Learning Where and What to Restore for Composite Image Restoration

Jiachen Jiang[1,*] **Tianyu Ding**[2,*,†] **Ke Zhang**[3] **Jinxin Zhou**[2]
Tianyi Chen[2] **Ilya Zharkov**[2] **Zhihui Zhu**[1] **Luming Liang**[2,†]
[1]The Ohio State University [2]Microsoft [3]Johns Hopkins University



Figure 1: Composite image restoration requires jointly deciding *where* and *what* to restore. The degraded input (center) is processed along two complementary axes. **Left (where):** each spatial patch is routed to one of the capacity-graded experts (e.g. E_1 to E_4, simple to complex). Flat regions are handled by lightweight experts, while detailed regions activate higher-capacity ones. **Right (what):** the expert chosen at each decoder block traces a curve across capacity levels; each elemental degradation (Low-light, Rain) follows its own characteristic trajectory, and the composite Low+Rain path (dashed) blends those of its single-degradation constituents.

Abstract

Real-world degraded images often contain multiple co-occurring degradation types, making composite image restoration fundamentally different from the single-degradation setting assumed by most existing all-in-one methods. These methods typically apply uniform spatial computation and single-label task conditioning, limiting both spatial adaptivity and explicit modeling of degradation mixtures. We propose CART (Composite-Adaptive Routing and Task Conditioning), a unified framework that jointly decides *where* to spend computation and *what* degradation cues should guide restoration: a spatial mixer routes patches to graded-capacity experts by local restoration difficulty, and a channel mixer routes each image through degradation-specific experts, conditioned on the global task feature with multi-hot classification supervision. Empirically, CART achieves state-of-the-art performance for composite degradation restoration on CDD-11 and remains competitive on the conventional 3-task and 5-task all-in-one benchmarks.

1 Introduction

Real-world degraded images rarely contain a single isolated corruption. Image restoration aims to recover clean images from degraded observations, and recent restoration models have achieved strong results across diverse settings [22, 43, 55, 6, 58]. Yet most existing methods are developed for specific degradation types or assume one dominant corruption, which

*Equal contribution. This work was done when Jiachen Jiang was an intern at Applied Sciences Group, Microsoft.
†Corresponding authors.

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

limits their practical use in autonomous driving, adverse-weather perception [7, 39] and nighttime surveillance [19], where low-light, haze, rain, blur, and other degradations may co-exist [2, 52, 53, 15, 44]. Figure 2 illustrates this phenomenon on a real-world dataset [59], where rain, snow, haze, and low-light co-occur in varying proportions rather than one dominant corruption. We refer to this setting as *composite image restoration*. Compared with the single-degradation setting assumed by most prior work, composite restoration requires the model to disentangle overlapping artifacts with different spatial scales and different restoration requirements.

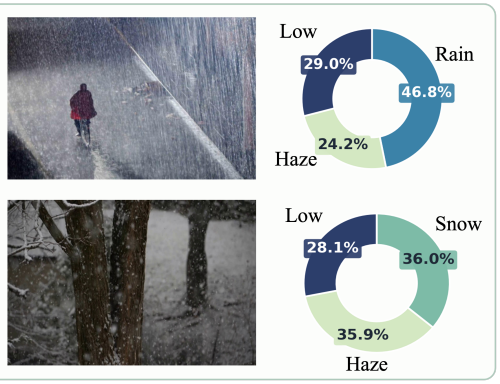


Figure 2: Real-world samples [59] contain multiple co-occurring degradations rather than one dominant corruption; proportions are predicted by our channel mixer (Section 3.2).

Recent all-in-one image restoration methods attempt to address multiple tasks within a unified model [21, 33, 57, 8, 51, 38, 24]. However, a key mismatch remains between their design assumptions and the composite setting. Different degradations place different demands on the network: denoising and deraining emphasize precise local detail processing, whereas deblurring, dehazing, and low-light enhancement depend more strongly on broad spatial context and image-level statistics [48]. Still, representative architectures largely apply a shared computation pattern across spatial regions rather than allocating computation according to local restoration difficulty [21, 33, 8, 38, 24]. Moreover, many conditioning-based methods summarize the degradation with one global signal, such as an image-level prompt or instruction [33, 40, 8, 1, 24], which is insufficient when multiple degradation types are present at once. Together, these limitations indicate that current composite restoration methods lack joint spatial and task adaptivity.

We address these limitations with CART, a unified restoration framework that adapts along two complementary axes: *where* the model spends computation, and *what* degradation types it specializes for. A *spatial mixer* routes patches to experts of graded capacity by local restoration difficulty. A *channel mixer* predicts a multi-hot degradation vector at the bottleneck and uses the pre-classifier feature to route each image through degradation-specialized channel experts at every decoder block. The resulting model adapts both spatially and across degradation types to mixed degradations without requiring manual task specification at inference time.

We evaluate CART on CDD-11 [15], a benchmark that covers single, double, and triple combinations of low-light, haze, rain, and snow. We further evaluate it on the conventional 3-task and 5-task all-in-one benchmarks spanning denoising, deraining, dehazing, deblurring, and low-light enhancement. Across these settings, CART sets the state of the art on CDD-11 and remains competitive on the conventional benchmarks, while remaining the most efficient among compared methods in FLOPs and runtime.

We summarize our contributions as follows:

- We formulate composite image restoration as requiring two complementary forms of adaptivity: spatial adaptivity to allocate computation by local difficulty, and task adaptivity to disentangle the compositional identity of co-occurring degradations.
- We propose CART, a unified framework that jointly learns patch-level compute allocation and degradation-aware expert selection. Patch-wise routing allocates computation by local restoration difficulty, while image-level channel routing makes hard expert selections from a task feature whose geometry is shaped by multi-hot degradation supervision, so no task labels are required at inference.
- We validate CART on CDD-11 and the conventional 3-task and 5-task all-in-one benchmarks. CART achieves state-of-the-art results on composite restoration and remains competitive on standard multi-degradation benchmarks, while remaining the most efficient among compared methods in FLOPs and runtime. Routing interventions confirm that the gains stem from the learned assignments rather than from the mixture of experts.

2 Related work

Prior work on all-in-one restoration is most related to our setting along two axes: how the model represents degradation identity, and how it allocates computation. Composite restoration requires both, since the model must infer which degradations co-occur and decide where to intensify processing.

All-in-one restoration and task conditioning. All-in-one restoration methods seek a single model that can cover multiple degradation types. Early approaches such as IPT [5] and AirNet [21] rely on a shared backbone to handle several tasks, while later methods inject task cues through prompts, instructions, or auxiliary descriptors, as in PromptIR [33], PromptRestorer [40], IDR [57], InstructIR [8], multimodal conditioning [1], DeFusion [24] and AdaIR [10]. These methods broaden the scope of all-in-one restoration, but most summarize the input with a single global condition or treat degradation cues independently. Such designs are effective when one degradation dominates, yet they are less well matched to composite inputs, where multiple degradation types may be active simultaneously and must be represented jointly rather than collapsed into one image-level prompt. Recent methods learn degradation-agnostic representations through Wasserstein-barycenter alignment (BaryIR [37]), channel-wise functional decomposition (MIRAGE [34]), or residual-conditioned optimal transport (DA-RCOT [36]); these objectives are complementary to the input-dependent execution of CART.

Adaptive computation. Dynamic computation [4, 35] allocates variable capacity according to input complexity. In restoration, MGA [17] uses a mask predictor to select under-resolved patches for a dedicated refine-net, Path-Restore [50] routes local regions through different depths, CAMixerSR [42] classifies patches as hard or easy for super-resolution, and MoCE-IR [51] routes whole images to experts of different complexity and receptive field. These methods confirm that restoration benefits from content-dependent computation, but their routing granularity is typically binary (skip/refine or easy/hard), and they address only *where* more computation should be spent without explicitly modeling *what* degradation types are present. Combining spatial and channel processing is not new in itself: for joint compression and restoration, [56] pairs channel-wise group attention with spatially decoupled attention, but applies both identically to every input, whereas CART makes conditional, capacity-graded routing decisions along both axes.

Composite degradation restoration. Composite restoration differs from generic all-in-one restoration in that the degradation label is itself *a set* rather than a single category, requiring the model to represent and disentangle co-occurring artifacts. Methods such as TransWeather [39], WeatherDiff [31], and WGWSNet [59] address multiple adverse-weather conditions, but they focus on a limited weather family and typically treat each image as having one dominant type. More directly related to our setting, OneRestore [15] introduces the CDD-11 benchmark covering mixtures of low-light, haze, rain, and snow, and conditions restoration on scene-level descriptors; AllRestorer [27] similarly relies on externally derived scene attributes; DPL [44] explores dual-level prototypes for composite inputs. These methods require additional descriptors to identify the active degradation mixture. BioIR [9] avoids external descriptors by using a bio-inspired architecture, but still lacks explicit task identification. DCPT [16] takes a different route by using degradation classification as a pre-training objective, yet the classification signal does not directly control the restoration pathway. CART instead couples both requirements: a multi-hot classification head infers the active degradation set and feeds it into task-conditioned channel routing, while patch-level spatial routing with a graded capacity spectrum allocates computation according to local restoration difficulty.

