ALLNet: Multi-task dense prediction for degraded images

Supplementary Material

In this supplementary document, we propose: (1) a more comprehensive explanation of experimental implementation,(2) more details about motivation, and (3) more quantitative and qualitative experimental results.

6. Implementation Details

6.1. Dataset

NYUD-v2 [66] contains 795 RGB-D indoor images for training and 654 for testing. In our experiments, we use the 13-category semantic annotations provided in [75], together with the ground-truth depth maps captured by a Microsoft Kinect sensor and the released surface normal annotations. All images are uniformly resized to 288×384 in our implementation to accelerate training.

PASCAL-Context[67] is an extension of PASCAL VOC that provides dense semantic annotations for most visible pixels in each image, covering a wide range of object and scene categories, and is a standard benchmark for evaluating scene understanding and contextual modeling.In this work, we use the dataset's pixel-level annotations for semantic segmentation, human part segmentation, and semantic edge detection, and further incorporate additional curated ground truth for surface normal estimation and saliency detection to form a multi-task dense prediction setting.

6.2. Metrics

To comprehensively evaluate model performance across different tasks, we employ a diverse set of quantitative metrics and perform rigorous experimental validation. The specific evaluation criteria are detailed below:

1. **mIoU:** Mean Intersection over Union
2. **rmse:** Root Mean Square Error (For surface normal estimation, we calculate the RMSE of normal angle)
3. **mErr:** Mean of Angle Error
4. **max-F:** Maximum of F_1-measure
5. **odsF:** Optimal Dataset Scale F-measure

To comprehensively evaluate the proposed method, we introduce a relative gain metric for each task. For task τ_j where $j = 1, \ldots, N$, the relative gain Δ_{τ_j} is defined as:

$$\Delta_{\tau_j} = (-1)^{l_j} \frac{M_{m,j} - M_{s,j}}{M_{s,j}}, \tag{11}$$

where $l_j = 1$ indicates that lower values are better for performance measure M_j of task j, and $l_j = 0$ indicates that higher values are better.

Additionally, we employ the multi-task performance metric Δ_{MTL} to assess mutual improvement across all tasks:

$$\Delta_{MTL} = \frac{1}{N} \sum_{j=1}^{N} \Delta_{\tau_j}. \tag{12}$$

6.3. Degradation Synthesis

To train ALLNet under diverse degradation conditions, we construct synthetic degraded versions of NYUD-v2 and PASCAL-Context by applying one degradation type to each clean RGB image. In all cases, a degraded observation $\tilde{I}$ is obtained from the clean image I via

$$\tilde{I} = \mathcal{D}(I; \boldsymbol{\theta}), \tag{13}$$

where $\mathcal{D}(\cdot)$ denotes the degradation operator and $\boldsymbol{\theta}$ are randomly sampled parameters controlling the degradation strength. We uniformly sample degradation types within each mini-batch and draw their parameters from type-specific ranges so that the training set covers a wide spectrum of degradation levels.

Gaussian noise. We adopt an additive white Gaussian noise model. For each image, the degraded observation is given by

$$\tilde{I}(x) = I(x) + n(x), \quad n(x) \sim \mathcal{N}(0, \sigma^2), \tag{14}$$

where x indexes pixels. After normalizing intensities to $[0, 1]$, the noise level is sampled as

$$\sigma \sim \mathcal{U}(\sigma_{\min}, \sigma_{\max}), \tag{15}$$

with $\sigma_{\min}$ and $\sigma_{\max}$ corresponding to light and heavy noise regimes (e.g., $5/255$ and $50/255$), respectively. Noise is added on-the-fly during training so that the same clean image can appear with different noise realizations.

