HierUQ: Hierarchical Uncertainty Quantification with Adaptive Granularity Reconciliation for Degraded Image Classification

Supplementary Material

1. A: Uncertainty Quantification and Confidence Adjustment

1.1. A1: ViT-GNN Feature Extractor Implementation

Global Feature Encoding via ViT. We adopt ViT-B/16 as the visual backbone to extract global features from degraded images. Given an input image $\mathbf{X} \in \mathbb{R}^{448\times448\times3}$, it is partitioned into $N = 784$ non-overlapping patches of size 16×16, each embedded as:

$$\mathbf{x}_p^{(i)} = \text{Flatten}(\mathbf{X}_{p_i}) \cdot \mathbf{E}, \quad \mathbf{E} \in \mathbb{R}^{768\times768}, \tag{1}$$

where $\mathbf{X}_{p_i}$ is the i-th image patch and $\mathbf{E}$ is a learnable embedding matrix. A classification token $\mathbf{x}_{\text{cls}}$ and positional encoding $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{(N+1)\times768}$ are appended:

$$\mathbf{z}_0 = [\mathbf{x}_{\text{cls}}; \mathbf{x}_p^{(1)}; \ldots; \mathbf{x}_p^{(N)}] + \mathbf{E}_{\text{pos}}. \tag{2}$$

The sequence is processed by $L = 12$ Transformer layers:

$$\mathbf{z}_\ell = \text{MLP}(\text{LN}(\text{MSA}(\text{LN}(\mathbf{z}_{\ell-1}))) + \mathbf{z}_{\ell-1}), \tag{3}$$

yielding final output $\mathbf{Z}_L = [\mathbf{z}_{\text{cls}}; \mathbf{z}_1; \ldots; \mathbf{z}_N] \in \mathbb{R}^{(N+1)\times768}$, where $\mathbf{z}_{\text{cls}}$ captures global semantics and $\mathbf{z}_i$ denotes the i-th patch token.

Semantic-Visual Fusion via Bilinear Attention. Patch tokens $\mathbf{F}_{\text{patch}} = [\mathbf{z}_1; \ldots; \mathbf{z}_N] \in \mathbb{R}^{N\times768}$ are fused with class semantics via bilinear attention. Class semantics are represented by pretrained GloVe embeddings $\mathbf{W}_{\text{semantic}} \in \mathbb{R}^{C\times300}$, where $C = 251$ is the number of categories. Both modalities are projected into a shared space:$\mathbf{F}_{\text{vis}} = \mathbf{F}_{\text{patch}}\mathbf{W}_{\text{vis}}, \quad \mathbf{F}_{\text{sem}} = \mathbf{W}_{\text{semantic}}\mathbf{W}_{\text{sem}}$, with $\mathbf{W}_{\text{vis}} \in \mathbb{R}^{768\times512}$ and $\mathbf{W}_{\text{sem}} \in \mathbb{R}^{300\times512}$.

Bilinear interaction is modeled as:

$$\mathbf{H}_{\text{inter}} = \tanh(\mathbf{F}_{\text{vis}} \odot \mathbf{F}_{\text{sem}}), \tag{4}$$

where $\odot$ denotes element-wise multiplication after broadcasting. Here, broadcasting is performed along the semantic class dimension to align the visual features $\mathbf{F}_{\text{vis}} \in \mathbb{R}^{N\times d}$ with the semantic features $\mathbf{F}_{\text{sem}} \in \mathbb{R}^{C\times d}$, yielding a bilinear interaction tensor $\mathbf{H}_{\text{inter}} \in \mathbb{R}^{N\times C\times d}$ for all N patches and C semantic classes. Specifically, $\mathbf{F}_{\text{vis}} \in \mathbb{R}^{N\times d}$ and $\mathbf{F}_{\text{sem}} \in \mathbb{R}^{C\times d}$ are broadcast to $\mathbb{R}^{N\times C\times d}$ to enable interaction between each visual patch and each semantic class embedding along the shared feature dimension. Attention weights are computed by:

$$\alpha_{i,j} = \text{softmax}(\mathbf{W}_a^\top \mathbf{W}_h \mathbf{h}_{i,j} + b_a), \tag{5}$$

with $\mathbf{W}_h \in \mathbb{R}^{512\times512}$ and $\mathbf{W}_a \in \mathbb{R}^{512\times1}$. The final fused feature for class j is:

$$\mathbf{G}_j = \sum_{i=1}^{N} \alpha_{i,j} \cdot \mathbf{z}_i, \quad \mathbf{G} \in \mathbb{R}^{C\times768}. \tag{6}$$