3 Method

Let $\mathcal{T} = \{T_i\}_{i=1}^{M}$ denote a set of M degradation types, and let $\mathcal{D} \subseteq \mathcal{T}$ ($|\mathcal{D}| \geq 1$) be the unknown active subset. Given a clean image I, a compositely degraded observation $I_{\mathcal{D}}$ is modeled as

$$I_{\mathcal{D}}(x) \;=\; \mathcal{H}_{\mathcal{D}}\big(I;\, x\big), \tag{1}$$

where x indexes spatial location and $\mathcal{H}_{\mathcal{D}}$ denotes the joint degradation operator induced by $\mathcal{D}$, whose effect varies across the image. The severity of $\mathcal{H}_{\mathcal{D}}$ is *spatially heterogeneous*: recovering fine textures demands more capacity than correcting flat regions, while the active subset $\mathcal{D}$ is itself *unobserved* at inference. We seek a single model f_θ such that $\hat{I} = f_\theta(I_{\mathcal{D}}) \approx I$ without access to either a severity map or the degradation label $\mathcal{D}$.

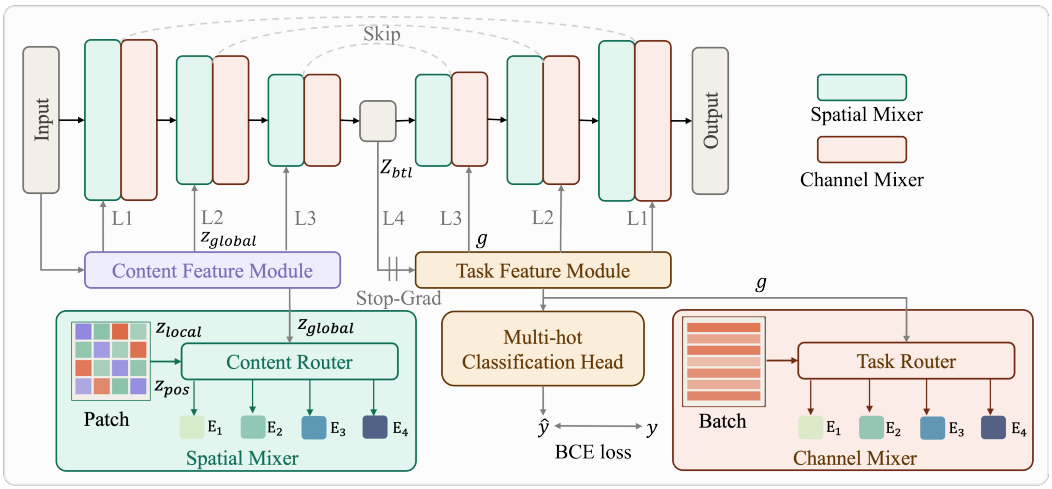


Figure 3: Overview of CART. A U-Net encoder-decoder alternates spatial and channel mixer blocks. The *spatial mixer* routes each patch to one of K experts of graded capacity by local restoration difficulty. A task feature module extracts a global task feature $\boldsymbol{g}$ at the bottleneck and supervises it with multi-hot degradation classification; the *channel mixer* then uses this feature to route each image through one of K channel-attention experts in the decoder.

Two challenges follow directly from (1). First, the effective severity of $\mathcal{H}_{\mathcal{D}}$ varies across spatial locations x: a uniform sky requires minimal computation to restore, while a building with dense edges and fine detail demands substantially more. Applying a fixed computational budget uniformly across all regions inevitably wastes capacity on simple areas and starves complex ones. Second, the active subset $\mathcal{D}$ is unknown at inference and combinatorial in nature—M elemental types yield $2^M - 1$ possible degradation mixtures, each requiring qualitatively different processing. Without an explicit mechanism to disentangle which degradations are present, a shared set of parameters is forced to compromise across tasks rather than specialize.

CART introduces two complementary mixer modules to tackle these challenges as shown in Figure 3. A *spatial mixer* splits features into patches and routes each patch to one of K spatial attention modules of increasing capacity, spending more computation on complex regions and less on simple ones (Section 3.1). A *channel mixer* uses a predicted multi-hot vector $\hat{\boldsymbol{y}} \in [0,1]^M$ to route each image through different channel attention modules, so that different degradation types activate different parameters instead of competing over shared ones (Section 3.2). The two mixers alternate within the backbone, letting the model jointly refine *where* computation is allocated and *what* degradation cues guide restoration at each feature scale.

3.1 Spatial mixer: where to allocate computation

In composite-degraded images, the spatial distribution of restoration difficulty is more heterogeneous than in the single-degradation case: rain-streaked edges, haze-covered flat regions, and areas where both overlap present qualitatively different demands within the same image. Prior spatially adaptive methods typically make binary routing decisions (skip/refine or easy/hard), which cannot capture this finer-grained variation in difficulty. We propose a spatial mixer in which each block contains K experts forming a graded capacity spectrum, and a lightweight router assigns each patch to exactly one expert based on local restoration difficulty.

Complexity experts. The K experts are windowed self-attention blocks that share the same attention structure but differ in hidden dimension d_k. We space $\{d_k\}_{k=0}^{K-1}$ linearly from d/K to d, where d is the block (residual) channel dimension, forming K ordered capacity levels rather than independent or uniform experts. This graded design lets the model match each patch to the minimal sufficient expert: simple regions (e.g., uniform sky) are handled by a low-dimensional expert, while complex regions (e.g., textured edges under mixed degradations) receive full-capacity attention. Selecting the appropriate expert for each patch, however, requires information beyond its local feature alone, which motivates the router design below.

Content router. The router must judge patch-level restoration difficulty, yet features in deeper encoder layers have progressively lost the local spatial detail. We address this by supplementing each patch's local representation with a content feature module drawn from first-level encoder features, which still retain fine-grained structural information. Concretely, the router receives three inputs: (i) a *local embedding* $\boldsymbol{z}_{\text{local}} \in \mathbb{R}^{C_\ell}$, obtained by average-pooling the residual stream features within each window; (ii) a *global content descriptor* $\boldsymbol{z}_{\text{global}} \in \mathbb{R}^{C_g}$, extracted from first-level encoder features and interpolated to each window position; and (iii) a *positional encoding* $\boldsymbol{z}_{\text{pos}} \in \mathbb{R}^2$, encoding normalized spatial coordinates. A two-layer MLP maps their concatenation to K routing logits:

$$\boldsymbol{l} = \text{MLP}([\boldsymbol{z}_{\text{local}},\ \boldsymbol{z}_{\text{global}},\ \boldsymbol{z}_{\text{pos}}]) \ \in \ \mathbb{R}^K. \tag{2}$$

At inference, each patch is hard-assigned to $\arg\max_k l_k$. During training, Gumbel-Softmax [18] makes this discrete assignment differentiable, and a standard load-balancing loss [12] prevents the router from collapsing to a single expert (details in Section A).

3.2 Channel mixer: what degradation cues guide restoration

In composite-degraded images, the active degradation set $\mathcal{D}$ is unobserved at inference and varies compositionally: an image corrupted by haze+rain is related to both haze-only and rain-only inputs, yet distinct from either. Prior all-in-one methods condition restoration on a single global cue such as a one-hot label, text prompt, or scene descriptor, which treats every mixture as an indivisible category and discards this compositional structure. We propose a channel mixer that uses a global task feature $\boldsymbol{g}$, extracted by a task feature module at the bottleneck and supervised by multi-hot degradation classification, to route each image through one of K channel-attention experts.

Task-aware routing signal. A task feature module extracts a global task feature $\boldsymbol{g}$ from the bottleneck feature $\boldsymbol{Z}_{\text{btl}}$, but $\boldsymbol{g}$ inherits the entanglement of content, degradation, and reconstruction cues from $\boldsymbol{Z}_{\text{btl}}$ and provides no *prior* for what the channel router needs. We therefore attach a classification head on top of $\boldsymbol{g}$ and supervise it with a degradation-recognition objective; under this classification supervision, $\boldsymbol{g}$ is shaped toward the neural-collapse geometry [32, 60] in which same-class features converge and different classes separate, so that $\boldsymbol{g}$ encodes task information. We apply a stop-gradient on $\boldsymbol{Z}_{\text{btl}}$ before the task feature module, so the classification objective does not back-propagate into the shared encoder and the bottleneck feature retains its reconstruction signal.

Compositional task representation. A natural classifier would use one output per degradation combination, but this has two drawbacks for composite restoration: the number of categories grows as 2^M-1, and each mixture is treated as an independent class, discarding the compositional structure that links, e.g., haze+rain to both haze-only and rain-only. We instead supervise M independent outputs with binary cross-entropy, producing $\hat{\boldsymbol{y}} \in [0,1]^M$. Under this multi-hot supervision, the global task feature $\boldsymbol{g}$ for a composite input aligns with components of its constituent single-degradation features, so parameter sharing across related tasks emerges as a property of the learned representation rather than a fixed architectural choice. A PCA projection of $\boldsymbol{g}$ (Figure 4) supports this: across CDD-11 conditions, clusters for different degradations are well separated, and each composite-degradation cluster lies between its constituent single-degradation ones. We do not claim that the elemental directions are orthogonal, which multi-hot BCE does not guarantee and the router does not need; the relevant property is *compositional decodability*, i.e., each elemental degradation stays recoverable from $\boldsymbol{g}$ when others co-occur, which Section B.4 tests quantitatively.

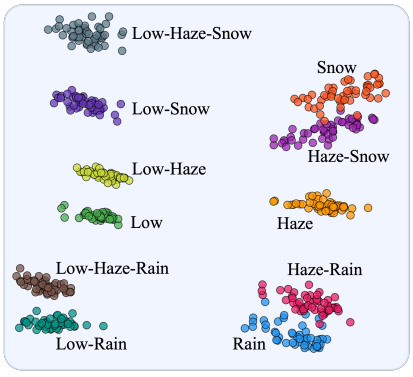


Figure 4: PCA of the task feature $\boldsymbol{g}$. Composite clusters lie between their single-degradation constituents.