Haze. To synthesize haze, we use the standard atmospheric scattering model:

$$I_{\text{haze}}(x) = J(x)\, t(x) + A\big(1 - t(x)\big), \tag{16}$$

where $J(x)$ is the clean scene radiance, A the global atmospheric light, and $t(x)$ the transmission map. The transmission is determined by the scene depth $d(x)$ via

$$t(x) = \exp\big(-\beta\, d(x)\big), \tag{17}$$

with β the scattering coefficient. We sample $A_c \sim \mathcal{U}(0.7, 1.0)$ for each color channel $c \in \{R, G, B\}$ and

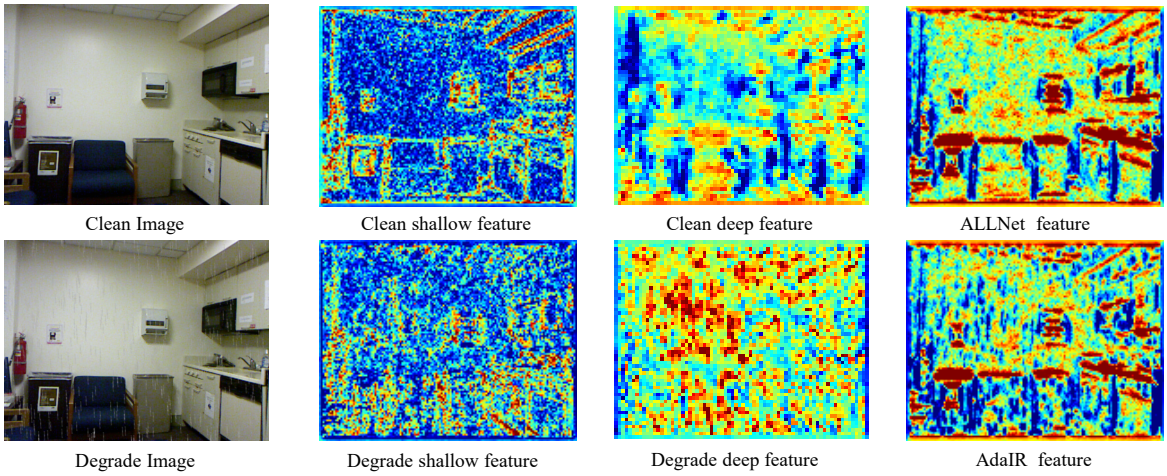

Figure 9. Degraded feature maps.

$\beta \sim \mathcal{U}(\beta_{\min}, \beta_{\max})$, and use ground-truth or predicted depth maps to approximate $d(x)$, generating haze from mild fog to dense conditions.

Rain. For rain streaks, we employ a region-dependent rain image model. The rainy image is expressed as

$$O(x) = B(x) + S(x)\, R(x), \tag{18}$$

where $B(x)$ is the clean background, $S(x)$ is the rain-streak intensity, and $R(x) \in \{0, 1\}$ is a sparse binary rain mask. Given a randomly sampled rain density, we generate $R(x)$ and convolve it with a bank of line kernels having random lengths, widths, and orientations to obtain $S(x)$. This yields diverse rain patterns ranging from sparse drizzle to heavy rain; optionally, a weak haze component can be added using the scattering model above to mimic accumulated distant rain.

Motion blur. For motion blur on static datasets, we follow the realistic blur modeling in RSBlur and synthesize blur directly in the image plane. Given a clean sRGB image I, we first map it to a linear radiance domain using an approximate camera response function (CRF),

$$L = g^{-1}(I), \quad g(x) = x^{1/\gamma},\ \gamma \approx 2.2. \tag{19}$$

We then sample a 2D motion trajectory $\{\mathbf{p}_i\}_{i=0}^{M-1}$ on the image plane and build a normalized point spread function (PSF)

$$k(\mathbf{u}) = \frac{1}{M} \sum_{i=0}^{M-1} \delta\big(\mathbf{u} - \mathbf{p}_i\big), \tag{20}$$

where the trajectory length, direction, and mild curvature are randomly drawn within predefined ranges, controlling blur strength and anisotropy. The blurred linear image is obtained by

$$\hat{L}(\mathbf{x}) = (k * L)(\mathbf{x}), \tag{21}$$

and mapped back to sRGB as

$$B(\mathbf{x}) = g\big(\hat{L}(\mathbf{x})\big). \tag{22}$$

Optionally, we inject Poisson–Gaussian noise in the linear domain consistent with our noise model and then apply the same CRF, yielding motion-blurred images whose spatial blur patterns and noise characteristics more closely resemble real-world observations.

