Paper-Conference.pdf.

Chao Ren, Xiaohai He, Chuncheng Wang, and Zhibo Zhao. Adaptive consistency prior based deep network for image denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8596–8606, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Ren_Adaptive_Consistency_Prior_Based_Deep_Network_for_Image_Denoising_CVPR_2021_paper.html.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, June 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper.html.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, volume 9351 of *Lecture Notes in Computer Science*, pp. 234–241. Springer, 2015. doi: 10.1007/978-3-319-24574-4_28. URL https://doi.org/10.1007/978-3-319-24574-4_28.

Xiaole Tang, Xiang Gu, Xiaoyi He, Xin Hu, and Jian Sun. Degradation-aware residual-conditioned optimal transport for unified image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(8):6764–6779, 2025. doi: 10.1109/TPAMI.2025.3562211. URL https://doi.org/10.1109/TPAMI.2025.3562211.

Jianyi Wang, Kelvin C. K. Chan, and Chen Change Loy. Exploring CLIP for assessing the look and feel of images. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(2):2555–2563, 2023. doi: 10.1609/aaai.v37i2.25353. URL https://doi.org/10.1609/aaai.v37i2.25353.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861. URL https://doi.org/10.1109/TIP.2003.819861.

Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. In *Proceedings of the British Machine Vision Conference*, pp. 155, 2018. URL https://bmva-archive.org.uk/bmvc/2018/contents/papers/0451.pdf.

Rongyuan Wu, Lingchen Sun, Zhiyuan Ma, and Lei Zhang. One-step effective diffusion network for real-world image super-resolution. In *Advances in Neural Information Processing Systems*, volume 37, pp. 92529–92553, 2024. doi: 10.52202/079017-2938. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/a8223b0ad64007423ffb308b0dd92298-Abstract-Conference.html.

Honglin Xiong, Chenjie Zhu, Jianbiao Ding, Zixuan Ni, Wei Li, Zhenpeng Mi, and Qian Wang. Beyond sequential tools: A unified VLM agent system for photographic post-processing via dynamic multi-expert fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41521–41530, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Xiong_Beyond_Sequential_Tools_A_Unified_VLM_Agent_System_for_Photographic_CVPR_2026_paper.html.

Qingsen Yan, Yixu Feng, Cheng Zhang, Guansong Pang, Kangbiao Shi, Peng Wu, Wei Dong, Jinqiu Sun, and Yanning Zhang. HVI: A new color space for low-light image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5678–5687, 2025. doi: 10.1109/CVPR52734.2025.00533. URL https:

//openaccess.thecvf.com/content/CVPR2025/html/Yan_HVI_A_New_Color_Space_for_Low-light_Image_Enhancement_CVPR_2025_paper.html.

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 Workshops*, pp. 1191–1200, 2022. URL https://openaccess.thecvf.com/content/CVPR2022W/NTIRE/html/Yang_MANIQA_Multi-Dimension_Attention_Network_for_No-Reference_Image_Quality_Assessment_CVPRW_2022_paper.html.

Wenhan Yang, Robby T. Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep joint rain detection and removal from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1357–1366, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Yang_Deep_Joint_Rain_CVPR_2017_paper.html.

Wenhan Yang, Shiqi Wang, Yuming Fang, Yue Wang, and Jiaying Liu. From fidelity to perceptual quality: A semi-supervised approach for low-light image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. doi: 10.1109/CVPR42600.2020.00313. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Yang_From_Fidelity_to_Perceptual_Quality_A_Semi-Supervised_Approach_for_Low-Light_CVPR_2020_paper.html.

Ke Yu, Chao Dong, Liang Lin, and Chen Change Loy. Crafting a toolchain for image restoration by deep reinforcement learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2443–2452, 2018. doi: 10.1109/CVPR.2018.00259. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Yu_Crafting_a_Toolchain_CVPR_2018_paper.html.

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14821–14831, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Zamir_Multi-Stage_Progressive_Image_Restoration_CVPR_2021_paper.html.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_The_Unreasonable_Effectiveness_CVPR_2018_paper.html.

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. doi: 10.48550/arXiv.2605.22104. URL https://arxiv.org/abs/2605.22104.

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*, pp. 57985–58013, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/921ac785fa9edc73cacaf2664f43d234-Paper-Conference.pdf.

Yushen Zuo, Qi Zheng, Mingyang Wu, Xinrui Jiang, Renjie Li, Jian Wang, Yide Zhang, Gengchen Mai, Lihong Wang, James Zou, Xiaoyu Wang, Ming-Hsuan Yang, and Zhengzhong Tu. 4KAgent: Agentic any image to 4K super-resolution. In *Advances in Neural Information Processing Systems*, volume 38, pp. 164106–164175, 2025. doi: 10.52202/085713-5476. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/f0075fe4e59652cf43148dcfab8d3c93-Abstract-Conference.html.

APPENDIX CONTENTS

A Implementation Details **17**
A.1 Models and LoRA Experts . . . 17
A.2 Localizer and Region Proposals . . . 18
A.3 Regional Sampling and Fusion . . . 19
A.4 Policy Interface and Inference . . . 21

B Supervision Data Construction **21**
B.1 Data Sources and Splits . . . 21
B.2 Action Sampling and Trajectory Search . . . 22
B.3 Outcome Scoring . . . 23
B.4 Preference Pairing, Selection, and Coverage . . . 24

C Policy Training Details **25**
C.1 Supervised Fine-Tuning and Training Setup . . . 25
C.2 Two-Stage Preference Optimization . . . 26

D Evaluation Details and Additional Results **27**
D.1 Benchmarks, Metrics, and Baseline Protocols . . . 27
D.2 Ablation Settings and Diagnostic Metrics . . . 28
D.3 Inference Efficiency and Cost Measurement . . . 29
D.4 Qualitative Comparisons . . . 30
D.4.1 Full-Image Comparisons . . . 30
D.4.2 Patch-Level Comparisons . . . 32

A Implementation Details

A.1 Models and LoRA Experts

Shared restorer. We build the shared restorer on OSEDiff (Wu et al., 2024) with Stable Diffusion 2.1 components. The model encodes an input image into VAE latents, performs one denoising update at diffusion timestep 999, and decodes the restored image through a VAE with encoder-to-decoder skip connections. Starting from pretrained OSEDiff weights, we fine-tune the model on paired degraded and clean images from the MiO training set (Kong et al., 2024), whose source images are disjoint from the MiO100 evaluation sources. Fine-tuning updates rank-4 LoRA adapters in the UNet and VAE encoder, the UNet input convolution, and the VAE skip-connection parameters. The training objective combines mean squared reconstruction error, LPIPS, and OSEDiff's variational score-distillation loss with weights 1, 2, and 1, respectively. The resulting shared parameters θ_0 provide both Base initialization and the common backbone for regional experts.

Expert construction. Each of the four restoration experts adds an independent LoRA adapter to the same shared checkpoint. We attach these adapters to the UNet attention key and value projections, using rank 16, LoRA scaling parameter 32, and dropout 0.05. During expert training, the shared UNet parameters and adapters, VAE, and text encoder remain frozen; optimization updates the newly added expert adapter. Table 4 lists the restoration tasks and training sources used to specialize each expert.

Table 4: Training tasks and data sources for the four LoRA experts.

Expert	Restoration tasks	Training sources
Structure/detail	Motion and defocus deblurring; detail restoration	LSDIR (Li et al., 2023), DIV2K (Agustsson & Timofte, 2017), GoPro (Nah et al., 2017), DPDD (Abuolaim & Brown, 2020), FFHQ (Karras et al., 2019)
Noise/compression	Denoising and JPEG artifact removal	SIDD (Abdelhamed et al., 2018), LSDIR, DIV2K, FFHQ
Visibility/ illumination	Low-light enhancement and dehazing	LOL-v2 Real, LOL-v2 Synthetic (Yang et al., 2020), RESIDE-6K (Li et al., 2019), FFHQ
Structured weather	Rain-streak, snow, and raindrop removal	Rain13K (Jiang et al., 2020), CSD (Chen et al., 2021), RaindropClarity (Jin et al., 2024), LSDIR, DIV2K

Expert training. Experts learn from degraded inputs paired with their corresponding clean images. We use aligned 512×512 random crops, apply the same horizontal flip to both images with probability 0.5, and normalize RGB values to $[-1, 1]$. For detail-restoration samples, DIV2K supplies bicubic low-resolution inputs at factors 2 and 4, while LSDIR inputs are synthesized by bicubic downsampling at the same factors. These inputs are upsampled to the clean-image dimensions before restoration, keeping the expert's input and output at the same resolution.

For a restored prediction $\widehat{I}$ and clean target G, the expert objective is $\mathcal{L}_{\text{expert}} = \mathcal{L}_1(\widehat{I}, G) + \mathcal{L}_{\text{LPIPS}}(\widehat{I}, G) + 0.1\mathcal{L}_{\text{grad}}(\widehat{I}, G)$. Here, $\mathcal{L}_1$ is mean absolute reconstruction error, $\mathcal{L}_{\text{LPIPS}}$ uses a frozen VGG-based perceptual network (Zhang et al., 2018), and $\mathcal{L}_{\text{grad}}$ sums the mean absolute discrepancies of horizontal and vertical finite differences. These terms supervise pixel reconstruction, perceptual similarity, and local structure, respectively.

We train each expert with AdamW for up to 30K updates, warming up the learning rate to 5×10^{-5} over the first 200 updates and keeping it constant thereafter. Training uses four images per GPU, no gradient accumulation, bfloat16 precision, weight decay 0.01, Adam coefficients $(0.9, 0.999)$, optimizer epsilon 10^{-8}, and gradient-norm clipping at 1.0.

The restorer's text conditioning follows the OSEDiff prompt pipeline: a frozen image tagger extracts prompts from clean targets during training, while validation uses the pretrained degradation-aware prompt extractor on degraded inputs.