Channel experts and image-level routing. Unlike spatial difficulty, which varies within an image, the active degradation set $\mathcal{D}$ is a global property of the image, so we route at the image level rather than per patch. A task router takes the concatenation of $\boldsymbol{g}$ and the global-average-pooled feature map and produces K logits; hard arg max is used at inference, with Gumbel-Softmax [18] and a standard load-balancing loss during training (same machinery as the spatial mixer; details in Section A). The prediction $\hat{\boldsymbol{y}}$ only supervises $\boldsymbol{g}$; the router makes a hard selection from $\boldsymbol{g}$ rather than blending experts by $\hat{\boldsymbol{y}}$. The K experts are channel-attention blocks sharing the same graded capacity schedule as the

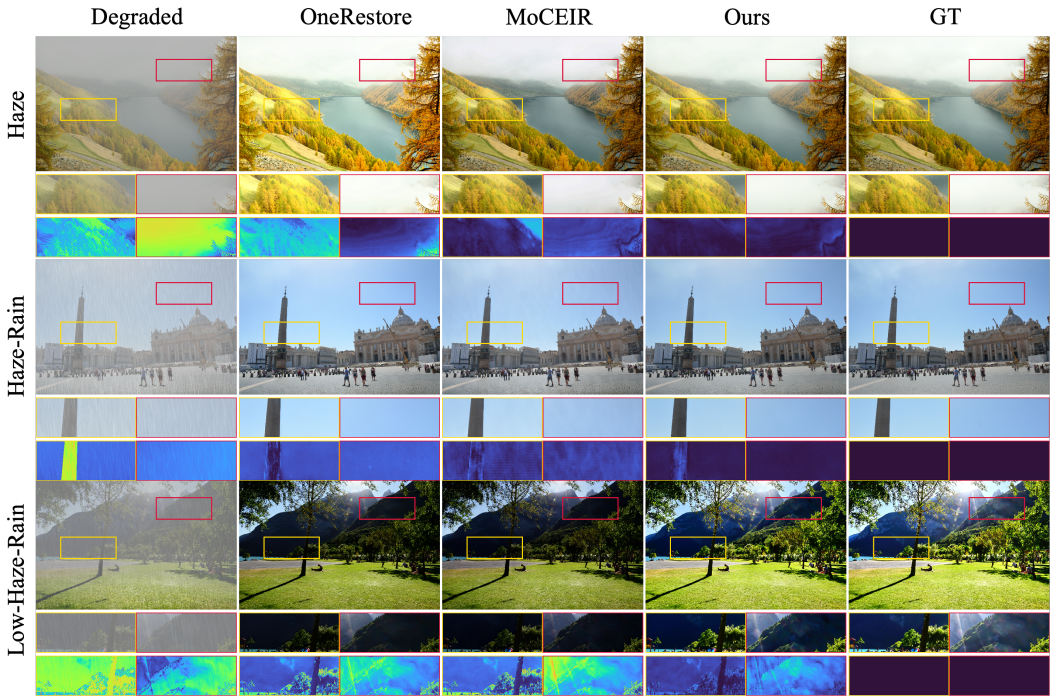


Figure 5: Visual results on CDD-11. CART vs. OneRestore [15] and MoCE-IR [51] on three cases: haze, haze+rain, and low+haze+rain. For each case, the top strip shows the degraded input, the restorations from each method, and the ground truth; the middle strip zooms into regions; the bottom strip shows the per-pixel error map to ground truth (darker = smaller residual).

spatial experts (hidden dimensions from d/K to d), but operating along the channel axis, so the spatial and channel mixers are complementary in the dimension they modulate.

Because this routing is applied at every decoder block, each elemental degradation induces a distinct routing path, and composite inputs produce paths that blend the constituent patterns. This yields task-specific specialization within a shared network without separate branches or task labels at inference, so different degradations activate their own parameter subsets rather than competing for a shared pool (visualized in Section 4.3).

3.3 Training objective

The training objective combines reconstruction losses on the output image with auxiliary losses that supervise the two mixer mechanisms:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda_{\text{ms}}\,\mathcal{L}_{\text{ms}} + \lambda_{\text{cdrl}}\,\mathcal{L}_{\text{cdrl}} + \lambda_{\text{lb}}\,\mathcal{L}_{\text{lb}} + \lambda_{\text{task}}\,\mathcal{L}_{\text{task}}. \tag{3}$$

Restoration quality is supervised by three terms: a smooth-ℓ_1 pixel loss $\mathcal{L}_{\text{recon}}$, a multi-scale structural similarity loss $\mathcal{L}_{\text{ms}}$ [45] that encourages perceptually consistent structure, and the composite-degradation contrastive loss $\mathcal{L}_{\text{cdrl}}$ [15], which contrasts the output against all degradation types rather than only those present in the input. Two auxiliary terms supervise the mixers: $\mathcal{L}_{\text{lb}}$ is a standard load-balancing loss [12] averaged over all routed stages (both spatial and channel), and $\mathcal{L}_{\text{task}}$ is the binary cross-entropy on the multi-hot degradation prediction at the bottleneck.

4 Experiments

4.1 Setup

CART targets *composite* image restoration, where multiple degradations co-occur in a single input. We thus evaluate primarily on the composite setting, and additionally on conventional all-in-one benchmarks to verify the design generalizes to inputs with a single active degradation.

Table 1: Results on CDD-11 (PSNR / SSIM). All methods are trained from scratch on CDD-11. L: low-light, H: haze, R: rain, S: snow. *Avg* is the macro-average over all 11 conditions. Row colors mark the overall best, second-best, and third-best methods (ranked by Avg PSNR).

	Single degradation				Two degradations					Three degradations		
Methods	L	H	R	S	L+H	L+R	L+S	H+R	H+S	L+H+R	L+H+S	**Avg**
AirNet [21]	24.83 / .778	24.21 / .951	26.55 / .891	26.79 / .919	23.23 / .779	22.82 / .710	23.29 / .723	22.21 / .868	23.29 / .901	21.80 / .708	22.24 / .725	23.75 / .814
PromptIR [33]	26.32 / .805	26.10 / .969	31.56 / .946	31.53 / .960	24.49 / .789	25.05 / .771	24.51 / .761	24.54 / .924	23.70 / .925	23.74 / .752	23.33 / .747	25.90 / .850
WGWSNet [59]	24.39 / .774	27.90 / .982	33.15 / .964	34.43 / .973	24.27 / .800	25.06 / .772	24.60 / .765	27.23 / .955	27.65 / .960	23.90 / .772	23.97 / .771	26.96 / .863
WeatherDiff [31]	23.58 / .763	21.99 / .904	24.85 / .885	24.80 / .888	21.83 / .756	22.69 / .730	22.12 / .707	21.25 / .868	21.99 / .868	21.23 / .716	21.04 / .698	22.49 / .799
OneRestore [15]	26.48 / .826	32.52 / .990	33.40 / .964	34.31 / .973	25.79 / .822	25.58 / .799	25.19 / .789	29.99 / .957	30.21 / .964	24.78 / .788	24.90 / .791	28.47 / .878
MoCE-IR [51]	**27.26** / .824	32.66 / .990	34.31 / .970	35.91 / .980	26.24 / .817	26.25 / .800	26.04 / .793	29.93 / .964	30.19 / .970	25.41 / .789	25.39 / .790	29.05 / .881
CART (ours)	27.05 / **.836**	**34.52 / .994**	**35.46 / .976**	**37.54 / .983**	**26.40 / .836**	**26.40 / .819**	**26.33 / .814**	**31.61 / .974**	**32.28 / .978**	**25.93 / .813**	**25.96 / .814**	**29.95 / .894**

Datasets. *Composite restoration.* CDD-11 [15] contains images with one to three co-occurring degradations from low-light (L), haze (H), rain (R), and snow (S), covering 11 conditions in total: 4 single, 5 two-way, and 2 three-way. *Conventional all-in-one (3D/5D).* Following [33], the 3-task setting (3D) combines Gaussian denoising on BSD400 [3]+WED [26] with $\sigma \in \{15, 25, 50\}$ (tested on BSD68 [28]), deraining on Rain100L [49], and dehazing on SOTS [20]; the 5-task setting (5D) adds deblurring on GoPro [30] and low-light enhancement on LOL-v1 [46].

Implementation details. CART adopts a four-level U-Net with base channel width d=48 and block counts $[2, 3, 3, 4]$ from top to bottom. Each routed layer uses K=4 experts whose hidden dimensions grow linearly as $[12, 24, 36, 48]$ at the base level and scale by 2^ℓ at level ℓ. We train with Adam (β_1=0.9, β_2=0.999) at batch size 32, with an initial learning rate of 2×10^{-4} and cosine annealing after a 15-epoch warmup. Inputs are 128×128 random crops with horizontal and vertical flips, and the Gumbel-Softmax temperature τ is annealed from 2.5 to 0.7 over training. Loss weights are λ_{ms}=0.3, λ_{cdrl}=0.1, λ_{lb}=0.01, and λ_{task}=0.01. We train one unified model for 800 epochs on CDD-11 and one for 120 epochs on each of the 3D and 5D benchmarks. We report PSNR (dB) and SSIM, with a macro-average over the 11 CDD-11 conditions and per-task averages on 3D/5D.

4.2 Main results

Composite degradations. On the composite setting, we evaluate CART on CDD-11 [15] (Table 1) against representative all-in-one and composite restoration methods: AirNet [21], PromptIR [33], WGWSNet [59], WeatherDiff [31], OneRestore [15], and the strongest prior baseline MoCE-IR [51]. CART attains the highest average PSNR of 29.95 dB (+0.90 dB over MoCE-IR) and ranks best on 10 of 11 conditions in PSNR and on all 11 in SSIM. The advantage widens on multi-degradation mixtures (e.g., +1.68 dB on H+R and +2.09 dB on H+S over MoCE-IR), where multi-hot task conditioning preserves the identities of co-occurring degradations that single-cue baselines collapse.