SGCA and Hierarchical Encoding via GatedGNN. To capture inter-modal dependency, we adopt Semantic-Guided Cross-Attention (SGCA), with visual query $\mathbf{Q}_{\text{vis}} = \mathbf{G}\mathbf{W}^Q$, semantic keys $\mathbf{K}_{\text{sem}} = \mathbf{W}_{\text{semantic}}\mathbf{W}^K$, and values $\mathbf{V}_{\text{sem}} = \mathbf{W}_{\text{semantic}}\mathbf{W}^V$, where all projection matrices are of appropriate dimensions. Cross-attention is computed as:

$$\text{SGCA} = \text{softmax}\left(\frac{\mathbf{Q}_{\text{vis}}\mathbf{K}_{\text{sem}}^\top}{\sqrt{768}}\right) \cdot \mathbf{V}_{\text{sem}}. \tag{7}$$

All projection matrices map to dimension $\mathbb{R}^{768\times d_a}$ with $d_a = 768$, ensuring the attention mechanism operates over a consistent hidden size.

Hierarchical structure is further encoded via a 3-layer GatedGNN. The class hierarchy is encoded in adjacency matrix $\mathbf{A} \in \mathbb{R}^{C\times C}$:

$$\mathbf{A}_{i,j} = \begin{cases} 1, & \text{if } i \text{ and } j \text{ are parent-child classes,} \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Each layer updates node states $\mathbf{H}^{(\ell)}$ as:

$$\mathbf{H}^{(\ell+1)} = \text{ReLU}(\mathbf{U}^{(\ell)}\mathbf{H}^{(\ell)} + \text{GRU}^{(\ell)}(\mathbf{A}, \mathbf{H}^{(\ell)}) + \mathbf{b}_u^{(\ell)}), \tag{9}$$

where the gated message passing unit is:

$$\begin{aligned} \text{GRU}^{(\ell)} = \mathbf{A}^\top \Big(&\sigma(\mathbf{A}\mathbf{V}_j^{(\ell)}\mathbf{H}^{(\ell)} + \mathbf{A}\mathbf{V}_i^{(\ell)}\mathbf{H}^{(\ell)} \\ &+ \mathbf{b}_v^{(\ell)}) \odot (\mathbf{A}\mathbf{U}_j^{(\ell)}\mathbf{H}^{(\ell)})\Big) \end{aligned} \tag{10}$$

1.2. A2: Hierarchical Consistency Constraints and Confidence Modeling

Hierarchical Consistency Constraints. We represent the hierarchy as $\mathcal{H} = \{L_1, \ldots, L_K\}$, where L_k denotes label sets at level k. Three structural principles are considered: (i) hierarchical inclusion: $c_{k+1} \subseteq c_k$; (ii) probability consistency:

$$P^{(k)}(c_k \mid \mathbf{x}) = \sum_{c_{k+1}\in\text{Ch}(c_k)} P^{(k+1)}(c_{k+1} \mid \mathbf{x}), \tag{11}$$

where $P^{(k)}(c_k \mid \mathbf{x})$ is the predicted probability at level k and $\mathrm{Ch}(c_k)$ denotes its children; (iii) parent-conditioned factorization approximation: $P(Y^{(k+1)} \mid Y^{(1:k)}, \mathbf{x}) \approx P(Y^{(k+1)} \mid Y^{(k)}, \mathbf{x})$. To enforce the structural constraints in (i) and (ii), we apply a KL-based constraint loss:

$$\mathcal{L}_{\text{constraint}} = \sum_{k=1}^{K-1} \lambda_k \cdot \mathrm{KL}(P^{(k)} \| \mathrm{Marg}(P^{(k+1)})), \quad (12)$$

where λ_k is either set manually or learned dynamically to reflect the relative importance of consistency at level k, and $\mathrm{Marg}(\cdot)$ denotes hierarchical marginalization using child-to-parent mapping [2]. Specifically, for a given class $c_k \in L_k$, the marginalized probability is defined as:

$$\mathrm{Marg}(P^{(k+1)})(c_k) = \sum_{c_{k+1} \in \mathrm{Ch}(c_k)} P^{(k+1)}(c_{k+1}). \quad (13)$$