Regional execution. During VLM optimization and inference, the shared restorer and expert adapters remain fixed, and the VLM selects the expert coefficients. These coefficients combine the expert updates on the common backbone as defined in Section 3.3. The composed model pro-

cesses the current-image crop C_t, and DSRF integrates its residual correction. The expert ordering and discrete coefficient values are specified in Section A.4.

A.2 Localizer and Region Proposals

Training states and inputs. The Localizer learns to identify restoration errors across untreated degraded inputs, Base outputs, and intermediate restoration results. Each sample pairs the image being assessed with its corresponding clean image. We assemble the 15-channel input X_t defined in Section 3.2 from LQ, Base, Current, the absolute difference between Current and LQ, and the absolute difference between Current and Base. These inputs expose both the current image content and its changes relative to the original observation and initial restoration. Figure 5 shows the Localizer's encoder–decoder architecture and convolutional building block. All states derived from the same source image belong to the same data partition, as described in Appendix B.1.

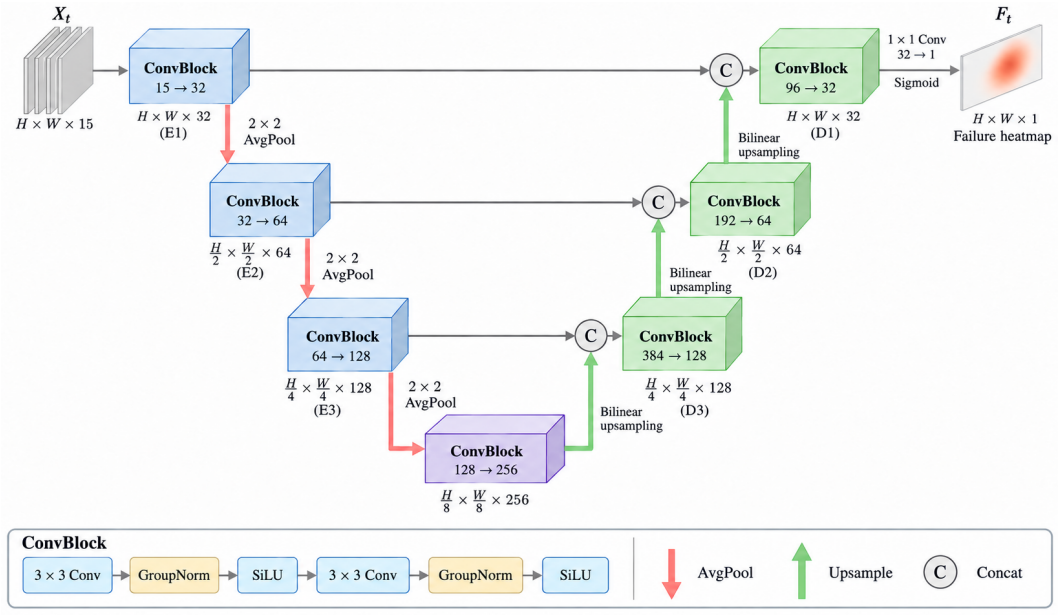

Figure 5: The Localizer maps the 15-channel input X_t to the failure heatmap F_t through an encoder–decoder with skip connections. The bottom inset details the ConvBlock structure.

Network architecture. We train a single U-Net with four encoder levels of widths 32, 64, 128, and 256. Each level contains two 3×3 convolutions, each followed by eight-group normalization and a SiLU activation. Three 2×2 average-pooling operations reduce the spatial resolution between encoder levels. At each decoder level, we bilinearly upsample the decoder features to match the spatial resolution of the corresponding encoder features and concatenate the two along the channel dimension. The concatenated features pass through the same two-convolution block, with output widths of 128, 64, and 32 at successive decoder levels. A 1×1 convolution followed by a sigmoid produces the single-channel failure heatmap F_t at the input image resolution.

Dense target construction. For each training state, we derive a continuous target map $T_t \in [0,1]^{H \times W}$ from discrepancies between the current image I_t and its corresponding clean image G. We combine four spatial error maps: the mean absolute RGB difference $e_{\mathrm{pix},t}$, a frozen spatial LPIPS-Alex map $e_{\mathrm{perc},t}$ (Zhang et al., 2018), the discrepancy in horizontal and vertical image gradients $e_{\mathrm{grad},t}$, and the local structural error $e_{\mathrm{struc},t}$. The LPIPS map is bilinearly resized to the image resolution. Gradient error sums the absolute differences of the horizontal and vertical finite differences, averaged over RGB channels. Structural error is $(1 - \mathrm{SSIM}_{11\times11}(I_t, G))/2$ using an 11×11 local window.

To place the four components on comparable scales, we estimate a fixed 95th-percentile scale $\kappa_j > 0$ for each component j from calibration image states, sampling 4,096 pixels per image and component. We normalize and clip each map before taking their equally weighted mean and smoothing it:

$$T_t = \mathrm{Avg}_{9\times 9}\left[\frac{1}{4}\sum_{j\in\mathcal{J}} \mathrm{clip}\left(\frac{e_{j,t}}{\kappa_j}, 0, 1\right)\right].$$

Here, $\mathcal{J} = \{\mathrm{pix}, \mathrm{perc}, \mathrm{grad}, \mathrm{struc}\}$ indexes the four error components, and $\mathrm{Avg}_{9\times 9}$ denotes spatial averaging with a 9×9 window. The calibration scales remain fixed across images and restoration steps, giving error severity a consistent reference as restoration progresses. Larger values of T_t indicate greater remaining restoration error.

Training objective and optimization. The associated binary target is $Y_t(p) = \mathbf{1}[T_t(p) \geq 0.5]$, where p indexes image pixels. We train the Localizer with the following loss:

$$\mathcal{L}_{\mathrm{loc}} = \mathcal{L}_{\mathrm{SmoothL1}}(F_t, T_t) + 0.5\mathcal{L}_{\mathrm{BCE}}(F_t, Y_t) + 0.5\mathcal{L}_{\mathrm{Dice}}(F_t, Y_t).$$

Smooth-ℓ_1 regression preserves the continuous severity signal, while binary cross-entropy and Dice supervision emphasize the location and spatial extent of high-error content. For one image, the Dice loss is $1 - (2\sum_p F_t(p)Y_t(p) + \epsilon)/(\sum_p F_t(p) + \sum_p Y_t(p) + \epsilon)$, with $\epsilon = 10^{-6}$. We average losses over the training batch.

We optimize the Localizer with AdamW for 3K updates, using 32 samples per GPU without gradient accumulation. The learning rate warms up over 200 updates to 3×10^{-4} and then follows a cosine schedule. Training uses weight decay 0.01, bfloat16 precision, gradient-norm clipping at 1.0, and random seed 2027. After training, we freeze the Localizer and use X_t to predict heatmaps for preference construction and inference.

Candidate region generation. We define window sizes relative to the image short side to maintain comparable regional coverage across resolutions. Let $s = \min(H, W) \geq 86$ pixels and $\rho \in \{0.15, 0.30, 0.45\}$ denote a relative scale. Its square-window side is $\ell_\rho = \mathrm{clip}(\mathrm{round}(\rho s), 64, \lfloor 0.75s \rfloor)$ pixels, and its stride is $d_\rho = \max(8, \mathrm{round}(\ell_\rho/4))$. We discard windows with area greater than $0.35HW$ and anchor the final window in each row and column to the image boundary for full coverage.

At each iteration, we score each window r by $q_t(r) = |r|^{-1}\sum_{p\in r} F_t(p)$, where $|r|$ is its number of pixels. We rank windows by this mean failure score and apply non-maximum suppression with an intersection-over-union threshold of 0.4, retaining the top $K = 4$ proposals. Ties follow a fixed order of window size, vertical coordinate, and horizontal coordinate. The proposer ranks windows without applying an absolute heatmap threshold; the 0.5 threshold above is used to form binary training targets. At inference, we add the previously selected region r_{t-1} to these new proposals and include duplicate boxes once, giving at most five candidates after the first refinement. The retained region keeps its image coordinates, but its crop is extracted from the updated current image. The VLM can therefore revisit the same region or select a newly proposed one.

A.3 Regional Sampling and Fusion

Fixed-grid regional sampling. The candidate regions vary in image-space size, while the shared restorer processes a fixed-resolution crop. We map the selected region to a central core of side $s_c = 200$ within a crop of side $s_p = 304$, leaving a halo of width $s_h = (s_p - s_c)/2 = 52$ on each side. These dimensions refer to the expert input grid. The surrounding halo supplies context for restoration and a transition area for fusion.

Let the selected region have half-open image coordinates $r_t = [x_1, x_2) \times [y_1, y_2)$, width $w_r = x_2 - x_1$, and height $h_r = y_2 - y_1$. With image pixel centers at half-integer coordinates, the image-to-crop transform is

$$B_t(x, y) = \left(s_c\frac{x - x_1}{w_r} + s_h,\ s_c\frac{y - y_1}{h_r} + s_h\right).$$

The core occupies $[52, 252) \times [52, 252)$ in crop coordinates. The full crop covers an image-space context box obtained by extending the selected region by $s_h w_r/s_c$ horizontally and $s_h h_r/s_c$ vertically on each side, giving a context-to-core size ratio of $s_p/s_c = 1.52$. For crop indices

$u, v \in \{0, \ldots, 303\}$, we sample both I_t and F_t at $B_t^{-1}(u + \frac{1}{2}, v + \frac{1}{2})$ to obtain C_t and $\widetilde{F}_t$. Sampling uses bilinear interpolation with `align_corners=false` and reflection padding beyond the image boundary. The shared transform keeps the image content and predicted restoration needs aligned.