Conventional all-in-one (3D and 5D). To verify that the design also generalizes to inputs with a single active degradation, we evaluate on the standard 3-task and 5-task all-in-one benchmarks (Tables 2 and 3). CART attains the best average PSNR on both (+0.37 dB on 3D and +0.29 dB on 5D over MoCE-IR), with the largest gains on tasks demanding fine local detail or large global statistics shifts: heavy-noise denoising (σ=50: +0.90 dB on 3D) and low-light enhancement (+0.87 dB on 5D). On well-saturated tasks such as dehazing and deblurring, CART is on par with the best prior method while remaining ahead on average, consistent with our two-mixer design that adapts capacity to local difficulty and specializes features per degradation type. Overall, CART remains competitive on conventional all-in-one restoration, while its strongest advantage lies in the composite setting for which the dual-routing design is introduced.

Visual results. Figure 5 compares CART to OneRestore and MoCE-IR on three CDD-11 cases of increasing difficulty. On haze, OneRestore introduces a color cast and MoCE-IR leaves residual haze, while CART preserves natural tones. On haze+rain and low+haze+rain, both baselines retain visible streaks and washed-out colors, whereas CART removes streaks cleanly and recovers sharper texture. The error maps mirror these observations: CART's residuals stay visibly darker, confirming that the quantitative gains in Table 1 translate into perceptually meaningful improvements.

Zero-shot real-world transfer. Applied without retraining to the real-world WeatherBench [13], the CDD-11-trained CART reaches 21.66 / 0.735 / 0.376 (PSNR / SSIM / LPIPS) versus 19.96 / 0.727 / 0.390 for MoCE-IR, the strongest baseline in Table 1 (Table 7); the low absolute scores indicate a synthetic-to-real gap (Section 6).

Table 2: Results on the conventional **3-task (3D)** all-in-one benchmark (PSNR / SSIM). Tasks: dehazing on SOTS, deraining on Rain100L, and Gaussian denoising on BSD68 at $\sigma \in \{15, 25, 50\}$. Row colors mark the overall best , second-best , and third-best methods.

Methods	Dehazing SOTS [20]		Deraining Rain100L [49]		Denoising (BSD68 [28]) $\sigma = 15$		$\sigma = 25$		$\sigma = 50$		Average	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
MPRNet [54]	25.28	0.955	33.57	0.954	33.54	0.927	30.89	0.880	27.56	0.779	30.17	0.899
Gridformer [41]	30.37	0.970	37.15	0.972	33.93	0.931	31.37	0.887	28.11	0.801	32.19	0.912
Art-PromptIR [47]	30.83	0.979	37.94	0.982	34.06	0.934	31.42	0.891	28.14	0.801	32.48	0.917
DA-CLIP [25]	29.46	0.963	36.28	0.968	30.02	0.821	24.86	0.585	22.29	0.476	28.58	0.763
UniProcessor [11]	**31.66**	0.979	38.17	0.982	34.08	0.935	31.42	0.891	28.17	0.803	32.70	0.918
PromptIR [33]	30.58	0.974	36.37	0.972	33.98	0.933	31.31	0.888	28.06	0.799	32.06	0.913
MoCE-IR [51]	31.34	**0.979**	38.57	**0.984**	34.11	0.932	31.45	0.888	28.18	0.800	32.73	0.917
CART-3D	30.96	0.977	**38.82**	**0.984**	**34.45**	**0.949**	**32.19**	**0.926**	**29.08**	**0.877**	**33.10**	**0.943**

Table 3: Results on the conventional **5-task (5D)** all-in-one benchmark (PSNR / SSIM). Tasks: the three 3D tasks, plus deblurring on GoPro, and low-light enhancement on LOL-v1. Row colors mark the overall best , second-best , and third-best methods.

Methods	Dehazing SOTS [20]		Deraining Rain100L [49]		Denoising BSD68 [28] (σ=25)		Deblurring GoPro [30]		Low-Light LOLv1 [46]		Average	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
NAFNet [6]	25.23	.939	35.56	.967	31.02	.883	26.53	.808	20.49	.809	27.76	.881
DGUNet [29]	24.78	.940	36.62	.971	31.10	.883	27.25	.837	21.87	.823	28.32	.891
Restormer [55]	24.09	.927	34.81	.962	31.49	.884	27.22	.829	20.41	.806	27.60	.881
MambaIR [14]	25.81	.944	36.55	.971	31.41	.884	28.61	.875	22.49	.832	28.97	.901
Transweather [39]	21.32	.885	29.43	.905	29.00	.841	25.12	.757	21.21	.792	25.22	.836
IDR [57]	25.24	.943	35.63	.965	**31.60**	.887	27.87	.846	21.34	.826	28.34	.893
Gridformer [41]	26.79	.951	36.61	.971	31.45	.885	29.22	.884	22.59	.831	29.33	.904
InstructIR-5D [8]	27.10	.956	36.84	.973	31.40	.873	29.40	.886	23.00	.836	29.55	.908
MoCE-IR [51]	**30.48**	.974	38.04	.982	31.34	.887	**30.05**	**.899**	23.00	.852	30.58	.919
CART-5D (ours)	30.35	**.976**	**38.87**	**.984**	31.59	**.917**	29.70	.895	**23.87**	**.861**	**30.87**	**.927**

4.3 Routing behavior analysis

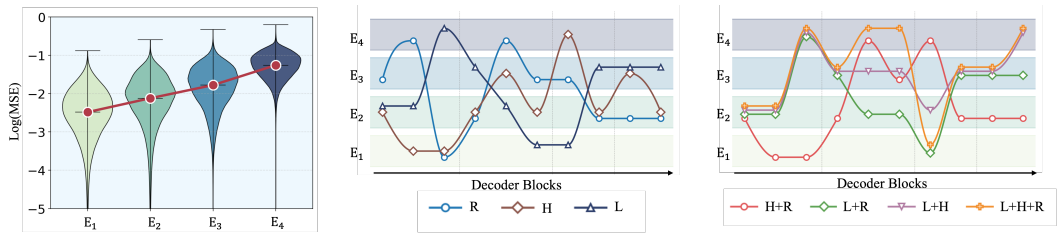
To check whether the two mixers instantiate the intended *where* / *what* decomposition of CART, we probe whether the spatial router routes harder patches to higher-capacity experts and whether the channel router sends different degradations through distinct decoder paths. Figure 6 reports both probes on the CDD-11 test set.

Spatial routing tracks patch difficulty. We measure per-patch difficulty by the log MSE between a degraded patch and its target, then group patches by their assigned expert (Figure 6a). The mean log MSE increases monotonically from E_1 to E_4: easy patches concentrate at the lightweight E_1 while heavily corrupted patches reach the full-capacity E_4. Without any direct supervision on the routing decision, the spatial router matches patch difficulty to expert capacity, consistent with the graded-capacity design of Section 3.1.

The learned spatial assignment matters. To go beyond this correlation, we fix the trained weights and replace the learned patch-to-expert assignment in every spatial-mixer block by a random permutation or by a *reverse-difficulty* assignment (harder patches to smaller experts), both preserving each block's expert-usage histogram and hence its FLOPs. PSNR on CDD-11 drops from 29.95 to 29.36 (-0.59) and 29.29 dB (-0.66), respectively (Table 5): the gain comes from matching patches to suitable capacity, not from the expert mixture.

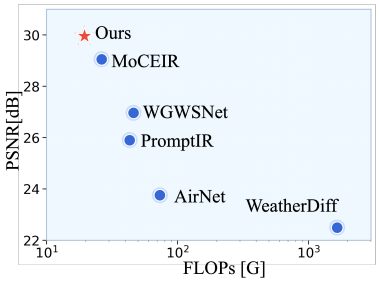
Channel routing reflects compositional task identity. For each task, we record the most-frequently selected expert at each channel-mixer block across the test images, yielding one representative path per task. Single degradations (Figure 6b, left) follow well-separated decoder paths, showing that the multi-hot classification objective distinguishes elemental tasks. Composite inputs (right) trace paths that largely overlap with those of their constituents; e.g., haze+rain blends routes selected by haze and by rain. The composite routing thus inherits elemental structure and recombines it at inference, providing a mechanism for the multi-degradation gains in Table 1.

Channel experts are all used and specialized. Aggregated over all channel-mixer blocks, the four experts receive 18%, 17%, 29%, and 35% of the assignments. Forcing every block to one fixed



(a) Spatial routing. (b) Channel routing: single (left) and composite (right) degradations.

Figure 6: Routing behavior on CDD-11. **(a)** Per-patch log MSE grouped by the spatial expert selected to process the patch; the mean increases monotonically from E_1 to E_4. **(b)** Most-frequent channel expert at each decoder block, one path per degradation type; composite paths largely overlap with those of their constituents. This arg-max path is a per-task summary, not a load distribution; all four experts receive substantial load.



Method	Params (M)	FLOPs (G)	Mem (MB)	Time (ms)
AirNet [21]	8.93	73.42	122	38.44
PromptIR [33]	35.59	43.18	288	35.33
WGWS-Net [59]	19.77	46.15	113	60.71
WeatherDiff [31]	85.61	1658.45	431	383.75
MoCE-IR [51]	25.35	26.29	161	45.46
CART (ours)	32.58	**19.47**	195	**32.11**

Figure 7: Computational efficiency on CDD-11. Left: PSNR vs. FLOPs trade-off (log scale); CART occupies the upper-left corner. Right: parameters, FLOPs, peak memory, and runtime measured on $1\times3\times128\times128$ inputs at batch size 1 on a single A5000 GPU in FP32.

expert lowers the average PSNR by 0.80–3.49 dB, in a degradation-dependent way: forcing E_4 costs 0.23 dB on low light but 8.45, 5.29, and 10.62 dB on haze, rain, and snow (Table 6), so the experts implement functionally different transformations.