Confidence Calibration. We adopt Brier and log scores for hierarchical calibration:

$$\mathrm{BS}_{\text{hier}} = \sum_{k=1}^{K} w_k \sum_{i=1}^{C_k} \left(p_i^{(k)} - y_i^{(k)} \right)^2, \quad (14)$$

$$\mathrm{LS}_{\text{hier}} = -\sum_{k=1}^{K} w_k \log p_{y^{(k)}}^{(k)} - \mu \sum_{k=1}^{K-1} \log \sum_{j \in \mathrm{Ch}(y^{(k)})} p_j^{(k+1)}. \quad (15)$$

where C_k is the number of classes at level k, $p_i^{(k)}$ and $y_i^{(k)}$ are predicted and ground-truth indicators, respectively, and μ is a fine-level semantic consistency factor that balances the contribution of child-level predictions to the overall hierarchical likelihood.

We further define hierarchical calibration error as:

$$\mathrm{HCC} = \frac{1}{K} \sum_{k=1}^{K} |\mathrm{Acc}^{(k)} - \mathrm{Conf}^{(k)}|. \quad (16)$$

1.3. A3: Dynamic Confidence Adjustment and Calibration

Confidence Estimation and Uncertainty. Each level k employs a confidence subnetwork $\mathcal{N}_k^{\text{conf}}(\mathbf{h})$ mapping input $\mathbf{h} \in \mathbb{R}^d$ to $[0, 1]$:

$$\mathcal{N}_k^{\text{conf}}(\mathbf{h}) = \sigma(\mathbf{W}_k^{(2)} \cdot \mathrm{ReLU}(\mathbf{W}_k^{(1)} \mathbf{h} + \mathbf{b}_k^{(1)}) + \mathbf{b}_k^{(2)}), \quad (17)$$

with learnable weights $\mathbf{W}_k^{(1)} \in \mathbb{R}^{128 \times d}$, $\mathbf{W}_k^{(2)} \in \mathbb{R}^{1 \times 128}$. Dropout is inserted between the two linear layers in $\mathcal{N}_k^{\text{conf}}$ to facilitate uncertainty quantification via Monte Carlo sampling.

Uncertainty is estimated via Monte Carlo dropout:

$$\mathcal{U}_{\text{var}}^{(k)}(\mathbf{h}) = \frac{1}{T} \sum_{t=1}^{T} \left[\mathcal{N}_k^{\text{conf}}(\mathbf{h}; \theta_t) - \bar{\mathcal{N}}_k^{\text{conf}}(\mathbf{h}) \right]^2, \quad (18)$$

where θ_t is a sampled network and $\bar{\mathcal{N}}$ the mean prediction [3].

Bayesian Calibration. We compute calibrated confidence as:

$$\mathcal{C}_{\text{Bayes}}^{(k)} = \frac{c_k p_{\text{cor}}^{(k)}}{c_k p_{\text{cor}}^{(k)} + (1 - c_k) p_{\text{incor}}^{(k)}}, \quad (19)$$

where c_k is raw confidence, $p_{\text{cor}}^{(k)}$ and $p_{\text{incor}}^{(k)}$ are estimated correct/incorrect likelihoods. Here, $p_{\text{cor}}^{(k)}$ and $p_{\text{incor}}^{(k)}$ are estimated from held-out validation predictions using confidence binning and empirical frequency estimation.

Consistency and Structure Loss. To enforce inter-level consistency, we define a loss that penalizes discrepancies between the probability of a coarse-level class and the sum of its corresponding fine-level predictions. Specifically, for each level k, the term $\mathcal{L}_{\text{prob-cons}}$ minimizes the squared difference between $P^{(k)}(c_k|\mathbf{x})$ and the aggregated probability $\sum_{c_{k+1} \in \mathrm{Ch}(c_k)} P^{(k+1)}(c_{k+1}|\mathbf{x})$, weighted by a coefficient α_k.