Spatial fusion masks. The configured restorer produces E_t from C_t as described in Section 3.3. DSRF controls its residual $\Delta_t = E_t - C_t$ using masks computed on the same 304×304 grid. Let Ω_{core} denote the core pixels and Ω_{halo} the remaining crop pixels. For a crop pixel p, let $d(p)$ be its Euclidean distance to the nearest core pixel, with $d(p) = 0$ inside the core. The geometric mask is

$$M_{\text{geo}}(p) = \frac{1}{2}\left[1 + \cos\left(\pi \min\left\{\frac{d(p)}{s_h}, 1\right\}\right)\right].$$

It retains unit weight throughout the core and smoothly reduces the correction toward the outer crop boundary.

The failure mask adapts this geometric support to the predicted error distribution. We threshold $\widetilde{F}_t$ at its 45th percentile within the crop, keeping values at or above the threshold. Morphological closing with a 5×5 square kernel, followed by opening with a 3×3 square kernel, yields the binary support Z_t. Closing fills small gaps and opening removes isolated responses. We multiply Z_t by the original continuous heatmap and use the grayscale current crop, scaled to $[0, 1]$, as guidance for filtering:

$$M_{\text{fail}} = \text{clip}\Big(\text{GF}\big(\text{gray}(C_t), Z_t \odot \widetilde{F}_t\big), 0, 1\Big).$$

Here, GF is the guided filter (He et al., 2013) with radius 9, a 19×19 local window, and regularization 10^{-3}. Continuous heatmap values encode restoration need, while guided filtering aligns mask transitions with current-image structure.

The edge mask regulates structural transitions in the halo. We convert C_t and E_t to grayscale in $[0, 1]$ and compute gradient magnitudes g_C and g_E using 3×3 Sobel filters. Their absolute difference is $d_g(p) = |g_E(p) - g_C(p)|$. We set σ_g to the 75th percentile of d_g over the entire crop, with a minimum of 10^{-3}, and define

$$M_{\text{edge}}(p) = \begin{cases} 1, & p \in \Omega_{\text{core}}, \\ \exp(-d_g(p)/\sigma_g), & p \in \Omega_{\text{halo}}. \end{cases}$$

The core remains available for structural correction, while halo locations with larger gradient-magnitude changes receive less weight. The action's fusion strength scales the combined gate as $M_t^\alpha = \alpha_t(M_{\text{geo}} \odot M_{\text{fail}} \odot M_{\text{edge}})$. All three masks lie in $[0, 1]$.

Multiscale residual fusion. We decompose the correction into four pyramid levels ($J = 4$), with spatial sizes 304, 152, 76, and 38 per side. Let $R_\ell = G_\ell(\Delta_t)$ and $A_\ell = G_\ell(M_t^\alpha)$ denote the Gaussian pyramids of the residual and gate. Starting from $R_0 = \Delta_t$ and $A_0 = M_t^\alpha$, each reduction applies the separable filter $(1, 4, 6, 4, 1)/16$ along both axes and subsamples by two. The corresponding expansion operator Up inserts zeros and applies the same two-dimensional filter scaled by four, restoring the preceding level's size. We use reflected boundary handling for these pyramid operations.

The Laplacian residual bands are $L_\ell(\Delta_t) = R_\ell - \text{Up}(R_{\ell+1})$ for $\ell = 0, 1, 2$, with $L_3(\Delta_t) = R_3$. We apply the gate at each scale and reconstruct from coarse to fine:

$$\widehat{R}_3 = A_3 \odot R_3,$$
$$\widehat{R}_\ell = \text{Up}(\widehat{R}_{\ell+1}) + A_\ell \odot L_\ell(\Delta_t), \qquad \ell = 2, 1, 0.$$

The reconstructed residual is $\widehat{\Delta}_t = \widehat{R}_0$. The scalar masks are broadcast across RGB channels. Coarser levels blend low-frequency appearance over wider transitions, while finer levels control local detail.

Inverse mapping and writeback. We map $\widehat{\Delta}_t$ back to image coordinates by bilinear interpolation with the same pixel-center convention. For an image pixel at column x and row y, we query the crop residual at $B_t(x + \frac{1}{2}, y + \frac{1}{2})$. The discrete context support uses the floor of the continuous box's lower bounds and the ceiling of its upper bounds. We intersect this support with $[0, W) \times [0, H)$, discard samples corresponding to reflected padding, and add the valid residual to I_t. This implements $\mathcal{W}_{B_t}^{-1}$ in Section 3.3: its output is zero outside the valid context support, so those pixels retain their current values. The resulting image becomes I_{t+1} for the next localization and action-selection step.

A.4 Policy Interface and Inference

Observations and prompt content. The VLM receives the original low-quality image, the immutable Base image, and the current image, together with numbered candidate overlays and crops from the current image. The candidate catalog associates each identifier with an image-space box. The text input supplies this catalog, the expert ordering, legal action values, the output action format, and the preceding structured actions. These inputs form the observation o_t defined in Section 3.2. The policy selects a region by its identifier in the current catalog. Candidate geometry and the rule for retaining the previously selected region are specified in Appendix A.2.

The same observation interface is used for training and inference. Clean reference images and offline preference scores supply supervision during data construction; they are excluded from the policy input. In the framework diagrams, S_t denotes the restoration state represented by I_t, and A_t denotes its structured action a_t.

Action fields and legal values. Each output specifies either `STOP` or one `REFINE` action. A refinement identifies the target region r_t, four expert coefficients $\mathbf{w}_t$, and fusion strength α_t. The coefficients follow the fixed order structure/detail, noise/compression, visibility/illumination, and structured weather. Their legal values and the fusion strengths are

$$\mathbf{w}_t \in \{0, 0.25, 0.5, 0.75, 1\}^4 \setminus \{\mathbf{0}\}, \qquad \alpha_t \in \{0.25, 0.5, 0.75, 1\}.$$

The coefficients configure the LoRA composition described in Appendix A.1, while α_t scales the residual gate in Appendix A.3. A `STOP` action terminates restoration and has no regional configuration fields.

The VLM serializes each action as a JSON object with a lowercase `action` value of `refine` or `stop`. For refinement, `region_id` is an integer identifier in the current catalog. The four-element `expert_weights` array follows the expert order above, and `fusion_strength` stores α_t. For example:

```
{
  "action": "refine",
  "region_id": 2,
  "expert_weights": [0.25, 0.50, 0.00, 0.75],
  "fusion_strength": 0.75
}
```

The stopping response is {`"action": "stop"`}.

Inference loop. Base restoration initializes $I_0 = I_{\text{Base}}$ and an empty action history. At each decision step, we assemble the observation from the current image and its candidate catalog, and the VLM generates one action. For `REFINE`, the executor resolves the selected identifier to its region coordinates, restores the Current crop with the selected expert coefficients, and applies DSRF with the selected fusion strength. We append the action to the history and use the fused image as I_{t+1}. The Localizer then refreshes the candidates for the next decision, allowing the policy to revisit the preceding region or select another one.

Inference returns the current image when the VLM emits `STOP` or after $T_{\max} = 8$ executed `REFINE` actions. Offline supervision samples the same action space using the procedure and search budgets in Appendix B.2.

B Supervision Data Construction

B.1 Data Sources and Splits

Training sources. We construct policy supervision from 8,000 source images, balanced across four degradation families with 2,000 sources each. Table 5 lists their datasets. This pool supplies the image states and executed candidates used to form SFT demonstrations and preference pairs. The expert-training datasets are specified separately in Appendix A.1.

Aligned image pairs. GoPro, SIDD, RESIDE-6K, and RaindropClarity provide official degraded–clean pairs. For clean-image sources, we synthesize degradations from the corresponding family:

Table 5: Source datasets for regional restoration supervision. Counts refer to training source images before action sampling and trajectory generation.

Degradation family	Source datasets	Sources
Structure/detail	LSDIR, DIV2K, GoPro, FFHQ	2,000
Noise/compression	LSDIR, DIV2K, SIDD, FFHQ	2,000
Visibility/illumination	RESIDE-6K, FFHQ	2,000
Structured weather	LSDIR, RaindropClarity, DIV2K	2,000
Total		8,000

resolution loss and blur, noise and JPEG compression, reduced illumination and haze, or rain, snow, and raindrops. Degradation choices and crop coordinates are determined by the source identity and a fixed random seed. We form aligned 512×512 RGB pairs by applying the same crop to the degraded and clean images, proportionally enlarging images whose shorter side is below 512 pixels. For resolution-loss samples, bicubic downsampling by a factor of 2 or 4 is followed by upsampling to the original crop size.

Source separation and initialization. We partition source identities before degradation synthesis, cropping, Base restoration, and rollout. Images from the same scene or burst remain grouped, and all degradation variants, intermediate states, and preference pairs derived from one source retain its partition. Calibration uses 256 additional source-disjoint images, separate from policy and Localizer training, model selection, and evaluation. For each training pair, the shared restorer produces the fixed Base image and initializes Current as described in Appendix A.4. The corresponding clean image is available to offline scoring but is excluded from the policy observation.

B.2 Action Sampling and Trajectory Search

Regional probes. For same-state action preference sampling, the discrete space in Appendix A.4 contains $5^4 - 1 = 624$ nonzero expert-weight vectors and four fusion strengths. Exhaustive evaluation on four regions would therefore require 9,984 `REFINE` candidates per state. We use hierarchical sampling to identify promising regions and expert configurations before allocating further restoration calls. All candidates at a parent state start from the same Current image and use the regional executor in Appendix A.3.

On each of four newly proposed regions, we first evaluate four single-expert anchors $\mathbf{w} = 0.75e_j$, where e_j is the unit vector for expert $j \in \{1, 2, 3, 4\}$. Each restored crop is fused at all four strengths. For region r and expert j, the anchor score is $u_{r,j} = \max_\alpha U_{\text{step}}(I_t, (r, 0.75e_j, \alpha))$, with the maximum taken over the legal fusion strengths. We select the two regions with the highest $\max_j u_{r,j}$. Within each selected region, the anchor scores rank the experts as $e_{(1)}, \ldots, e_{(4)}$. These rankings guide the subsequent configuration expansion. Ties are resolved by region identifier and the fixed expert order.