4.4 Complexity analysis

Figure 7 shows that CART requires only 19.47 G FLOPs (26% less than MoCE-IR) and runs the fastest at 32.11 ms among all compared methods, while also attaining the highest Avg PSNR on CDD-11. The MoE design stores all K experts even though only one is activated per input, so CART has a higher parameter count (32.58 M) and peak memory (195 MB) than MoCE-IR; we view this as a worthwhile trade since compute and latency dominate at inference.

4.5 Ablation studies

We ablate three design choices on CDD-11: the multi-hot classification head, the two routing branches, and the training objective. Each variant changes only the component under study and shares the backbone, training schedule, and $K=4$ expert configuration with the full model. Scores are averaged within the single-, two-, and three-degradation subsets.

Multi-hot classification supervision. To check whether composite task identity benefits from a compositional representation, we replace the multi-hot BCE head with an 11-class softmax that treats each degradation combination as a separate label. The Avg PSNR drops from 29.95 to 29.15 dB (-0.80 dB), the largest single drop in the ablation, with a wider gap on three-way mixtures (-0.55 dB). Multi-hot supervision lets elemental components be shared across mixtures, whereas the single-label head forces each combination to be learned in isolation.

Spatial / channel mixer design. To isolate the two adaptive decisions, we remove spatial and channel routing separately. For fair comparison, each removed branch (expert widths $[12, 24, 36, 48]$) is replaced by a single FLOPs-equivalent expert with $d=32$. Removing channel routing causes

Table 4: Ablations on CDD-11. Each cell reports PSNR / SSIM averaged over single, two-, and three-degradation conditions; *Avg* is the macro-average over all 11 conditions.

Variant	Single	Two	Three	Avg
CART (full)	**33.64 / .947**	**28.60 / .884**	**25.95 / .814**	**29.95 / .894**
Multi-hot classification supervision				
Single-label (11-class CE)	32.67 / .945	27.83 / .878	25.40 / .805	29.15 / .889
Spatial/channel mixer design				
w/o spatial routing	33.59 / .948	28.58 / .884	25.86 / .812	29.90 / .894
w/o channel routing	33.24 / .947	28.15 / .881	25.72 / .809	29.56 / .892
w/o both routing	32.95 / .946	28.07 / .882	25.79 / .812	29.43 / .893
Training objective				
w/o $\mathcal{L}_{\text{task}}$	33.14 / .944	28.19 / .878	25.48 / .803	29.49 / .889
w/o $\mathcal{L}_{\text{lb}}$	32.88 / .947	28.14 / .883	25.69 / .812	29.42 / .893
w/o $\mathcal{L}_{\text{cdrl}}$	33.15 / .946	28.28 / .881	25.73 / .809	29.59 / .892
w/o $\mathcal{L}_{\text{ms}}$	33.39 / .947	28.55 / .884	25.77 / .813	29.80 / .894

the larger drop (-0.39 dB on Avg), confirming the dominant role of explicit task-aware routing; removing spatial routing accounts for a -0.05 dB drop. Removing both drops Avg by -0.52 dB, slightly above the sum of the individual effects, so the two routers contribute complementary signals and jointly support the *where* / *what* design.

Training objective. To assess each auxiliary loss, we remove one term at a time and keep the rest of the objective unchanged. Removing $\mathcal{L}_{\text{lb}}$ (-0.53 dB) or $\mathcal{L}_{\text{task}}$ (-0.46 dB) causes the largest drops, confirming that both balanced expert usage and explicit multi-hot supervision are central to the routing mechanism. The contrastive term $\mathcal{L}_{\text{cdrl}}$ (-0.36 dB) contributes by sharpening task-specific reconstructions, and the multi-scale SSIM term $\mathcal{L}_{\text{ms}}$ (-0.15 dB) plays a smaller but still positive role.

5 Conclusion

We introduced CART, a composite image restoration framework that jointly decides *where* to spend computation and *what* degradation types to specialize for. By coupling patch-level spatial routing with multi-hot task-conditioned channel routing, CART achieves state-of-the-art performance on CDD-11 and remains competitive on the conventional 3-task and 5-task all-in-one benchmarks, while remaining the most efficient among compared methods in FLOPs and runtime. Promising directions for future work include training and evaluation on real-world degradations and held-out degradation types with perceptual studies, improving expert efficiency for real-time use, and extending CART to video restoration with temporally stable routing.

6 Limitations

Our evaluation targets composite restoration, i.e., identifying and jointly removing multiple co-occurring degradations drawn from a known set of elemental types. CDD-11 systematically varies the number and composition of these degradations, and the 3D and 5D benchmarks show that the routing design also applies to conventional restoration tasks; however, none of these experiments tests generalization to degradation types or data distributions that are absent from training. Such out-of-distribution generalization is a complementary question, which we leave to future evaluation on held-out degradation types and unconstrained real-world images. In zero-shot transfer to the real-world WeatherBench, CART improves over MoCE-IR, but its absolute performance remains limited, revealing a gap between synthetic training data and real degradations; closing this gap would likely require training or fine-tuning on real-world degradation data.

Acknowledgments and Disclosure of Funding

This work was supported by Microsoft, which funded Jiachen Jiang's research internship at the Applied Sciences Group and provided the GPU computing resources used in this work. The authors have no competing interests to declare beyond their listed affiliations.

References

[1] Yuang Ai, Huaibo Huang, Xiaoqiang Zhou, Jiexiang Wang, and Ran He. Multimodal prompt perceiver: Empower adaptiveness, generalizability, and fidelity for all-in-one image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25432–25444, 2024.

[2] Saeed Anwar, Nick Barnes, and Lars Petersson. Attention-based real image restoration. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[3] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2010.

[4] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[5] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *CVPR*, 2021.

[6] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *European Conference on Computer Vision*, pages 17–33. Springer, 2022.

[7] Yu-Wei Chen and Soo-Chang Pei. Always clear days: Degradation type and severity aware all-in-one adverse weather removal. *arXiv preprint arXiv:2310.18293*, 2023.

[8] Marcos V. Conde, Gregor Geigle, and Radu Timofte. Instructir: High-quality image restoration following human instructions. In *Computer Vision – ECCV 2024*, pages 1–21, 2024. URL `https://www.ecva.net/papers/eccv_2024/papers_ECCV/papers/05238.pdf`.

[9] Yuning Cui, Wenqi Ren, and Alois Knoll. Bio-inspired image restoration. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[10] 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 *13th international conference on learning representations, ICLR 2025*, pages 57335–57356. International Conference on Learning Representations, ICLR, 2025.

[11] Huiyu Duan, Xiongkuo Min, Sijing Wu, Wei Shen, and Guangtao Zhai. Uniprocessor: a text-induced unified low-level image processor. In *European Conference on Computer Vision*, pages 180–199. Springer, 2024.

[12] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23 (120):1–39, 2022.

[13] Qiyuan Guan, Qianfeng Yang, Xiang Chen, Tianyu Song, Guiyue Jin, and Jiyu Jin. WeatherBench: A real-world benchmark dataset for all-in-one adverse weather image restoration. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 12607–12613, 2025.

[14] Hang Guo, Jinmin Li, Tao Dai, Zhihao Ouyang, Xudong Ren, and Shu-Tao Xia. Mambair: A simple baseline for image restoration with state-space model. In *European conference on computer vision*, pages 222–241. Springer, 2024.

[15] Yu Guo, Yuan Gao, Yuxu Lu, Huilin Zhu, Ryan Wen Liu, and Shengfeng He. Onerestore: A universal restoration framework for composite degradation. In *Computer Vision – ECCV 2024*, pages 255–272, 2024. URL `https://www.ecva.net/papers/eccv_2024/papers_ECCV/papers/02906.pdf`.

[16] JiaKui Hu, Lujia Jin, Zhengjian Yao, and Yanye Lu. Universal image restoration pre-training via degradation classification. *arXiv preprint arXiv:2501.15510*, 2025.

[17] Xiaotao Hu, Jun Xu, Shuhang Gu, Ming-Ming Cheng, and Li Liu. Restore globally, refine locally: A mask-guided scheme to accelerate super-resolution networks. In *European Conference on Computer Vision*, pages 74–91. Springer, 2022.

[18] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[19] Mohit Lamba and Kaushik Mitra. Restoring extremely dark images in real time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3487–3497, 2021.

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

[21] Boyun Li, Xiao Liu, Peng Hu, Zhongqin Wu, Jiancheng Lv, and Xi Peng. All-in-one image restoration for unknown corruption. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17452–17462, June 2022. URL `https://openaccess.thecvf.com/content/CVPR2022/html/Li_All-in-One_Image_Restoration_for_Unknown_Corruption_CVPR_2022_paper.html`.

[22] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *ICCV Workshops*, 2021.

[23] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12009–12019, 2022.

[24] Wenyang Luo, Haina Qin, Zewen Chen, Libin Wang, Dandan Zheng, Yuming Li, Yufan Liu, Bing Li, and Weiming Hu. Visual-instructed degradation diffusion for all-in-one image restoration. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12764–12777, 2025.

[25] Ziwei Luo, Fredrik K Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B Schön. Controlling vision-language models for universal image restoration. *arXiv preprint arXiv:2310.01018*, 3(8), 2023.

