UGPL: Uncertainty-Guided Progressive Learning for Evidence-Based Classification in Computed Tomography

Supplementary Material

A1. Extended Methodology

A1.1. Global Uncertainty Estimation and Evidential Learning

Our global uncertainty estimator serves two critical functions: producing initial class predictions and generating a spatial uncertainty map to guide subsequent patch selection. We formulate this as an evidential learning problem that explicitly models uncertainty in the classification process.

A1.1.1. Global Model Architecture

Given an input CT image $\mathbf{I} \in \mathbb{R}^{H\times W\times 1}$, we employ a ResNet backbone [**?**] $\mathcal{F}_\theta$ with parameters θ to extract feature maps $\mathbf{F} \in \mathbb{R}^{h\times w\times d}$, where $h = H/32$, $w = W/32$, and d is the feature dimension:

To accommodate grayscale CT images, we modify the first convolutional layer of the ResNet [**?**] to accept single-channel inputs while preserving the pretrained weights by averaging across the RGB channels. The feature maps $\mathbf{F}$ are then processed by two parallel heads: a classification head $\mathcal{C}_\phi$ and an evidence head $\mathcal{E}_\psi$. The classification head applies global average pooling followed by a fully connected layer to produce class logits:

$$\mathbf{z}_g = \mathcal{C}_\phi(\mathbf{F}) = \mathbf{W}_\phi \cdot \text{GAP}(\mathbf{F}) + \mathbf{b}_\phi \tag{1}$$

where $\mathbf{z}_g \in \mathbb{R}^C$ represents the logits for C classes, $\mathbf{W}_\phi \in \mathbb{R}^{C\times d}$ and $\mathbf{b}_\phi \in \mathbb{R}^C$ are learnable parameters, and GAP denotes global average pooling.

$$\mathbf{F} = \mathcal{F}_\theta(\mathbf{I}) \tag{2}$$

A1.1.2. Evidential Uncertainty Estimation

The evidence head $\mathcal{E}_\psi$ generates pixel-wise Dirichlet concentration parameters that quantify uncertainty at each spatial location:

$$\mathbf{E} = \mathcal{E}_\psi(\mathbf{F}) \in \mathbb{R}^{h\times w\times 4C} \tag{3}$$

Here, $\mathbf{E}$ encodes four parameters $(\alpha, \beta, \gamma, \nu)$ for each class at each spatial location, representing a Dirichlet distribution. We implement $\mathcal{E}_\psi$ as a sequence of convolutional layers that preserve spatial dimensions while expanding the channel dimension to $4C$. Following the principles of subjective logic [**?**], we parameterize the Dirichlet distribution using these four parameters:

$$\alpha_{i,j,c} = \beta_{i,j,c} \cdot \nu_{i,j,c} + 1 \tag{4}$$

Algorithm 1 Global Uncertainty Estimation

Require: Input image $\mathbf{I} \in \mathbb{R}^{H\times W\times 1}$
Ensure: Global logits $\mathbf{z}_g$, Uncertainty map $\hat{\mathbf{U}}$
$\mathbf{F} \leftarrow \mathcal{F}_\theta(\mathbf{I})$ {Extract features using backbone}
$\mathbf{z}_g \leftarrow \mathcal{C}_\phi(\mathbf{F})$ {Compute global logits}
$\mathbf{E} \leftarrow \mathcal{E}_\psi(\mathbf{F})$ {Generate evidence parameters}
for each spatial location (i, j) and class c **do**
  $\beta_{i,j,c} \leftarrow \text{softplus}(\mathbf{E}_{i,j,c}) + \epsilon$
  $\nu_{i,j,c} \leftarrow \frac{e^{\mathbf{E}_{i,j,c+C}}}{\sum_{k=1}^{C} e^{\mathbf{E}_{i,j,k+C}}}$
  $\alpha_{i,j,c} \leftarrow \beta_{i,j,c} \cdot \nu_{i,j,c} + 1$
end for
for each spatial location (i, j) **do**
  $\mathbf{U}_{i,j} \leftarrow \frac{1}{C} \sum_{c=1}^{C} \left(\frac{1}{\alpha_{i,j,c}} + \frac{\beta_{i,j,c}}{\alpha_{i,j,c}(\alpha_{i,j,c}+1)} \right)$
end for
$\hat{\mathbf{U}} \leftarrow \frac{\mathbf{U}-\min(\mathbf{U})}{\max(\mathbf{U})-\min(\mathbf{U})+\epsilon}$ {Normalize uncertainty map}
return $\mathbf{z}_g, \hat{\mathbf{U}}$