7. Motivation

Based on the feature visualizations presented in Fig.9, we summarize two key observations under degraded conditions that directly motivate our method design. First, for clear images, the network effectively captures distinct hierarchical features: low-level characteristics such as edges and textures in shallow layers, and high-level semantic representations in deeper layers. In contrast, under adverse conditions like raindrop degradation, shallow features become corrupted and contaminated with noise, while deep features exhibit weakened semantic responses to objects, leading to reduced discriminative power. This performance gap highlights the critical need for integrated feature enhancement in degraded scenarios.

Furthermore, the comparison in the last column reveals that while conventional two-stage approaches accomplish the task to some extent, they suffer from noticeable limitations in preserving structural and semantic information. Specifically, features restored by AdaIR appear blurred at edges and lack structural coherence, whereas the intermediate recovered features extracted by our proposed ALLNet retain sharper edges and more complete semantic content. This contrast demonstrates that the sequential processing paradigm inevitably causes information fragmentation, whereas

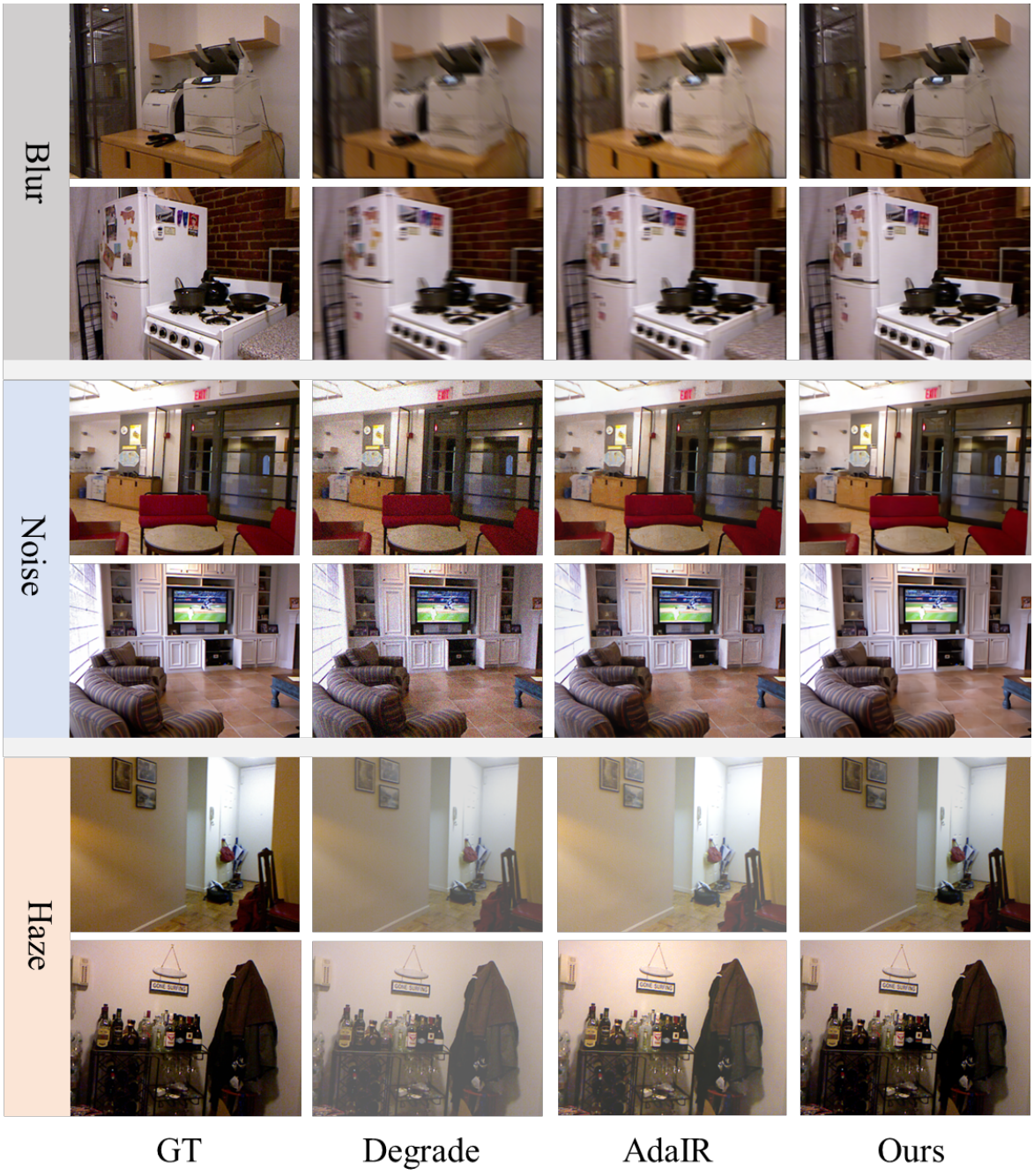

Figure 10. Visual comparisons between our method and state-of-the-art all-in-one methods, such as AdaIR[65].

our collaboratively optimized mechanism enables more effective joint recovery of both low-level details and high-level semantics. These observations collectively justify our design of an end-to-end architecture that simultaneously addresses degradation removal and feature enhancement, thereby maintaining robust semantic discriminability while restoring image fidelity.

8. Experimental Results

The proposed image restoration network is built upon the InternImage-S encoder for feature extraction, combined with an MAE decoder. Comparative experiments leveraging this network were conducted to evaluate its performance against other methods

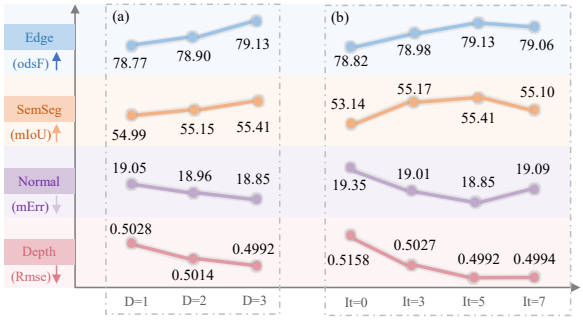

Figure 11. Hyper-parameter analysis of task-aware modules across four tasks on the degraded NYUD-v2 dataset.

8.1. Visual Results