Configuration expansion. For each selected region, we generate 14 additional weight vectors based on the expert ranking, using the templates in Table 6. The templates vary the strength of individual experts and introduce combinations with two, three, or four active experts. Each vector requires one restored crop, which we reuse across all four fusion strengths. The four-region probe stage requires 16 restoration calls and yields 64 fused candidates. Expanding two regions adds 28 calls and 112 candidates, for a total of 44 calls and 176 `REFINE` candidates per sampled state. A `STOP` option requires no restoration call. Output reuse is confined to a common Current image, region, and expert configuration.

Trajectory search. Within each search trajectory, the region coordinates remain fixed across refinement steps, while expert weights and fusion strengths may vary. Search starts from Base with an empty history. Each parent state contributes at most eight eligible children, and a beam of width four retains the highest-scoring distinct successor states for further expansion. Eligibility and scoring follow Appendices B.4 and B.3. After every retained refinement, we recompute the failure heatmap and update the fusion mask at the fixed region coordinates.

At every parent state, an explicit `STOP` branch adds its current image and accumulated score to the terminal pool without consuming an active beam slot. Paths reaching eight `REFINE` actions

Table 6: Expert-weight templates added on each region selected for expansion. Subscripts in parentheses denote that region's expert ranking from the anchor scores. The four anchors at coefficient 0.75 are already evaluated.

Configuration type	Additional weight vectors	Count
Single expert	$\gamma e_{(j)}, \quad \gamma \in \{0.25, 0.5, 1\},\ j \in \{1, 2\}$	6
Two experts	$0.25e_{(1)} + 0.25e_{(2)}, \quad 0.5e_{(1)} + 0.5e_{(2)},$ $0.75e_{(1)} + 0.25e_{(2)}, \quad 0.25e_{(1)} + 0.75e_{(2)}$	4
Three experts	$0.25 \sum_{j=1}^{3} e_{(j)} + 0.25e_{(k)}, \quad k \in \{1, 2, 3\}$	3
Four experts	$0.25 \sum_{j=1}^{4} e_{(j)}$	1
Total		14

also enter this pool. Search ends when the refinement budget is exhausted or no eligible successor remains. Terminal scores select demonstrations and label trajectory preferences, with the pairing constraints defined in Appendix B.4.

B.3 Outcome Scoring

Quality measurements. Offline scoring combines three full-reference metrics, PSNR, SSIM, and LPIPS, with three no-reference metrics, MANIQA, MUSIQ, and CLIP-IQA. The corresponding clean image supplies the reference for the first group. PSNR and SSIM use RGB images, while LPIPS uses the AlexNet backbone. Full-reference measurements evaluate both the complete image and the selected core region for single-step decisions. No-reference metrics evaluate complete images. All candidates use the same metric implementations and preprocessing.

Fixed normalization scales. We align each metric with improvement through d_m, setting $d_m = -1$ for LPIPS and $d_m = +1$ otherwise. Calibration candidates provide metric-gain distributions from which we estimate fixed 95th-percentile scales. Each source contributes equal total weight, regardless of the number of its candidates. The full-image scale s_m^{global} is the source-balanced 95th percentile of $|d_m[m(I) - m(I_{\text{Base}})]|$ over calibration candidates I. The three full-reference metrics additionally use separate ROI scales s_m^{roi} for regional gains. Scales are frozen before supervision generation and remain common to all states and candidate configurations.

Single-step utility. Let $I_t^{(a)}$ be the image obtained by executing a `REFINE` action a from I_t, and let $m_\Omega(I)$ denote a metric evaluated over domain Ω, using the aligned clean reference when required. Here, $\Omega = \text{global}$ denotes the full image and $\Omega = \text{roi}$ denotes the selected core in original image coordinates. We define the normalized gain

$$g_{m,t}^{\Omega}(a) = \text{clip}\left(\frac{d_m[m_\Omega(I_t^{(a)}) - m_\Omega(I_t)]}{s_m^{\Omega} + \epsilon}, -1, 1 \right),$$

where $\epsilon > 0$ ensures numerical stability. For a full-reference metric, $g_{m,t}(a) = \frac{1}{2} g_{m,t}^{\text{global}}(a) + \frac{1}{2} g_{m,t}^{\text{roi}}(a)$; for a no-reference metric, $g_{m,t}(a) = g_{m,t}^{\text{global}}(a)$. Normalization and clipping precede this combination. The quality gain is $Q_{\text{step}}(I_t, a) = \frac{1}{6} \sum_{m \in \mathcal{M}} g_{m,t}(a)$, where $\mathcal{M}$ contains the six metrics above. The full-image term measures the effect on the restored image, while the ROI term retains sensitivity to corrections within a small selected region.

We obtain the action utility by subtracting the corresponding penalties:

$$\begin{aligned} U_{\text{step}}(I_t, a) = Q_{\text{step}}(I_t, a) - \lambda_{\text{out}} P_t^{\text{out}}(a) - \lambda_{\text{bnd}} P_t^{\text{bnd}}(a) \\ - \lambda_{\text{len}} - \lambda_{\text{rep}} p_t^{\text{rep}}(a) - \lambda_{\text{base}} P_{\text{base}}(I_t^{(a)}). \end{aligned}$$

Each `REFINE` action contributes one unit of action cost and inherits its parent trajectory's history for penalty computation. The indicator $p_t^{\text{rep}}(a)$ records whether its resulting image repeats a preceding state. We use $\lambda_{\text{out}} = 0.5$, $\lambda_{\text{bnd}} = 0.25$, $\lambda_{\text{len}} = 0.002$, $\lambda_{\text{rep}} = 0.1$, and $\lambda_{\text{base}} = 1$.

Halo penalty. Let Ω_t^{halo} denote the halo pixels inside the valid writeback support but outside the selected core, and let G be the corresponding clean image. The halo penalty measures error increases

at these pixels:

$$P_t^{\text{out}}(a) = \frac{1}{3|\Omega_t^{\text{halo}}|} \sum_{p \in \Omega_t^{\text{halo}}} \sum_{c=1}^{3} \Big[|I_t^{(a)}(p,c) - G(p,c)| - |I_t(p,c) - G(p,c)| \Big]_+ ,$$

where p indexes pixels, c indexes RGB channels, and $[z]_+ = \max(z, 0)$. A zero penalty is assigned when the valid halo is empty. This term permits improvements in the halo and penalizes only increases in error. Pixels beyond the writeback support are unchanged by the regional executor.

Boundary penalty. Let M_t^{core} be the binary mask of the selected core in the full image. Its boundary band Ω_t^{bnd} is the support of $\text{dilate}(M_t^{\text{core}}) - \text{erode}(M_t^{\text{core}})$, using a 17×17 square structuring element. The band extends eight pixels inward and outward from the core boundary and is restricted to valid image pixels.

For training scores, we use the same gradient computation as in boundary evaluation. We convert RGB uint8 images to grayscale using OpenCV's `COLOR_RGB2GRAY`, cast them to float32 while retaining the $[0, 255]$ intensity range, and apply 3×3 Sobel filters with `scale=1`. The resulting full-image gradient magnitude is $v(I) = \|\nabla_{\text{Sobel}} \text{gray}(I)\|_2$, and its error relative to the clean image is $e_\nabla(I, G) = |v(I) - v(G)|$. The penalty is

$$P_t^{\text{bnd}}(a) = \frac{1}{|\Omega_t^{\text{bnd}}|} \sum_{p \in \Omega_t^{\text{bnd}}} \Big[e_\nabla(I_t^{(a)}, G)(p) - e_\nabla(I_t, G)(p) \Big]_+ .$$

An empty band receives zero penalty. This term averages positive changes in boundary gradient error relative to the parent image I_t. The coefficient $\lambda_{\text{bnd}} = 0.25$ applies to this unnormalized 8-bit gradient scale.

Repeated states. We quantize each restored image to RGB uint8 and compute a digest of its pixel bytes. Let $h(I)$ denote this digest. For a `REFINE` action at step t, we set $p_t^{\text{rep}}(a) = \mathbf{1}\{h(I_t^{(a)}) \in \{h(I_0), \ldots, h(I_t)\}\}$, where $\mathbf{1}\{\cdot\}$ is the indicator function and $I_0 = I_{\text{Base}}$. The indicator counts one repetition for an output whose digest matches any preceding state. Repeated states are excluded from active search; the weighted repetition term also records their cost when scoring candidate outputs. Deduplication of identical action JSON records is a separate candidate-level operation. A `STOP` action creates no new restored image and adds no repetition penalty.

Final-image Base penalty. The immutable Base image provides a whole-image reference for detecting deterioration after refinement. For a candidate image I, we define its normalized gain over Base on metric m as

$$g_m^B(I) = \text{clip}\left(\frac{d_m[m(I) - m(I_{\text{Base}})]}{s_m^{\text{global}} + \epsilon}, -1, 1 \right).$$

Each candidate sequence receives a single Base penalty evaluated at its endpoint; for a one-action continuation, this endpoint is the successor image. Let $\delta_m \in [0, 1)$ be a fixed metric-specific tolerance for a decrease in normalized gain. We reject a candidate whenever $-g_m^B(I) > \delta_m$ for any metric m. Before filtering, the scoring term measures the average excess deterioration:

$$P_{\text{base}}(I) = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} [-g_m^B(I) - \delta_m]_+.$$

Candidates satisfying every metric tolerance have $P_{\text{base}} = 0$, while candidates exceeding a tolerance receive a positive penalty and are excluded from further expansion.