where (i, j) denotes spatial location, c indicates the class, and $\alpha_{i,j,c} > 0$ is the concentration parameter for class c at location (i, j). The parameters $\beta_{i,j,c} > 0$ represents the inverse of uncertainty, $\nu_{i,j,c}$ represents the mass belief, and we constrain $\sum_{c=1}^{C} \nu_{i,j,c} = 1$ to ensure the mass beliefs form a valid probability distribution.

To ensure numerical stability, we apply a softplus activation $f(x) = \log(1 + e^x)$ to compute $\beta_{i,j,c}$ and a softmax function across the class dimension to compute $\nu_{i,j,c}$:

$$\beta_{i,j,c} = f(\mathbf{E}_{i,j,c}) + \epsilon \tag{5}$$

$$\nu_{i,j,c} = \frac{e^{\mathbf{E}_{i,j,c+C}}}{\sum_{k=1}^{C} e^{\mathbf{E}_{i,j,k+C}}} \tag{6}$$

where ϵ is a small positive constant for numerical stability. From these parameters, we compute the pixel-wise uncertainty map $\mathbf{U} \in \mathbb{R}^{h\times w}$ by aggregating the uncertainty across all classes:

$$\mathbf{U}_{i,j} = \frac{1}{C} \sum_{c=1}^{C} \left(\frac{1}{\alpha_{i,j,c}} + \frac{\beta_{i,j,c}}{\alpha_{i,j,c}(\alpha_{i,j,c} + 1)} \right) \tag{7}$$

This formulation captures both aleatoric uncertainty (first term) and epistemic uncertainty (second term). The aleatoric component $\frac{1}{\alpha_{i,j,c}}$ represents uncertainty due to inherent data noise, while the epistemic component

$\frac{\beta_{i,j,c}}{\alpha_{i,j,c}(\alpha_{i,j,c}+1)}$ represents uncertainty due to model knowledge limitations.

We normalize the uncertainty map to the range $[0, 1]$ for easier interpretation and subsequent processing:

$$\hat{\mathbf{U}} = \frac{\mathbf{U} - \min(\mathbf{U})}{\max(\mathbf{U}) - \min(\mathbf{U}) + \epsilon} \tag{8}$$

This normalized uncertainty map $\hat{\mathbf{U}}$ is then used to guide the patch selection process, focusing attention on regions where the global model exhibits high uncertainty. Algorithm 1 summarizes the complete process for generating the global class predictions and uncertainty map. The uncertainty map $\hat{\mathbf{U}}$ provides spatial localization of regions where the global model is uncertain about its predictions. High values in $\hat{\mathbf{U}}$ indicate regions that require further analysis through local patch examination. This uncertainty-guided approach allows our model to focus computational resources on diagnostically relevant regions.

A1.2. Uncertainty-Guided Patch Selection and Local Refinement

A1.2.1. Progressive Patch Extraction

Given an input image $\mathbf{I} \in \mathbb{R}^{H \times W \times 1}$ and its corresponding uncertainty map $\hat{\mathbf{U}} \in \mathbb{R}^{h \times w}$ from the global model, we first upsample the uncertainty map to match the input resolution:

$$\mathbf{U}' = \mathcal{U}(\hat{\mathbf{U}}, (H, W)) \tag{9}$$

where $\mathcal{U}$ represents bilinear upsampling to dimensions (H, W). Our objective is to extract K patches of size $P \times P$ from regions with high uncertainty while ensuring diversity among the selected patches. We formulate this as a sequential optimization problem where each patch is selected to maximize uncertainty while maintaining a minimum distance from previously selected patches. For the first patch, we simply select the region with maximum uncertainty:

$$(x_1, y_1) = \arg\max_{(x,y)} \mathbf{U}'_{x:x+P, y:y+P} \tag{10}$$

where (x_1, y_1) represents the top-left corner of the first patch, and $\mathbf{U}'_{x:x+P,y:y+P}$ denotes the mean uncertainty within the patch region. For subsequent patches $k = 2, 3, \ldots, K$, we introduce a spatial penalty to encourage diversity:

$$(x_k, y_k) = \arg\max_{(x,y)} \Big[\mathbf{U}'_{x:x+P,\, y:y+P} - \lambda \cdot \min_{i<k} d((x, y),\, (x_i, y_i)) \Big] \tag{11}$$

where $d((x, y), (x_i, y_i))$ computes the Euclidean distance between patch centers, λ is a weighting parameter controlling diversity, and $\min_{i<k}$ finds the minimum distance to any previously selected patch. To implement this efficiently while avoiding explicit computation of the penalty term for all possible locations, we apply a non-maximum suppression (NMS) approach. After selecting each patch, we suppress a region around it by applying a penalty mask to the uncertainty map:

$$\mathbf{U}'_{x-M:x+P+M,\, y-M:y+P+M} = \mathbf{U}'_{x-M:x+P+M,\, y-M:y+P+M} \times (1 - \mathbf{G}) \tag{12}$$

where M is a margin parameter and $\mathbf{G}$ is a Gaussian kernel that applies a stronger suppression near the center of the selected patch and gradually reduces toward the edges.

Our algorithm incorporates several fallback mechanisms to handle edge cases and ensure reliable operation:

- **Uncertainty Threshold Handling:** In situations where no high-uncertainty regions remain (when all uncertainty values are suppressed below a specified threshold), the method falls back to random selection to preserve sample diversity.
- **Boundary Checking:** Comprehensive boundary checking is applied to prevent selected patches from extending beyond the image borders, ensuring valid patch extraction even at image edges.
- **Dynamic Size Adjustment:** To accommodate extremely small images or atypical aspect ratios, the algorithm dynamically adjusts patch sizes, ensuring consistent and valid outputs across varying input dimensions.

These mechanisms collectively ensure robust operation across diverse medical imaging datasets with varying characteristics.

A1.2.2. Local Refinement Network Architecture

After extracting the K patches $\{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_K\}$, we process each patch independently using a local refinement network. This network comprises three components: a feature extractor, a classification head, and a confidence estimation head.

The feature extractor $\mathcal{L}_f$ processes each patch to obtain local feature vectors:

$$\mathbf{f}_k = \mathcal{L}_f(\mathbf{P}_k) \in \mathbb{R}^{d_l} \tag{13}$$

where d_l is the feature dimension. We implement $\mathcal{L}_f$ as a sequence of convolutional layers followed by pooling operations to progressively reduce spatial dimensions while increasing feature depth. Specifically, our implementation uses four convolutional blocks with increasing channel dimensions (64→128→256→256), each followed by batch normalization, ReLU activation, and max pooling. The final features undergo adaptive average pooling to produce a fixed-dimensional representation regardless of input patch size.

The classification head $\mathcal{L}_c$ maps these features to class logits:

$$\mathbf{z}_{l,k} = \mathcal{L}_c(\mathbf{f}_k) \in \mathbb{R}^C \tag{14}$$

This head is implemented as a two-layer MLP with a hidden dimension of 128 and ReLU activation between layers. Simultaneously, the confidence estimation head $\mathcal{L}_{\text{conf}}$ produces a scalar confidence score for each patch:

$$c_k = \mathcal{L}_{\text{conf}}(\mathbf{f}_k) \in [0, 1] \tag{15}$$

where c_k represents the model's confidence in its prediction for patch k. We implement $\mathcal{L}_{\text{conf}}$ as a small MLP with a sigmoid activation function on the output to constrain the confidence score to the range $[0, 1]$. This two-layer MLP has a hidden dimension of 64 and uses ReLU activation between layers.