Confidence-probability alignment:

$$\mathcal{L}_{\text{conf-cons}} = \sum_k \gamma_k \left| \mathcal{C}_{\text{Bayes}}^{(k)} - \max_{c \in \mathcal{C}_k} P^{(k)}(c|\mathbf{x}) \right|. \quad (20)$$

Logical violations penalized as:

$$\mathcal{L}_{\text{str-c}} = \sum_i \sum_{k=1}^{K-1} \delta_k \cdot \mathbb{I} \left[\hat{c}_i^{(k)} \notin \mathrm{Anc} \left(\hat{c}_i^{(k+1)} \right) \right], \quad (21)$$

Here, $\mathrm{Anc}(\cdot)$ denotes the ancestor set in the class hierarchy, enforcing that predicted fine-grained classes remain descendants of their coarse-level predictions [5].

Fusion and Calibration. We introduce a confidence-guided fusion mechanism:

$$\mathcal{W}_{\text{adaptive}}(\mathbf{h}, \mathbf{s}) = \mathrm{Softmax}(\mathrm{MLP}(\mathrm{MH\text{-}Attn}(\mathbf{W}_Q[\mathbf{h}; \mathbf{s}], \mathbf{K}, \mathbf{V}))), \quad (22)$$

Here, $\mathbf{K}$ and $\mathbf{V}$ are derived from semantic-guided representations, and MH-Attn denotes multi-head attention between joint features $[\mathbf{h}; \mathbf{s}]$ and semantic keys.

To further calibrate the model's confidence, we apply temperature scaling as a post-hoc adjustment:

$$\tilde{P}^{(k)}(c|\mathbf{x}) = \frac{\exp(z_c^{(k)}/T_k)}{\sum_j \exp(z_j^{(k)}/T_k)}, \quad (23)$$

$$T_k^* = \arg\min_{T_k} \left(-\sum_i \log \tilde{P}^{(k)}(y_i^{(k)}|\mathbf{x}_i) \right). \quad (24)$$

where the temperature parameter T_k is optimized on a held-out validation set via temperature scaling to improve confidence calibration.

2. B: Confidence-Aware Path Adjustment (CAPA)

Bidirectional Logical Reasoning. Top-down reasoning is formulated as conditional inference:

$$P_{\text{TD}}(c_{k+1}|c_k, \mathbf{x}) = \frac{\exp(\psi_{\text{TD}})}{\sum_{c'} \exp(\psi'_{\text{TD}})}, \tag{25}$$

$$\psi_{\text{TD}} = \mathbf{e}_{c_k}^\top \mathbf{W}_{\text{TD}} \mathbf{e}_{c_{k+1}} + \mathbf{h}^\top \mathbf{W}_{\text{feat}} (\mathbf{e}_{c_k} \otimes \mathbf{e}_{c_{k+1}}). \tag{26}$$

where $\mathbf{e}_{c_k}, \mathbf{e}_{c_{k+1}}$ are class embeddings, $\mathbf{h}$ is the feature vector.

Bottom-up reasoning uses learned relational encoding $\mathbf{r}_{c_{k+1} \to c_k}$:

$$P_{\text{BU}}(c_k|c_{k+1}, \mathbf{x}) = \frac{\exp(\text{MLP}_{\text{BU}}([\mathbf{h}; \mathbf{e}_{c_{k+1}}; \mathbf{r}_{c_{k+1} \to c_k}]))}{\sum_{c'} \exp(\cdot)}. \tag{27}$$

Here, $\mathbf{r}_{c_{k+1} \to c_k}$ denotes a learnable relation embedding that encodes semantic and structural transitions from fine to coarse levels, and is jointly optimized with classification loss.

Variational Alignment. We unify top-down and bottom-up inference using variational approximation, following the variational autoencoder formulation [4]. Specifically, we optimize the following evidence lower bound (ELBO):

$$\mathcal{L}_{\text{var}} = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{y}|\mathbf{z})] - \text{KL}(q(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})) \tag{28}$$

where $q(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mu, \sigma^2)$. The latent variable $\mathbf{z}$ serves as a task-adaptive reasoning state that captures hierarchical semantic representations with uncertainty.

Granular Feature Harmonization. To ensure inter-level consistency, we minimize the Wasserstein-2 distance between adjacent-level feature distributions:

$$\mathcal{L}_{\text{align}} = \sum_{k=1}^{K-1} \mathcal{W}_2 \left(\mathcal{F}^{(k)},\ \mathbf{W}^{(k)} \mathcal{F}^{(k+1)} + \mathbf{b}^{(k)} \right), \tag{29}$$

where $\mathcal{F}^{(k)}$ and $\mathcal{F}^{(k+1)}$ denote the feature representations at level k and $k+1$, and $\mathcal{W}_2$ represents the Wasserstein-2 distance [6].