Complete-trajectory scores. The terminal quality term of the trajectory score $r(\tau_t)$ in Section 3.4 is computed directly from the final image's six equally weighted, normalized full-image gains relative to Base. Its scale s_m is s_m^{global} above. Halo and boundary penalties accumulate over executed actions, the refinement count gives the action cost, and $N_{\text{rep}} = \sum_{i<t} p_i^{\text{rep}}(a_i)$ counts repeated states. The Base penalty applies to the final image. An explicit `STOP` branch retains the image and accumulated penalties at its parent state, allowing immediate termination and continued restoration to be compared with the same trajectory score.

B.4 Preference Pairing, Selection, and Coverage

SFT demonstrations. For each source, we select the highest-scoring completed branch and export its observation–action sequence. To assess the confidence of a target action a at observation

o, we define its branch value as $V(o,a) = \max_{\tau \in \mathcal{T}(o,a)} r(\tau)$, where $\mathcal{T}(o,a)$ contains completed trajectories that pass through o and then execute a. The automatic SFT selection margin between the best and next-best branch values is 0.02. Stopping labels are determined by score comparisons between immediate termination and completed refinement branches. Maximum-depth termination serves solely as a rollout limit.

Action preference pairs. Paired `REFINE` actions share a parent observation and differ in exactly one configuration field: the region identifier, the expert-weight vector, or fusion strength. All other fields remain fixed. Their utility difference $\Delta_{\text{step}} = U_{\text{step}}(I_t, a^+) - U_{\text{step}}(I_t, a^-)$ determines the preferred action a^+, with an automatic-selection margin of 0.03. Region comparisons use the same expert configuration and fusion strength, and each action's ROI score is evaluated on its own selected core. Expert and fusion comparisons hold the region coordinates fixed.

Under a common parent observation, `STOP–REFINE` pairs compare immediate termination with continued restoration using completed trajectory scores. The preferred action is the first decision of the higher-scoring branch. The score compares full restoration outcomes, while the loss selects the `action` value tokens as defined in Appendix C.2.

Trajectory preference pairs. We replace one refinement action in a completed fixed-region search trajectory with a different legal configuration, retaining its target region. From the shared preceding image state, we execute the replacement and the original subsequent actions, recomputing image states, heatmaps, and fusion masks. Both trajectories share Base, the target region, and all action entries except the replacement, while their later image states may differ. After both trajectories terminate, their score difference $\Delta_{\text{traj}} = r(\tau^+) - r(\tau^-)$ determines the preference, with an automatic-selection margin of 0.05. In Figure 3, F^q and $r^q = r(\tau^q)$ denote the final image and score of trajectory τ^q, respectively, for $q \in \{1, 2\}$.

Filtering and coverage. We retain legal, executable actions that match their parent region catalog and have complete restoration outputs and metric evaluations. Active beam successors must satisfy $U_{\text{step}} > 0.02$, produce a state distinct from every preceding state on that branch, and pass the Base safety check described in Appendix B.3. Duplicate actions, outputs, and preference records are removed before selecting supervision. We sample across source datasets, degradation families, trajectory depths, action types, expert configurations, and fusion strengths to distribute supervision across restoration states and decisions.

Pairs are accepted automatically when their score gaps exceed the corresponding thresholds and their metric evidence is consistent. Low-margin or metric-conflicting cases undergo VLM-assisted screening and human verification. Candidate order is randomized and numerical scores and preference labels are hidden during visual review. The review checks residual degradations, local seams, structural artifacts, and consistency with the selected region. The VLM flags cases for inspection, while human verification determines whether the metric-derived label can be retained. The resulting training set contains 10K action preference pairs and 20K trajectory preference pairs, together with the SFT demonstrations.

C Policy Training Details

We train the policy through SFT, step-level DPO, and trajectory-level DPO using the supervision sets in Appendix B and the observation and action interface in Appendix A.4.

C.1 Supervised Fine-Tuning and Training Setup

Trainable policy. We adapt Qwen3-VL-4B-Instruct using LoRA on the query, key, value, and output projections of the language decoder's attention layers. The adapters have rank 16 and scaling parameter 32. The pretrained VLM weights remain frozen, including the visual encoder, and only these policy adapters receive gradients. The shared image restorer, its four expert adapters, and the Localizer also remain fixed.

Training records and serialization. Each SFT example pairs a selected demonstration action with its observation o_t, including the current candidate catalog and preceding structured actions. We apply the model's chat template to the observation and place one action JSON object in the assistant response. The response contains no explanatory text. Refinement fields follow the fixed order

`action`, `region_id`, `expert_weights`, and `fusion_strength`. Demonstrations and preference alternatives use the same compact JSON serialization without optional whitespace. Region identifiers are integers, while expert coefficients and fusion strengths are formatted to two decimal places. A stopping response contains only {`"action":"stop"`}.

SFT supervision. Teacher forcing trains the model to generate the demonstrated action from the corresponding observation. In the notation of Figure 3, x is the observation with its prompt, y_j is response token j, and $y_{<j}$ is its response prefix. The per-example objective is $\ell_{\mathrm{SFT}} = -\sum_{j=1}^{T} m_j \log p_\theta(y_j \mid x, y_{<j})$, where T is the padded response length and m_j selects the complete target JSON and the template's end-of-response token. Prompt, image, and padding positions do not contribute target-token losses. Here, θ denotes the policy parameters used in the figure, corresponding to φ in the main text, and j indexes tokens rather than restoration iterations.

Optimization configuration. Table 7 specifies the training recipe. All stages use four A100 GPUs and an effective batch size of 128. A DPO batch unit is a preference pair, comprising a chosen response and a rejected response under one observation. We increase gradient accumulation in DPO to maintain the same effective batch size with smaller per-device batches. Optimization uses AdamW with $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon_{\mathrm{Adam}} = 10^{-8}$, weight decay 0.01, and gradient-norm clipping at 1.0. A cosine learning-rate schedule follows a linear warm-up over the first 3% of optimizer updates in each stage. The optimizer and schedule restart at stage transitions while the learned adapter weights are retained.

Table 7: Configuration for policy training. Effective batch sizes count demonstration examples for SFT and preference pairs for DPO. Both DPO stages use equal pair weights and the selected-token objective in Section 3.5.

Setting	SFT	Step-level DPO	Trajectory-level DPO
Policy initialization	Pretrained VLM	SFT checkpoint	Step-level checkpoint
Frozen reference	—	SFT checkpoint	Step-level checkpoint
Learning rate	1×10^{-4}	5×10^{-6}	5×10^{-6}
Training epochs	3	1	1
Per-device batch size	8	2 pairs	2 pairs
Gradient accumulation	4	16	16
Effective batch size	128	128 pairs	128 pairs
LoRA dropout	0.05	0	0
DPO coefficient β	—	0.1	0.1

Training uses bfloat16 precision, scaled dot-product attention, and gradient checkpointing. The maximum input-plus-response length is 8,192 tokens, including visual tokens. The image processor uses minimum and maximum pixel budgets of 200,704 and 589,824 per image. Each retained record must fit the length limit with its complete observation and target response. For preference pairs, this check applies to both alternatives, and supervised responses are not truncated. Each training example is processed as a separate sequence. The final checkpoint of each stage initializes the next stage, and the final trajectory-level checkpoint supplies the inference policy.

C.2 Two-Stage Preference Optimization

Both stages use the selected-token objective in Section 3.5, with preference labels supplied by Appendix B.4. They differ in the response span receiving supervision and the restoration outcomes used to rank alternatives. Each retained pair has equal weight. The objective sums selected-token log-probability ratios without token-count normalization or additional score-gap weighting.

Token-mask construction. We locate the supervised span in each serialized response before mapping it to tokens. For a field f, let $[b_f, e_f)$ be the character interval occupied by its complete JSON value, and let $[b_j, e_j)$ be the interval represented by response token y_j. Its mask is $\mu_j^f = \mathbf{1}\{b_j < e_f \wedge e_j > b_f\}$, where $\mathbf{1}\{\cdot\}$ is the indicator function. Thus, a token is selected when its character span intersects the designated value. A token crossing a value boundary is included in full. We derive token positions from the same complete, chat-templated sequence used for likelihood evaluation and express their offsets relative to the assistant response.

Table 8 specifies the spans. String values include their quotation marks, and the expert-weight value includes both array brackets, internal commas, and all four coefficients. Field names and separators outside these value spans are not selected except when they share a boundary-crossing token with the value. For trajectory-level supervision, the selected span instead extends from the JSON object's opening brace through its closing brace. Both DPO stages exclude prompt, image, padding, and chat-template end-of-response tokens. The chosen and rejected responses receive independently aligned masks, since their token counts can differ.

Table 8: Supervised response spans in the two DPO stages. Examples show serialized values; supervised token spans are obtained through the character-to-token alignment described above.

Preference comparison	Selected span	Example or extent
Step: region	`region_id` value	`2`
Step: expert configuration	`expert_weights` value	`[0.25,0.50,0.00,0.75]`
Step: fusion strength	`fusion_strength` value	`0.75`
Step: stop or refine	`action` value	`"stop"` or `"refine"`
Trajectory: alternative action	Complete action JSON	All tokens overlapping the sole differing action's JSON object

Masking selects the log-probability terms entering the loss, while each selected token remains conditioned on its complete preceding response prefix $y_{<j}$. The input action history $\mathcal{H}_t$ describes earlier restoration decisions and belongs to the observation o_t. These earlier actions are distinct from the token prefix within the current JSON response.

Step-level DPO. Each training record is (o, a^+, a^-, f), where o is the common observation, a^+ and a^- are the preferred and rejected actions, and f is the compared field. The supervised set is $\mathcal{T}(a) = \{j : \mu_j^f = 1\}$. For two `REFINE` actions, the one-step utilities determine the preference while the loss selects only the differing configuration value. Comparing alternative expert vectors, for example, supervises the entire weight array while holding the region and fusion strength fixed.