[26] Kede Ma, Zhengfang Duanmu, Qingbo Wu, Zhou Wang, Hongwei Yong, Hongliang Li, and Lei Zhang. Waterloo exploration database: New challenges for image quality assessment models. *IEEE Transactions on Image Processing*, 26(2):1004–1016, 2016.

[27] Jiawei Mao, Yu Yang, Xuesong Yin, Ling Shao, and Hao Tang. Allrestorer: All-in-one transformer for image restoration under composite degradations. *arXiv preprint arXiv:2411.10708*, 2024.

[28] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, volume 2, pages 416–423. Ieee, 2001.

[29] Chong Mou, Qian Wang, and Jian Zhang. Deep generalized unfolding networks for image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17399–17410, 2022.

[30] 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*, pages 3883–3891, 2017.

[31] Ozan Özdenizci and Robert Legenstein. Restoring vision in adverse weather conditions with patch-based denoising diffusion models. *IEEE transactions on pattern analysis and machine intelligence*, 45(8):10346–10357, 2023.

[32] Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117 (40):24652–24663, 2020.

[33] Vaishnav Potlapalli, Syed Waqas Zamir, Salman H. Khan, and Fahad Shahbaz Khan. Promptir: Prompting for all-in-one image restoration. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL `https://papers.nips.cc/paper_files/paper/2023/hash/e187897ed7780a579a0d76fd4a35d107-Abstract-Conference.html`.

[34] Bin Ren, Yawei Li, Xu Zheng, Yuqian Fu, Danda Pani Paudel, Hong Liu, Ming-Hsuan Yang, Luc Van Gool, and Nicu Sebe. Efficient degradation-agnostic image restoration via channel-wise functional decomposition and manifold regularization. In *International Conference on Learning Representations*, pages 126404–126430, 2026.

[35] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR)*, 2017.

[36] 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.

[37] Xiaole Tang, Xiaoyi He, Jiayi Xu, Xiang Gu, and Jian Sun. Learning continuous Wasserstein barycenter space for generalized all-in-one image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(7):8395–8410, 2026.

[38] Xiangpeng Tian, Xiangyu Liao, Xiao Liu, Meng Li, and Chao Ren. Degradation-aware feature perturbation for all-in-one image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28165–28175, 2025.

[39] Jeya Maria Jose Valanarasu, Rajeev Yasarla, and Vishal M. Patel. Transweather: Transformer-based restoration of images degraded by adverse weather conditions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2353–2363, 2022.

[40] Cong Wang, Jinshan Pan, Wei Wang, Jiangxin Dong, Mengzhu Wang, Yakun Ju, and Junyang Chen. Promptrestorer: A prompting image restoration method with degradation perception. *Advances in Neural Information Processing Systems*, 36:8898–8912, 2023.

[41] Tao Wang, Kaihao Zhang, Ziqian Shao, Wenhan Luo, Bjorn Stenger, Tong Lu, Tae-Kyun Kim, Wei Liu, and Hongdong Li. Gridformer: Residual dense transformer with grid structure for image restoration in adverse weather conditions. *International journal of computer vision*, 132 (10):4541–4563, 2024.

[42] Yan Wang, Yi Liu, Shijie Zhao, Junlin Li, and Li Zhang. Camixersr: Only details need more" attention". In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25837–25846, 2024.

[43] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17683–17693, June 2022.

[44] Zhongze Wang, Haitao Zhao, Lujian Yao, Jingchao Peng, and Kaijie Zhao. Dual-level prototype learning for composite degraded image restoration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14006–14016, October 2025.

[45] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The thrity-seventh asilomar conference on signals, systems & computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

[46] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. *arXiv preprint arXiv:1808.04560*, 2018.

[47] Gang Wu, Junjun Jiang, Kui Jiang, and Xianming Liu. Harmony in diversity: Improving all-in-one image restoration via multi-task collaboration. In *Proceedings of the 32nd ACM international conference on multimedia*, pages 6015–6023, 2024.

[48] Yimin Xu, Nanxi Gao, Zhongyun Shan, Fei Chao, and Rongrong Ji. Unified-width adaptive dynamic network for all-in-one image restoration. *arXiv preprint arXiv:2401.13221*, 2024.

[49] 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*, pages 1357–1366, 2017.

[50] Ke Yu, Xintao Wang, Chao Dong, Xiaoou Tang, and Chen Change Loy. Path-restore: Learning network path selection for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):7078–7092, 2021.

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

[52] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Cycleisp: Real image restoration via improved data synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2696–2705, 2020.

[53] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Learning enriched features for real image restoration and enhancement. In *Computer Vision – ECCV 2020*, pages 492–511. Springer, 2020.

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

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

[56] Huimin Zeng, Jiacheng Li, Ziqiang Zheng, and Zhiwei Xiong. All-in-one image compression and restoration. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pages 609–619, 2025.

[57] Jinghao Zhang, Jie Huang, Mingde Yao, Zizheng Yang, Hu Yu, Man Zhou, and Feng Zhao. Ingredient-oriented multi-degradation learning for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5825–5835, 2023.

[58] Jinxin Zhou, Tianyu Ding, Tianyi Chen, Jiachen Jiang, Ilya Zharkov, Zhihui Zhu, and Luming Liang. Dream: Diffusion rectification and estimation-adaptive models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8342–8351, 2024.

[59] Yurui Zhu, Tianyu Wang, Xueyang Fu, Xuanyu Yang, Xin Guo, Jifeng Dai, Yu Qiao, and Xiaowei Hu. Learning weather-general and weather-specific features for image restoration under multiple adverse weather conditions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21747–21758, 2023.

[60] Zhihui Zhu, Tianyu Ding, Jinxin Zhou, Xiao Li, Chong You, Jeremias Sulam, and Qing Qu. A geometric analysis of neural collapse with unconstrained features. *Advances in Neural Information Processing Systems*, 34:29820–29834, 2021.

A Detailed Architecture

This appendix provides implementation details that complement the main method in Section 3 and the overview in Figure 3. We fix base channel dimension $d = 48$ and $K = 4$ experts throughout; all configurations scale proportionally across the four encoder-decoder levels.

A.1 Encoder-Decoder Configuration

The U-Net has four levels with channel dimensions $[48, 96, 192, 384]$, attention head counts $[1, 2, 4, 8]$, and block counts $[2, 3, 3, 4]$. Each block alternates a channel mixer and a spatial mixer. Downsampling is performed by a 3×3 convolution followed by PixelUnshuffle(2); upsampling uses a 3×3 convolution followed by PixelShuffle(2). Skip connections from encoder to decoder are concatenated and fused with a 1×1 convolution.

A.2 Attention Mechanisms

Both the spatial mixer and channel mixer are built on attention, but they operate along different dimensions. This subsection expands the attention operators used by the two mixers introduced in Sections 3.1 and 3.2.

Window-based spatial attention. The spatial attention module operates within non-overlapping $q \times q$ windows. Given an input $\boldsymbol{X} \in \mathbb{R}^{N\times C}$ where $N = q^2$ is the number of tokens per window, queries, keys, and values are computed by a single linear projection and split across h attention heads with head dimension $d_h = C/h$:

$$\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} = \mathrm{split}(\boldsymbol{W}_{\mathrm{qkv}}\, \boldsymbol{X}), \qquad \boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{h\times N\times d_h}. \tag{4}$$

The attention matrix incorporates a *continuous relative position bias* (CPB) [23]. Rather than storing a fixed bias table, a small MLP ($2 \to 512 \to h$) maps normalized relative coordinates to per-head biases, allowing generalisation to unseen window sizes:

$$\mathrm{Attn} = \mathrm{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_h}} + \boldsymbol{B}_{\mathrm{CPB}}\right)\boldsymbol{V}, \tag{5}$$

where $\boldsymbol{B}_{\mathrm{CPB}} \in \mathbb{R}^{h\times N\times N}$ is the position bias output of the CPB-MLP indexed by pairwise relative positions. The output is projected back to C dimensions via a linear layer.

Each `SpatialExpert` wraps this attention module at a configurable hidden dimension d_k. When $d_k \neq C$, linear projections $\boldsymbol{W}_{\mathrm{in}} \in \mathbb{R}^{C\times d_k}$ and $\boldsymbol{W}_{\mathrm{out}} \in \mathbb{R}^{d_k\times C}$ map features into and out of the expert's subspace; when $d_k = C$ these projections are omitted.

Transposed channel attention. The channel attention module follows the multi-DConv head transposed attention (MDTA) design [55]. Unlike spatial attention which computes affinities between spatial tokens, MDTA transposes the attention to operate along the *channel* dimension. Given an input $\boldsymbol{X} \in \mathbb{R}^{C\times HW}$, queries, keys, and values are generated by a 1×1 convolution followed by a 3×3 depth-wise convolution:

$$\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} = \mathrm{split}(\mathrm{DWConv}_{3\times3}(\mathrm{Conv}_{1\times1}(\boldsymbol{X}))), \qquad \boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{h\times\frac{C}{h}\times HW}. \tag{6}$$

Queries and keys are ℓ_2-normalized along the spatial dimension, and a learnable temperature $\boldsymbol{\alpha} \in \mathbb{R}^h$ scales the attention logits:

$$\mathrm{Attn} = \mathrm{Softmax}\Big(\hat{\boldsymbol{Q}}\,\hat{\boldsymbol{K}}^{\mathrm{T}} \cdot \boldsymbol{\alpha}\Big)\boldsymbol{V}, \qquad \hat{\boldsymbol{Q}} = \frac{\boldsymbol{Q}}{\|\boldsymbol{Q}\|_2}, \;\; \hat{\boldsymbol{K}} = \frac{\boldsymbol{K}}{\|\boldsymbol{K}\|_2}, \tag{7}$$

where the matrix product $\hat{\boldsymbol{Q}}\,\hat{\boldsymbol{K}}^{\mathrm{T}} \in \mathbb{R}^{h\times\frac{C}{h}\times\frac{C}{h}}$ captures cross-channel correlations. The output is reshaped to spatial layout and projected by a 1×1 convolution. Because the attention dimension is C/h rather than HW, this module has complexity $\mathcal{O}(C^2HW/h)$ instead of the $\mathcal{O}(CH^2W^2)$ of standard spatial attention, making it efficient for high-resolution features.