Adaptive Fallback Policy. A fallback policy $\pi(a_t|s_t)$ is trained with actor-critic learning:

$$\pi(a_t|s_t) = \text{softmax}(\mathbf{W}_\pi \tanh(\mathbf{W}_s^{(2)} \text{ReLU}(\mathbf{W}_s^{(1)} s_t))), \tag{30}$$

$$A_t = \sum_l (\gamma\lambda)^l \delta_{t+l}. \tag{31}$$

Here, δ_{t+l} is the temporal difference (TD) error at step $t+l$, which can be computed using standard TD(λ) or generalized advantage estimation (GAE) for training stability [7]. The state s_t is a concatenation of task-level features, predicted confidence scores, semantic embeddings, and historical fallback signals, providing comprehensive context for fallback decisions.

Multi-Objective Decision. The optimal granularity k^* is selected via Pareto front:

$$\min_k \mathbf{f}(k) = [-\text{Acc}^{(k)},\ \text{Unc}^{(k)},\ 1/k,\ \text{Cost}^{(k)}]^\top, \tag{32}$$

$$\mathcal{P} = \{k^* \mid \nexists\, k:\ \mathbf{f}(k) \prec \mathbf{f}(k^*)\}. \tag{33}$$

Here, $\text{Acc}^{(k)}$ denotes the classification accuracy at level k, $\text{Unc}^{(k)}$ refers to average predictive uncertainty quantified by entropy or variance, and $\text{Cost}^{(k)}$ encodes the task complexity or computational latency associated with classifying at level k.

3. C: Multi-Level Joint Optimization (MLJO)

Multi-Level Consistency. To enforce hierarchical constraints, we use:

$$P(Y^{(k)} = c_k|\mathbf{x}) = \sum_{c_{k+1} \in \text{Ch}(c_k)} P(Y^{(k+1)} = c_{k+1}|\mathbf{x}) \cdot P(c_k|c_{k+1}, \mathbf{x}), \tag{34}$$

This conditional formulation ensures that coarse-level predictions are inferred consistently from fine-level predictions via hierarchical decomposition.

We then define the overall consistency loss as:

$$\mathcal{L}_{\text{hierarchy}} = \sum_{k=1}^{K-1} \omega_k \cdot \mathcal{L}_{\text{consist}}^{(k,k+1)} + \lambda_{\text{global}} \cdot \mathcal{L}_{\text{global}}. \tag{35}$$

Here, $\mathcal{L}_{\text{global}}$ captures hierarchical agreement across all levels, while $\mathcal{L}_{\text{consist}}^{(k,k+1)}$ enforces local consistency between adjacent levels.

Granularity Harmonization. Bidirectional distributions are regularized via:

$$\mathcal{L}_{\text{harm.}} = \text{KL}(\mathbf{p}_{\text{TD}} \| \mathbf{p}_{\text{BU}}) + \text{KL}(\mathbf{p}_{\text{BU}} \| \mathbf{p}_{\text{TD}}) + \gamma \| \mathbf{p}_{\text{TD}} - \mathbf{p}_{\text{BU}} \|_2^2. \tag{36}$$

The bidirectional distributions $\mathbf{p}_{\text{TD}}$ and $\mathbf{p}_{\text{BU}}$ are derived from top-down and bottom-up logits, and regularized to maintain alignment between reasoning directions.

Difficulty-Aware Scheduling. Information gain is used to guide curriculum:

$$\alpha_k = \frac{H(Y^{(k)}) - H(Y^{(k)}|Y^{(k-1)})}{\sum_j [H(Y^{(j)}) - H(Y^{(j)}|Y^{(j-1)})]}. \tag{37}$$

$H(Y^{(k)}|Y^{(k-1)})$ is estimated from the empirical joint distribution over hierarchical labels to quantify the conditional uncertainty at level k.

Adaptive Weight Optimization. Weights ω_k are updated through:

$$\tilde{\omega}_k = \omega_k \cdot \frac{\bar{g}}{\|\nabla_{\boldsymbol{\theta}} \mathcal{L}_k\|_2 + \epsilon}, \quad \mathcal{L}_{\text{final}} = \sum_k \tilde{\omega}_k \cdot \mathcal{L}_k. \tag{38}$$

Here, $\bar{g}$ denotes the average gradient norm across levels, and ϵ is a small constant to ensure numerical stability.