For `STOP`–`REFINE` pairs, preferences compare immediate termination with completed continuation branches, as specified in Appendix B.4. Their mask selects only the `action` value on each side. The trainable policy and its frozen reference both initialize from the SFT checkpoint, with $\beta_s = 0.1$.

Trajectory-level DPO. Let $t^\dagger$ be the sole differing position in a retained trajectory pair (τ^+, τ^-). The complete-trajectory scores determine the preference label for a training example $(o_{t^\dagger}, a_{t^\dagger}^+, a_{t^\dagger}^-)$. Both alternatives share $o_{t^\dagger}$, comprising the common current image, candidate catalog, and structured action history before the differing decision. The loss selects the complete alternative action JSON at that position, assigning the fixed-continuation comparison to the decision that differs between the trajectories.

The trainable policy and its frozen reference both initialize from the final step-level checkpoint, with $\beta_\tau = 0.1$. Each reference snapshot includes the learned policy adapters from its initialization stage and remains fixed throughout the following DPO stage. Reference probabilities are evaluated without gradients. Dropout is disabled for DPO probability evaluation in both policies, and the reference is refreshed only at the transition from step-level to trajectory-level optimization.

D Evaluation Details and Additional Results

This appendix details benchmark evaluation, component ablations, and efficiency measurements, and presents supplementary full-image and patch-level comparisons.

D.1 Benchmarks, Metrics, and Baseline Protocols

Mixed-degradation benchmark. The MiO100-derived benchmark follows the ordered degradation combinations of AgenticIR (Zhu et al., 2025). Groups A and B contain two degradations per image, while Group C contains three. The evaluation partition comprises 640 images in Group A and 400 in each of Groups B and C, totaling 1,440 degraded inputs. Each input is paired with its clean

source image. The original operator order is retained because changing that order changes the restoration problem. For low-resolution inputs, bicubic interpolation aligns the input with the clean-image dimensions before Base restoration. Metrics are computed for each restored image and then averaged within its group.

Single-degradation benchmarks. The evaluation suite used by RAR (Chen et al., 2026) contains eight datasets covering seven restoration tasks: SIDD (Abdelhamed et al., 2018) and Kodak24 (Eastman Kodak Company, 1993) for denoising, DIV2K (Agustsson & Timofte, 2017) for super-resolution, GoPro (Nah et al., 2017) for deblurring, LOL (Wei et al., 2018) for low-light enhancement, SOTS-Outdoor (Li et al., 2019) for dehazing, Rain100 (Yang et al., 2017) for deraining, and Raindrop (Qian et al., 2018) for raindrop removal. RAR reports a combined denoising entry and averages its seven task entries. Each evaluation manifest specifies the input–reference pairs, dataset subset, and preprocessing. Synthetic noise additionally requires its standard deviation and random seed, while super-resolution requires the scale factor and interpolation rule.

Metric implementation. We report the six metrics introduced in Section 4.1 separately; the normalized utility in Appendix B.3 is used for offline preference construction. RAR's released evaluator uses `pyiqa` version 0.1.15.post2, with RGB PSNR, YIQ-luminance SSIM, AlexNet LPIPS (version 0.1), and the KonIQ checkpoint for MANIQA. Its public launcher resizes inputs and references to 256×256 and selects at most the first 300 entries of each subset's manifest.

Baseline provenance. The mixed-degradation baseline values in Table 1 are taken from RAR (Chen et al., 2026), 4KAgent (Zuo et al., 2025), and Xiong et al. (2026). The sources report identical values for overlapping entries; Xiong et al. (2026) report results only for Group C. The single-degradation baseline values in Table 2 are taken from RAR; LD denotes latent diffusion (Rombach et al., 2022). These results follow the evaluation protocols of their respective sources.

D.2 Ablation Settings and Diagnostic Metrics

Preference supervision. Table 9 defines the six variants in Table 3(a). Their shared restorer, expert checkpoints, Localizer, action space, fusion module, and maximum refinement budget are fixed. B and C use the same action preference pairs and DPO update budget, isolating the effect of supervising the differing component. E and F use the same trajectory preference pairs, step-level checkpoint, and DPO update budget, isolating the scope of trajectory supervision. D initializes both the trainable policy and its frozen reference from SFT. In E and F, both initialize from the step-level checkpoint.

Table 9: Training variants for preference optimization. An arrow denotes sequential training. SFT and the two DPO stages use the configurations in Appendix C.

Variant	Training sequence	Preference-loss scope
A	SFT	No preference loss
B	SFT $\rightarrow$ whole-action DPO	Complete action JSON for action pairs
C	SFT $\rightarrow$ step-level DPO	Differing component for action pairs
D	SFT $\rightarrow$ trajectory-level DPO	Sole differing action for trajectory pairs
E	C $\rightarrow$ suffix DPO	Differing action and all subsequent actions
F	C $\rightarrow$ trajectory-level DPO	Sole differing action for trajectory pairs

For suffix supervision in E, the selected log-ratio statistic sums complete-action terms from the differing position to termination. Each action probability is conditioned on that branch's current image and preceding history. This preserves the policy's one-action-at-a-time interface as the two branches produce different images. F applies the loss only at the sole differing position, under the common observation defined in Appendix C.2.

We define the mean refinement count as $N_{\text{ref}} = n^{-1} \sum_{i=1}^{n} T_i$, where n is the number of evaluated images and T_i counts the executed `REFINE` actions for image i.

Local fusion. The local-fusion rerun for Table 3(b) uses two evaluation modes. Complete inference measures the final PSNR and LPIPS of each variant. Single-step replay of the mask and pyramid variants fixes the current image, selected region, expert output, and fusion strength, isolating how a local correction is written back. Removing a mask replaces it with unit weights. Removing the

Laplacian pyramid multiplies the same composite mask by the residual at its original resolution, omitting decomposition and reconstruction.

For each `REFINE` action, the single-expert variant retains the expert with the largest coefficient in the VLM output. Given the predicted weights $\mathbf{w}_t$, let $j^\star = \arg\max_{j\in\{1,\dots,4\}} w_{t,j}$, breaking ties by the fixed expert order. The executed weights are $\mathbf{w}_t^{\text{single}} = w_{t,j^\star} e_{j^\star}$, where e_j is the jth unit vector; the selected region and fusion strength remain as specified by the VLM. During complete inference, the executed single-expert action is added to the history, and subsequent decisions use this variant's updated image. The `STOP` rule is unchanged. Its single-step replay fixes the current image, region, and fusion strength, but recomputes the restored crop and its output-dependent edge mask.

Let Ω_t^{write} be the valid writeback support in image coordinates. We define the set of pixels well restored by Base as

$$\Omega_t^{\text{keep}} = \left\{ p \in \Omega_t^{\text{write}} : \frac{1}{3}\sum_{c=1}^{3} |I_{\text{Base}}(p,c) - G(p,c)| \le \epsilon_{\text{keep}} \right\},$$

where G is the clean reference and c indexes RGB channels. With RGB intensities in $[0,1]$, we fix $\epsilon_{\text{keep}} = 5/255$, corresponding to a mean absolute error of five 8-bit intensity levels across RGB channels. This set depends on Base and the reference, and is shared by all replay variants. The added error is

$$E_{\text{keep},t} = \frac{1}{3|\Omega_t^{\text{keep}}|} \sum_{p\in\Omega_t^{\text{keep}}} \sum_{c=1}^{3} \left[|I_{t+1}(p,c) - G(p,c)| - |I_t(p,c) - G(p,c)| \right]_+ .$$

Only increases in error contribute. Replays with an empty Base-defined pixel set are excluded from this diagnostic.

Boundary error uses the fixed band Ω_t^{bnd} from Appendix B.3, extending eight image pixels to each side of the core boundary. For this diagnostic, grayscale intensities are scaled to $[0,1]$ before applying 3×3 Sobel filters with unit scale. Let $v_{01}(I)$ denote the resulting gradient magnitude. The boundary error is

$$E_{\text{bnd},t} = \frac{1}{|\Omega_t^{\text{bnd}}|} \sum_{p\in\Omega_t^{\text{bnd}}} |v_{01}(I_{t+1})(p) - v_{01}(G)(p)|.$$

This diagnostic measures the post-fusion boundary error; the training penalty P_t^{bnd} measures its increase relative to the input state on the 8-bit gradient scale. Both E_{keep} and E_{bnd} are multiplied by 10^3 for tabular display. Replay statistics are averaged within each source and then across sources, preventing images with more sampled states from receiving greater weight.

D.3 Inference Efficiency and Cost Measurement

End-to-end inference. We measure inference time on NVIDIA A100 GPUs using a common input manifest and a batch size of one. Models are loaded before timing and warmed up on separate images. We synchronize the GPU immediately before and after each timed call. RegionAgent's elapsed time includes Base restoration, all Localizer and VLM calls, regional restoration, and DSRF, including the VLM call that emits `STOP`. Disk access, model loading, and offline quality scoring are excluded.

Table 10 reports mean and 95th-percentile latency and peak GPU memory. RegionAgent retains $T_{\max} = 8$, and the AgenticIR configuration uses Qwen32B for decision-making. RAR is evaluated with its iterative inference procedure (Chen et al., 2026). Each baseline is timed using its complete implementation and stopping rule. RegionAgent requires 18.60 seconds per image on average and 9.28 GB of peak GPU memory, compared with 299.15 seconds and 108.90 GB for AgenticIR.

Table 10: End-to-end inference latency and peak GPU memory on a common evaluation manifest.