Each `ChannelExpert` wraps this MDTA module at a configurable hidden dimension d_k with the same projection strategy as the spatial experts.

Feed-forward network. Both mixer blocks share the same gated depth-wise feed-forward network (GDFN) [55]. A 1×1 convolution expands the channel dimension by a factor r into two branches, a 3×3 depth-wise convolution enriches local context, and a gating mechanism modulates the output:

$$\mathrm{GDFN}(\boldsymbol{X}) = \boldsymbol{W}_{\mathrm{out}}(\mathrm{GELU}(\boldsymbol{x}_1) \odot \boldsymbol{x}_2)\,, \qquad [\boldsymbol{x}_1, \boldsymbol{x}_2] = \mathrm{DWConv}_{3\times3}(\boldsymbol{W}_{\mathrm{in}}\,\boldsymbol{X}), \tag{8}$$

where $\boldsymbol{W}_{\mathrm{in}} \in \mathbb{R}^{C\times 2rC}$, $\boldsymbol{W}_{\mathrm{out}} \in \mathbb{R}^{rC\times C}$, and $\odot$ denotes element-wise multiplication. We use expansion ratio $r = 2$ throughout.

A.3 Spatial Mixer Details

Content feature module. This paragraph specifies the content feature module used by the content router in Section 3.1. The global content descriptor $\boldsymbol{z}_{\mathrm{global}}$ is extracted from the first-level encoder feature rather than from the deeper feature being routed. Let $\boldsymbol{F}_1 \in \mathbb{R}^{B\times d\times H\times W}$ be the feature after the overlapped patch embedding. We apply a lightweight convolutional predictor

$$\boldsymbol{G} = \sigma\Big(\mathrm{Conv}_{3\times3}^{8\to2}\big(\sigma(\mathrm{Conv}_{1\times1}^{d\to8}(\boldsymbol{F}_1))\big)\Big) \in \mathbb{R}^{B\times2\times H\times W}, \tag{9}$$

where σ is LeakyReLU with slope 0.1. At each encoder, bottleneck, decoder, and refinement stage, $\boldsymbol{G}$ is bilinearly resized to the current feature resolution and partitioned into the same non-overlapping $q \times q$ windows as the residual stream. This gives a per-token content cue $\boldsymbol{G}_w \in \mathbb{R}^{q^2\times2}$ for every routed window, preserving low-level structural information that may be attenuated in deeper layers.

Content router. This paragraph gives the implementation corresponding to the router input triplet and logits in Equation (2). For a current feature map $\boldsymbol{X} \in \mathbb{R}^{B\times C\times H\times W}$, the spatial mixer first applies LayerNorm and reshapes each $q \times q$ window into token form $\boldsymbol{X}_w \in \mathbb{R}^{q^2\times C}$. The router also uses a fixed two-dimensional coordinate grid $\boldsymbol{P}_w \in [-1,1]^{q^2\times2}$ inside each window. For every window, the implementation forms the same three inputs used in the main text by averaging the corresponding token-level signals:

$$\begin{aligned}
\boldsymbol{z}_{\mathrm{local}} &= \frac{1}{q^2}\sum_{n=1}^{q^2} \boldsymbol{X}_w(n) \in \mathbb{R}^{C_\ell},\\
\boldsymbol{z}_{\mathrm{global}} &= \frac{1}{q^2}\sum_{n=1}^{q^2} \boldsymbol{G}_w(n) \in \mathbb{R}^{C_g},\\
\boldsymbol{z}_{\mathrm{pos}} &= \frac{1}{q^2}\sum_{n=1}^{q^2} \boldsymbol{P}_w(n) \in \mathbb{R}^{2}.
\end{aligned} \tag{10}$$

where $C_\ell = C$ and $C_g = 2$ in our implementation. A two-layer router then predicts the expert logits using the same notation as Equation (2):

$$\boldsymbol{l} = \boldsymbol{W}_2\,\mathrm{GELU}(\boldsymbol{W}_1\,\mathrm{LN}([\boldsymbol{z}_{\mathrm{local}}, \boldsymbol{z}_{\mathrm{global}}, \boldsymbol{z}_{\mathrm{pos}}])) \in \mathbb{R}^K, \tag{11}$$

with hidden width $\max(8, (C_\ell + C_g + 2)/4)$ in our implementation. During training, hard Gumbel-Softmax produces a one-hot expert assignment while keeping the logits differentiable. At inference, the router applies softmax followed by hard arg $\max_k l_k$, groups windows by selected expert, processes only the assigned windows with that expert, and scatters the outputs back to the original window order. The patch load-balancing loss follows the Switch formulation, using the mean hard assignment as load and the mean softmax probability as importance.

A.4 Channel Mixer Details

Task feature module. This paragraph details the task feature module described in Section 3.2; the resulting feature $\boldsymbol{g}$ is the representation visualized in Figure 4. The global task feature $\boldsymbol{g}$ is extracted once from the bottleneck feature map $\boldsymbol{Z}_{\mathrm{btl}} \in \mathbb{R}^{B\times C_{\mathrm{btl}}\times H'\times W'}$ and is reused by all routed channel mixers in the decoder and refinement stages. The module has three complementary branches:

1. **High-frequency branch.** A frozen depth-wise 3×3 Laplacian filter with kernel $[-1,-1,-1;\ -1,8,-1;\ -1,-1,-1]$ extracts edge and texture responses, followed by GELU, spatial mean pooling, and a two-layer MLP $C_{\mathrm{btl}} \to 2C_{\mathrm{btl}} \to C_{\mathrm{btl}}$.

2. **Low-frequency branch.** A 3×3 average pool with stride 1 and padding 1 smooths the bottleneck feature, followed by a 1×1 convolution, GELU, and spatial mean pooling.
3. **Statistics branch.** Channel-wise mean and standard deviation are concatenated into a $2C_{\mathrm{btl}}$-dimensional vector and passed through a two-layer MLP $2C_{\mathrm{btl}} \to 2C_{\mathrm{btl}} \to C_{\mathrm{btl}}$.

The three branch outputs are concatenated and fused as

$$\boldsymbol{g} = \boldsymbol{W}_g^{(2)}\,\mathrm{GELU}\Big(\boldsymbol{W}_g^{(1)}\,\mathrm{LN}([\boldsymbol{z}_{\mathrm{high}}, \boldsymbol{z}_{\mathrm{low}}, \boldsymbol{z}_{\mathrm{stat}}])\Big) \in \mathbb{R}^{C_{\mathrm{btl}}}, \tag{12}$$

where $\boldsymbol{W}_g^{(1)} : 3C_{\mathrm{btl}} \to 2C_{\mathrm{btl}}$ and $\boldsymbol{W}_g^{(2)} : 2C_{\mathrm{btl}} \to C_{\mathrm{btl}}$.

Multi-hot classification head and task router. This paragraph expands the multi-hot task supervision and image-level channel routing introduced in Section 3.2. A linear classification head maps $\boldsymbol{g}$ to M degradation logits:

$$\hat{\boldsymbol{y}} = \mathrm{Sigmoid}(\boldsymbol{W}_{\mathrm{cls}}\boldsymbol{g}) \in [0,1]^M, \tag{13}$$

and the head is supervised by binary cross-entropy with the multi-hot degradation label. In our CDD-11 setting, $M = 4$ for rain, snow, haze, and low-light bases. For channel routing, each routed decoder or refinement block concatenates $\boldsymbol{g}$ with the global-average-pooled current feature map and feeds the result to the same LayerNorm $\to$ Linear $\to$ GELU $\to$ Linear router used by the spatial mixer. Training uses hard Gumbel-Softmax and the corresponding image-level load-balancing loss; inference uses hard arg max and routes each image to a single channel expert.

A.5 Training Losses

MS-SSIM loss. This paragraph expands the structural term $\mathcal{L}_{\mathrm{ms}}$ in the total objective Equation (3). Given the restored image $\hat{I}$ and clean target J, we use the differentiable multi-scale SSIM loss

$$\mathcal{L}_{\mathrm{ms}}(\hat{I}, J) = 1 - \prod_{s=1}^{S-1} \mathrm{CS}_s(\hat{I}, J)^{\alpha_s} \cdot \mathrm{SSIM}_S(\hat{I}, J)^{\alpha_S}, \tag{14}$$

where $S = 5$ and $\boldsymbol{\alpha} = [0.0448,\ 0.2856,\ 0.3001,\ 0.2363,\ 0.1333]$. At each scale, SSIM is computed with an 11×11 Gaussian window:

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \qquad \mathrm{CS}(x,y) = \frac{2\sigma_{xy} + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}. \tag{15}$$

Images are downsampled by $2\times$ average pooling between scales, matching the implementation.