Multi-Objective Strategy. We adopt Chebyshev scalarization:

$$\mathcal{L}_{\text{total}} = \max_j \{\lambda_j \cdot |\mathcal{L}_j - z_j^*|\} + \rho \sum_j \lambda_j \cdot |\mathcal{L}_j - z_j^*|. \tag{39}$$

Each z_j^* denotes the reference value (e.g., historical best or target) for loss $\mathcal{L}_j$, guiding the model toward balanced multi-objective optimization.

Convergence Guarantee. Following Lyapunov theory:

$$V(\boldsymbol{\theta}, t) = \frac{1}{2} \sum_j \omega_j \mathcal{L}_j^2 + \frac{\gamma}{2} \|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2^2. \tag{40}$$

4. D: Experimental Data and Experimental Setup

The dataset used in this study is derived from HRSC, with further pre-processing and new data generation to meet the experimental requirements. The dataset comprises remote sensing ship images of varying scales and resolutions (300×300 to 1500×900), hierarchically annotated into coarse-grained categories and fine-grained categories. To reduce manual labeling costs, we employed an automatic hierarchical classification (HC) model based on Chang et al. [1]. The model was pre-trained on the original HRSC annotations and used to automatically label newly generated samples. The predicted hierarchical labels were then reviewed and refined to ensure consistency and integrity across levels.

To further enhance the generalization and robustness of our model, we introduced artificial degradation via a dataset augmentation function $G(t, \sigma, \eta, \lambda, \delta)$. The augmentations include noise injection, image blurring, downsampling, and cutout-based occlusion, where σ, η, λ, and δ control the strength of each transformation. Table 1 summarizes the parameter settings used for degradation simulation. The constructed samples were further filtered using the HC model to remove logically inconsistent annotations. For instance, if a sample is labeled as "Nimitz-class aircraft carrier" in fine-grained prediction but "commercial ship" in coarse-grained prediction, it is discarded. Only samples that satisfy hierarchical constraints—e.g., encoded as $s = [1, 1]$ or $[1, 0]$, where 1 denotes correct classification—are retained. To increase complexity, we use two levels—3 coarse-grained and 21 fine-grained categories. Figure 1 presents representative examples of both clean and degraded images, with corresponding distortion parameters summarized in Table 2.

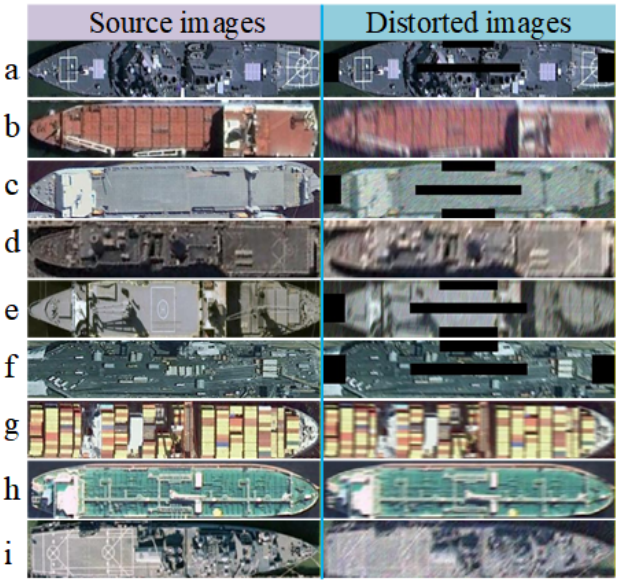

Figure 1. Examples of original and distorted images from HRSC-Deg. Distortions include noise, occlusion, and resolution degradation, simulating real-world scenarios and highlighting the challenges of fine-grained and HC under adverse conditions.