Method	Mean time (s)	P95 time (s)	Peak GPU memory (GB)
InstructIR	0.11	0.21	0.84
AgenticIR (Qwen32B)	299.15	373.01	108.90
RAR	3.25	3.78	37.21
RegionAgent	18.60	22.03	9.28

D.4 Qualitative Comparisons

D.4.1 Full-Image Comparisons

Figures 6–9 present full-image comparisons with InstructIR, AdaIR, AgenticIR, and RAR under mixed degradations. Across the illustrated scenes, RegionAgent suppresses residual haze and rain streaks while recovering image contrast and structural detail.

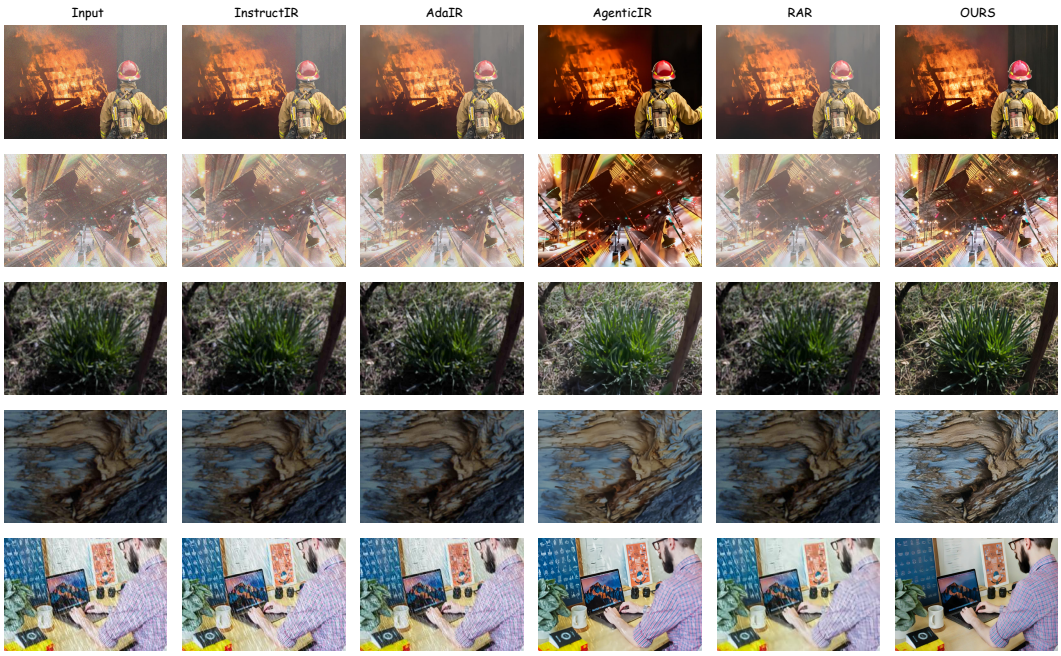

Figure 6: **Full-image qualitative comparisons under mixed degradations**, including haze, low light, noise, and blur.

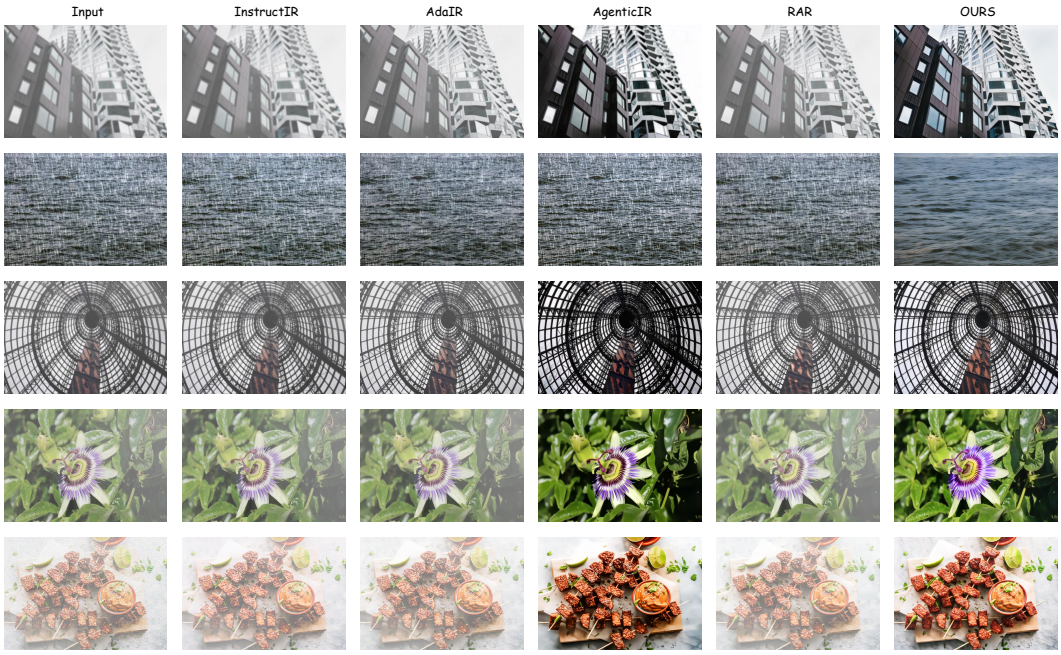

Figure 7: **Full-image qualitative comparisons under mixed degradations**, including rain, haze, low light, noise, and low resolution.

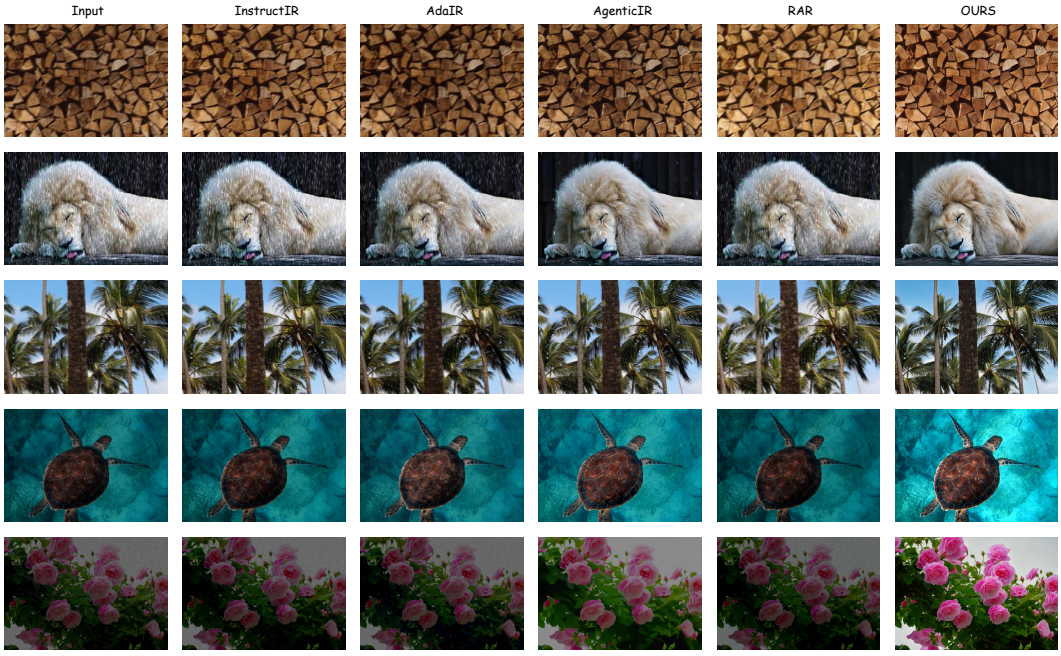

Figure 8: **Full-image qualitative comparisons under mixed degradations**, including rain, blur, low light, and low resolution.

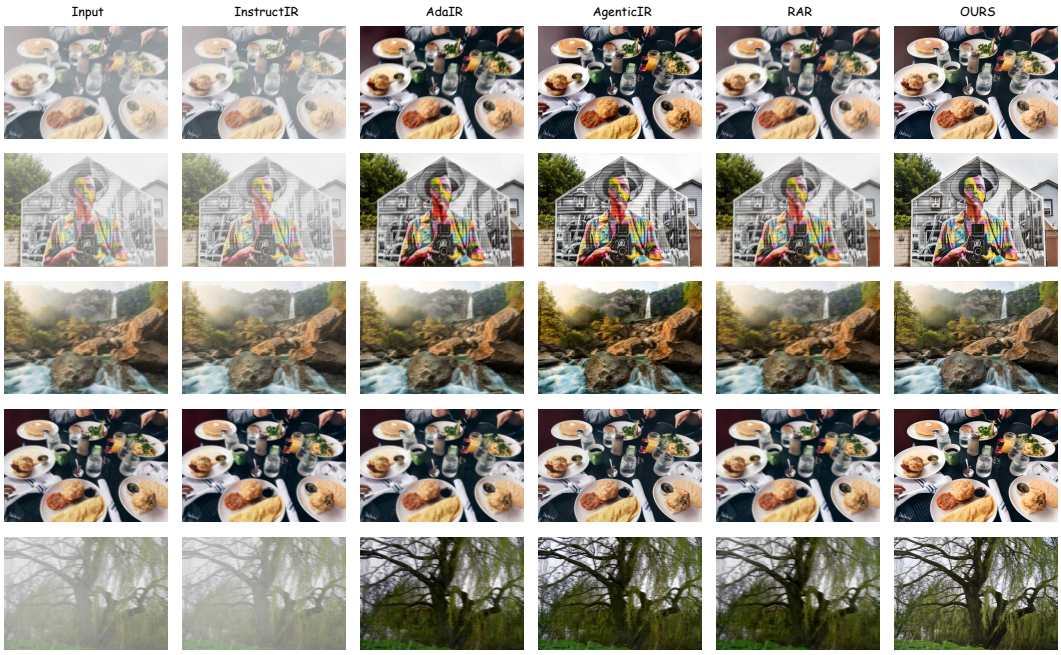

Figure 9: **Full-image qualitative comparisons under mixed degradations**, including haze, blur, and low resolution.