Fig.10 presents the visual comparisons under degraded settings. In scenes with severe haze, AdaIR fails to completely remove the haze, leading to noticeable color disparities; in contrast, our method achieves more accurate color reconstruction. Furthermore, in challenging blur conditions, AdaIR still exhibits significant edge distortions, whereas our approach effectively mitigates such residual artifacts, distinguishing it from other alternatives. Additionally, our method produces deblurring and denoising results with clearer details and sharper edges, while introducing fewer visible artifacts. Combined with quantitative evaluations, these qualitative comparisons further demonstrate the superiority and robustness of our method. Moreover, the reconstructed results provide higher-quality feature information for subsequent dense prediction tasks.

Table 5. Results on different Methods on the degraded NYUD-v2 dataset.

Method	dehaze	denoise	deblur
ours	**31.10**	**30.22**	**28.22**
AdaIR[65]	27.28	28.22	26.14

Comparison to State-of-the-Art Methods. On the degraded NYUD-v2 dataset, we compare our all-in-one restoration approach with the specialized method AdaIR [65]. As demonstrated in the results, our model achieves superior restoration quality across all three tasks—dehazing, denoising, and deblurring—delivering robust and leading performance over AdaIR [65]. This indicates that the proposed unified framework exhibits stronger generalization and restoration capability in handling diverse degradation scenarios.

8.2. Hyperparameters

Different Decoder Stages. The MAE decoder employed in our paper consists of 12 layers. To investigate the impact of integrating different numbers of MAE decoder layers into our in-task module, we designed three experimental configurations: Stage 1 with [2, 2, 2, 4] layers, Stage 2 with [2, 2, 3, 4] layers, and Stage 3 with [2, 2, 4, 4] layers, respectively. As shown in Fig.11(a), increasing the number of decoder layers leads to a corresponding improvement in accuracy, with the best performance achieved when all stages are incorporated.

Iterations. To determine the optimal setting of the iteration parameter It that influences the convergence of ALLNet, we conducted a series of experiments across a range of Itvalues. As illustrated in Fig.11(b), the peak performance is achieved at $\mathbf{It} = 5$, which was consequently selected as the balanced trade-off between accuracy and complexity in subsequent experiments.