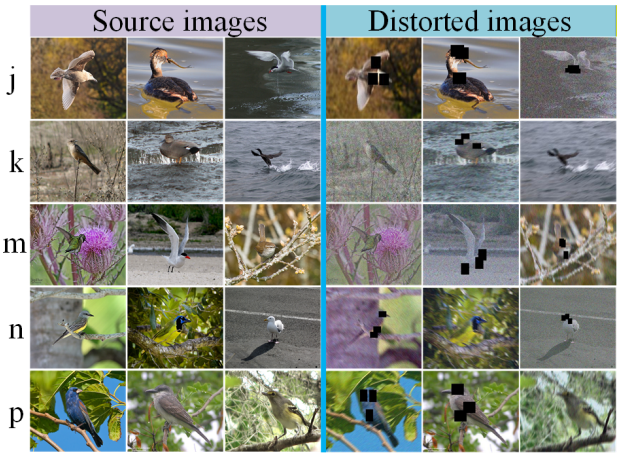

Figure 2. Examples of original and distorted images from CUB-Deg. Distortions include noise, occlusion, and resolution degradation, simulating real-world scenarios and highlighting the challenges of fine-grained and HC under adverse conditions.

In parallel, we construct the CUB-Deg dataset based on CUB-200-2011, which contains 11,788 bird images across 200 species. We restructured the taxonomy into a three-level hierarchy (order, family, species) and gener-

Table 1. Parameterized Degradation Simulation Methods for Constructing Robustness-Aware Datasets

Method	**Description**	**Parameters / Range**
White noise	Adds Gaussian white noise to the R, G, B channels of the image with the same standard deviation σ.	Standard deviation σ: 0.1 to 0.2
Motion blur	Applies motion blur to the R, G, B channels of the image using a directional kernel. The direction is randomly selected from 0° to 180°	Kernel size η: 5 to 14; Direction: 0° to 180°
Downsampling	Simulates low-resolution images by downsampling the image. The downsampling rate λ is defined as the ratio of the output image area to the original image area.	Downsampling rate λ: 0.4 to 0.7
Cutout	Simulates occlusion by randomly selecting one of five positions (center, left, right, top, bottom) within important regions of the ship image and applying zero-masking.	Occlusion position: randomly selected

Table 2. Examples of Distortion Parameters for Different Image Data. The letters "a"–"i", "j"–"k" and "m"–"p" represent randomly selected samples from the newly generated dataset.

HRSC-Deg					CUB-Deg				
Distorted img. No.	σ	δ	η	λ	Distorted img. No.	σ	δ	η	λ
a	-	0.0294	-	-	j	0.1955	0.0400	10	0.5710
b	0.1612	-	7	-	k	-	0.0385	13	0.4153
c	0.1958	0.0333	-	0.7583	m	0.1829	-	-	0.4142
d	-	-	10	0.7457	n	0.1815	0.0323	11	0.6511
e	0.1347	0.0400	-	-	p	0.1312	0.0400	13	-
f	-	0.0400	-	0.6255	-	-	-	-	-
g	-	-	9	-	-	-	-	-	-
h	-	-	-	0.7470	-	-	-	-	-
i	0.1155	-	12	0.7417	-	-	-	-	-

ated hierarchical labels accordingly. To simulate real-world degradation, we applied the same augmentation strategy $G(t, \sigma, \eta, \lambda, \delta)$ used for HRSC-Deg. The degradation parameters were randomly sampled to introduce variability in resolution, occlusion, and noise across the dataset. Similar to HRSC-Deg, samples that violate hierarchical consistency were filtered out. The final CUB-Deg dataset consists of 5994 training and 5794 testing samples, comprising 1960/1054/2980 instances (order/family/species) for training, and 1895/1052/2847 for testing, with the same multi-level annotation structure. Figure 2 shows examples of original and distorted images.

These two datasets from distinct domains—remote sensing and natural fine-grained recognition—enable comprehensive evaluation of our model's robustness and adaptability under complex, uncertain, and hierarchically structured conditions.

References

[1] Dongliang Chang, Kaiyue Pang, Yixiao Zheng, Zhanyu Ma, Yi-Zhe Song, and Jun Guo. Your "flamingo" is my "bird": Fine-grained, or not. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 11476–11485, 2021. 4

[2] Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. Large-scale object classification using label relation graphs. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pages 48–64, 2014. 2

[3] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pages 1050–1059, 2016. 2

[4] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014. 3

[5] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated re-

gression. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pages 2796–2804, 2018. 2

[6] Gabriel Peyré and Marco Cuturi. Computational optimal transport: With applications to data science. *Found. Trends Mach. Learn.*, 11(5–6):355–607, 2019. 3

[7] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016. 3