D.4.2 Patch-Level Comparisons

Figures 10–13 provide enlarged comparisons of the regions marked by red boxes, with AutoDIR included as an additional baseline. The crops highlight RegionAgent's noise suppression and preservation of local contours and fine textures under different degradation combinations.

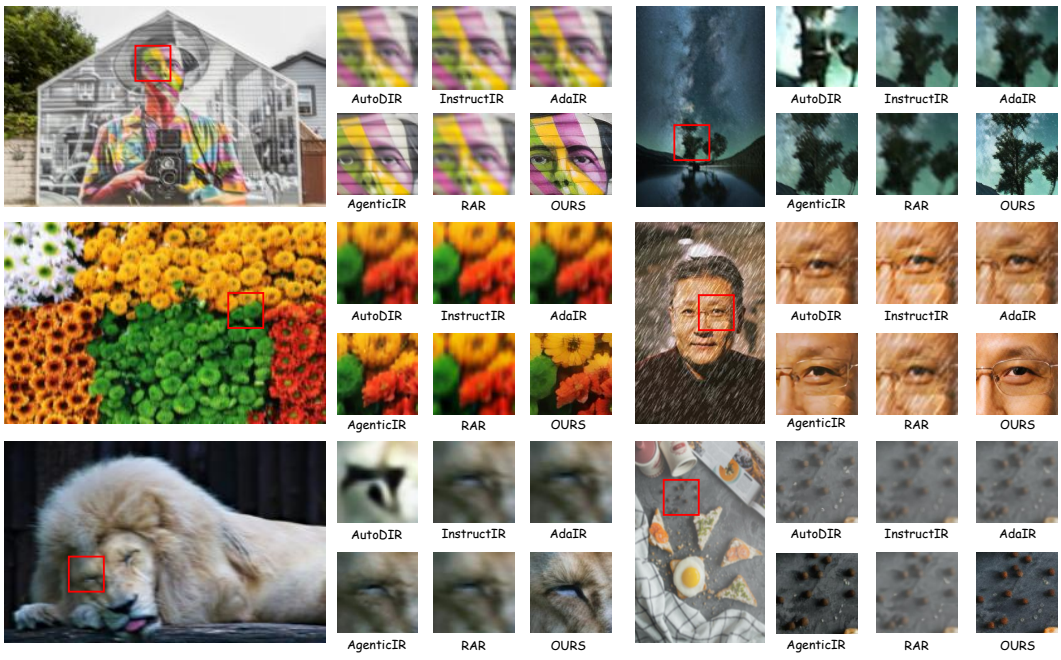

Figure 10: **Patch-level qualitative comparisons under mixed degradations**, including blur, rain, haze, and low resolution.

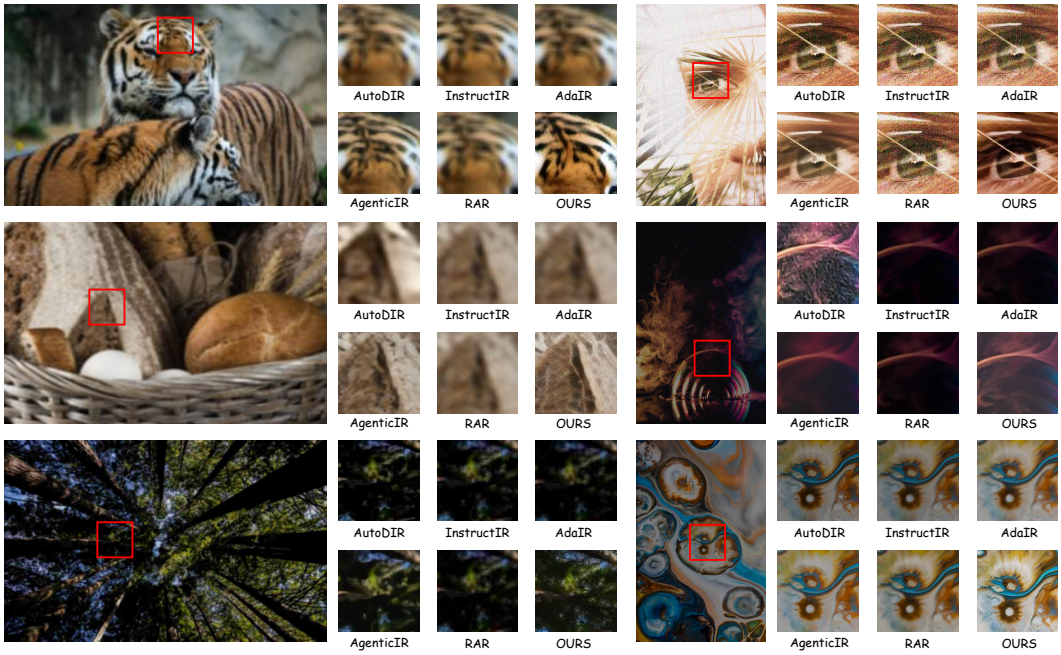

Figure 11: **Patch-level qualitative comparisons under mixed degradations**, including blur, noise, low light, and JPEG compression artifacts.

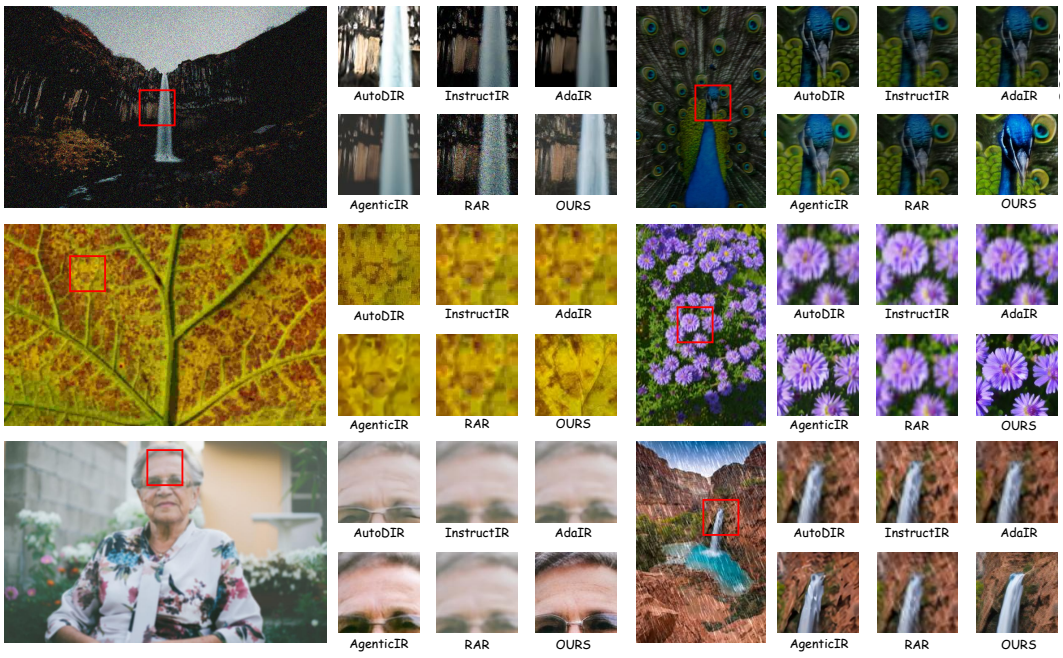

Figure 12: **Patch-level qualitative comparisons under mixed degradations**, including low light, noise, blur, and haze.

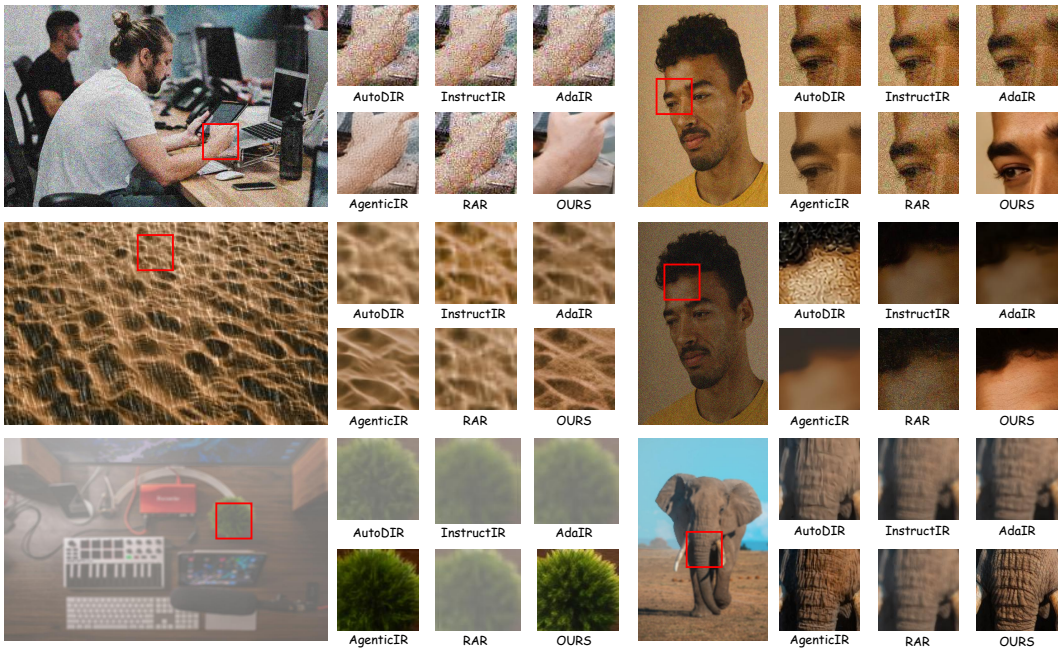

Figure 13: **Patch-level qualitative comparisons under mixed degradations**, including blur, haze, rain, JPEG compression artifacts, noise, and low resolution.