Composite-degradation contrastive loss. This paragraph gives the implementation of $\mathcal{L}_{\mathrm{cdrl}}$ used in Section 3.3. Let I be the degraded input, J the clean positive target, $\hat{I}$ the restored output, and $\{I_o\}_{o=1}^{O}$ the other degraded negatives. For a perceptual feature extractor V, we use the VGG-16 feature layers $\mathcal{K} = \{3, 8, 15\}$ when available; otherwise the same expression falls back to pixel space. The contrastive loss is

$$\mathcal{L}_{\mathrm{cdrl}} = \frac{1}{|\mathcal{K}|} \sum_{k\in\mathcal{K}} \frac{\|V_k(\hat{I}) - V_k(J)\|_1}{\frac{1}{O+1}\left(\|V_k(\hat{I}) - V_k(I)\|_1 + \sum_{o=1}^{O} \|V_k(\hat{I}) - V_k(I_o)\|_1\right) + \epsilon}, \tag{16}$$

with $\epsilon = 10^{-7}$. This pulls the restoration toward the clean target while keeping it separated from the input degradation and the other degraded negatives.

Load-balancing loss. This paragraph expands the load-balancing term $\mathcal{L}_{\mathrm{lb}}$ used by both routers in Sections 3.1 and 3.2 and summarized in Section 3.3. For each routed layer r, let $p_{i,k}^{(r)}$ be the softmax probability that routed item i selects expert k, and let $a_{i,k}^{(r)}$ be the corresponding hard one-hot assignment from Gumbel-Softmax during training. The per-layer Switch-style loss is

$$\mathcal{L}_{\mathrm{lb}}^{(r)} = K \sum_{k=1}^{K} \underbrace{\frac{1}{N_r}\sum_{i=1}^{N_r} a_{i,k}^{(r)}}_{\mathrm{load}_k^{(r)}} \underbrace{\frac{1}{N_r}\sum_{i=1}^{N_r} p_{i,k}^{(r)}}_{\mathrm{importance}_k^{(r)}}, \tag{17}$$

where N_r is the number of routed windows for a spatial mixer or the batch size for a channel mixer. We average this loss over routed spatial stages and routed channel stages, then average the two available groups to obtain $\mathcal{L}_{\mathrm{lb}}$ used in Equation (3).

B Additional Routing and Representation Analyses

This appendix details the inference-time interventions summarized in Section 4.3, the decodability tests referenced in Section 3.2, and the zero-shot evaluation on real-world images. All interventions keep the trained weights and the test images fixed and modify only routing decisions; results are on the CDD-11 test set (n=1,100).

B.1 Controlled spatial-routing intervention

The correlation between patch difficulty and selected capacity (Figure 6a) could arise either because the model uses a favorable overall mixture of expert capacities or because it assigns the right patches to the right experts. To separate the two effects, for every image and every spatial-mixer block we keep the expert-usage histogram of the learned router fixed, so that each expert processes exactly the same number of patches and the total FLOPs are unchanged, and vary only which patches are assigned to each capacity level:

- **Learned**: the assignments predicted by the trained spatial router;
- **Random**: the learned expert labels randomly permuted across patches;
- **Reverse-difficulty**: patches with larger input-to-ground-truth log MSE are assigned to smaller experts, and easier patches to larger ones.

Learned routing outperforms random routing by 0.587 dB at identical FLOPs (Table 5); since both activate each expert equally often, the gain cannot be attributed to using more high-capacity experts. Reverse-difficulty routing is a further 0.072 dB below random routing, so the assignment also has a meaningful direction.

Table 5: Controlled spatial-routing intervention on CDD-11. All variants share the per-block expert-usage histogram of the learned router, and hence its FLOPs.

Routing	PSNR (dB)	Δ vs. learned
Learned	**29.950**	–
Random, histogram-matched	29.363	-0.587
Reverse-difficulty	29.291	-0.659

B.2 Channel-expert usage and forced pathways

Figure 6b shows, for each degradation type and decoder block, only the most frequently selected expert; an expert absent from this arg-max path may still be selected often. Aggregated over all channel-mixer blocks and test images, the four channel experts receive 18%, 17%, 29%, and 35% of the assignments, so no expert is globally inactive. As in standard sparse MoE models, all experts are stored but only the selected one is executed; experts that are not selected for an input add parameters but no FLOPs or latency (Section 4.4).

To test whether the channel experts are interchangeable, we force every channel-mixer block to use the same expert and evaluate each fixed pathway (Table 6). Learned routing outperforms every fixed pathway, by 0.80–3.49 dB on average, and the effect depends strongly on the degradation: forcing E_4 remains competitive on low light (-0.23 dB) but loses 8.45, 5.29, and 10.62 dB on haze, rain, and snow, where E_1 and E_2 are substantially more effective. Different pathways therefore implement functionally different transformations, and the router benefits from selecting and combining them across decoder blocks. Because the experts also differ in capacity and co-adapt across decoder stages, we do not assign a single semantic degradation label to each expert.

B.3 Forced spatial experts

As a diagnostic of the capacity levels, we force every spatial patch through a single expert E_1, E_2, E_3, or E_4, which yields 29.05, 29.20, 29.24, and 29.40 dB on CDD-11, respectively (learned routing: 29.95 dB). Performance increases consistently with expert capacity, and removing the highest-capacity expert causes a 0.19 dB drop, larger than removing any lower-capacity expert. The

Table 6: Forcing all channel-mixer blocks to a single expert (PSNR in dB on CDD-11). Rows show the four single-degradation conditions and the average over all 11 conditions.

Degradation	Learned routing	Forced E_1	Forced E_2	Forced E_3	Forced E_4
Low light	**27.05**	26.09	26.34	26.79	26.82
Haze	**34.52**	33.60	33.59	31.70	26.07
Rain	**35.46**	35.02	34.99	33.00	30.17
Snow	**37.54**	35.86	35.81	35.73	26.92
Average (11 conditions)	**29.95**	29.08	29.15	28.70	26.46

four capacity levels are thus ordered and functionally distinguishable rather than interchangeable. These diagnostics use the trained $K{=}4$ model and do not replace retraining with a different K.

B.4 Compositional decodability of the task feature

We examine whether elemental degradations remain identifiable in $\boldsymbol{g}$ when several degradations co-occur (Section 3.2), using two tests on the frozen representation. (i) *Directional consistency.* For each valid pair, we compare the feature change induced by adding degradation A to condition B, $\boldsymbol{g}(A{+}B) - \boldsymbol{g}(B)$, with the corresponding single-degradation direction $\boldsymbol{g}(A) - \boldsymbol{g}(\mathrm{clear})$; their mean cosine similarity is 0.781, so adding the same degradation moves the representation in a consistent direction across contexts. (ii) *Additive reconstruction.* Predicting each composite class mean from the directions of its elemental constituents explains 77.8% of the variance across composite class means, so a substantial part of the composite representation can be recovered from its elemental components. These results indicate that elemental degradations act as consistent, approximately additive directions in the learned task space, which supports compositional decodability; they do not establish an orthogonal basis, which we do not claim.

B.5 Zero-shot evaluation on real-world images

We evaluate the CDD-11-trained CART and MoCE-IR [51] directly on WeatherBench [13], a real-world adverse-weather restoration benchmark with paired degraded and clean images, without retraining or fine-tuning (Table 7). CART outperforms MoCE-IR on all three metrics, but the absolute performance remains limited. The methods reported by the WeatherBench benchmark are trained on its paired real-world data and are therefore not directly comparable to this zero-shot setting. Closing this synthetic-to-real gap would require additional training or fine-tuning on real degradations, which we leave to future work.

Table 7: Zero-shot synthetic-to-real transfer on WeatherBench. Both models are trained on CDD-11 only.

Method	PSNR↑	SSIM↑	LPIPS↓
MoCE-IR [51] (zero-shot)	19.961	0.7271	0.3902
CART (zero-shot)	**21.658**	**0.7347**	**0.3759**

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the composite image restoration setting, the CART architecture, and the empirical claims later evaluated in Sections 3 and 4.2.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Section 6 discusses the scope of the evaluation (composite restoration over a known set of elemental degradations, without out-of-distribution tests) and the synthetic-to-real gap observed in zero-shot WeatherBench results; Section 5 lists future directions such as real-world evaluation, efficient implementation, and video restoration.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper proposes and evaluates an empirical image restoration method and does not present theoretical results or formal proofs.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: Sections 3, 4, and A describe the architecture, losses, datasets, training schedule, and evaluation metrics needed to reproduce the reported experiments.

Guidelines:

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

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

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

   Answer: [Yes]

   Justification: We will release the source code, trained checkpoints, and restoration results with instructions to reproduce the main experiments.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 4 specifies the datasets, task splits, crop size, optimizer, learning rate schedule, batch size, training epochs, routing temperature schedule, loss weights, and evaluation metrics.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Section 4.3 analyzes routing behavior across samples and reports distributional evidence for the significance of the learned routing patterns.

   Guidelines:

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

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Section 4.4 provides a complexity analysis table reporting parameters, FLOPs, memory, and runtime for the evaluated methods.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research uses standard image restoration benchmarks and does not involve human subjects, private data, or other procedures that would conflict with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: The paper studies a foundational image restoration method and does not make direct deployment claims or introduce a specific societal application.

Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release high-risk generative models, pre-trained language models, or scraped datasets requiring special misuse safeguards.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The experiments use open-source models and publicly available datasets, including the real-world WeatherBench benchmark used for zero-shot evaluation, and the corresponding creators are credited through citations in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

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

13. **New assets**

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

    Answer: [N/A]

    Justification: The paper does not introduce or release a new dataset, model checkpoint, benchmark, or other standalone asset.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The research does not involve crowdsourcing experiments or human-subject studies.

    Guidelines:

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

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

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

    Answer: [N/A]

Justification: The research does not involve crowdsourcing or human-subject studies, so IRB or equivalent approval is not applicable.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLMs are not an important, original, or non-standard component of the proposed method or experimental evaluation.

Guidelines:

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