The confidence score serves two critical purposes: (1) it allows the model to express uncertainty about individual patch predictions, and (2) it provides a weight for the subsequent fusion of local predictions. Patches with higher confidence scores will contribute more significantly to the final classification decision. For each patch k, we obtain both class logits $\mathbf{z}_{l,k}$ and a confidence score c_k. The combined local prediction is computed as a confidence-weighted average of the patch predictions:

$$\mathbf{z}_l = \frac{\sum_{k=1}^{K} c_k \cdot \mathbf{z}_{l,k}}{\sum_{k=1}^{K} c_k + \epsilon} \tag{16}$$

where ϵ is a small constant (typically 10^{-6}) for numerical stability. This formulation naturally handles cases where some patches have very low confidence, effectively reducing their contribution to the final prediction.

The local refinement network provides detailed analysis of suspicious regions identified by the global model, capturing fine-grained features that might be missed in the global analysis. By assigning confidence scores to each patch, the network also performs an implicit form of attention, focusing on the most discriminative patches for the final classification decision.

A1.3. Adaptive Fusion and Training Objectives

A1.3.1. Adaptive Fusion Module

The adaptive fusion module dynamically determines the optimal weighting between global and local predictions for each input image. Given the global logits $\mathbf{z}_g \in \mathbb{R}^C$ and uncertainty map $\hat{\mathbf{U}} \in \mathbb{R}^{h \times w}$ from the global model, and local logits $\mathbf{z}_l \in \mathbb{R}^C$ with patch confidence scores $\{c_1, c_2, \ldots, c_K\}$ from the local refinement network, we compute a scalar representation of the global uncertainty by averaging across the spatial dimensions:

$$u_g = \frac{1}{h \cdot w} \sum_{i=1}^{h} \sum_{j=1}^{w} \hat{\mathbf{U}}_{i,j} \tag{17}$$

This scalar uncertainty $u_g \in [0, 1]$ quantifies the overall confidence of the global model. The fusion network $\mathcal{F}_\omega$ takes as input the global logits $\mathbf{z}_g$ and the global uncertainty score u_g, concatenated into a single vector $[\mathbf{z}_g, u_g] \in \mathbb{R}^{C+1}$. The network outputs a fusion weight $w_g \in [0, 1]$ that determines the relative contribution of global versus local predictions:

$$w_g = \mathcal{F}_\omega([\mathbf{z}_g, u_g]) \tag{18}$$

We implement $\mathcal{F}_\omega$ as a multi-layer perceptron with sigmoid activation on the output:

$$\mathcal{F}_\omega([\mathbf{z}_g, u_g]) = \sigma(W_2 \cdot \text{ReLU}(W_1 \cdot [\mathbf{z}_g, u_g] + b_1) + b_2) \tag{19}$$

where $W_1 \in \mathbb{R}^{d_f \times (C+1)}$, $W_2 \in \mathbb{R}^{1 \times d_f}$, $b_1 \in \mathbb{R}^{d_f}$, and $b_2 \in \mathbb{R}$ are learnable parameters, d_f is the hidden dimension, and σ is the sigmoid function. The fusion weight w_g represents the contribution of the global prediction, while $w_l = 1 - w_g$ represents the contribution of the local prediction. The fused logits $\mathbf{z}_f$ are computed as:

$$\mathbf{z}_f = w_g \cdot \mathbf{z}_g + (1 - w_g) \cdot \mathbf{z}_l \tag{20}$$

This adaptive weighting allows the model to rely more on global features when the global model is confident (low uncertainty), and more on local features when the global model is uncertain (high uncertainty).

A1.3.2. Multi-component Loss Function

Our comprehensive loss function addresses multiple objectives simultaneously. The total loss $\mathcal{L}_{\text{total}}$ is a weighted sum of several components:

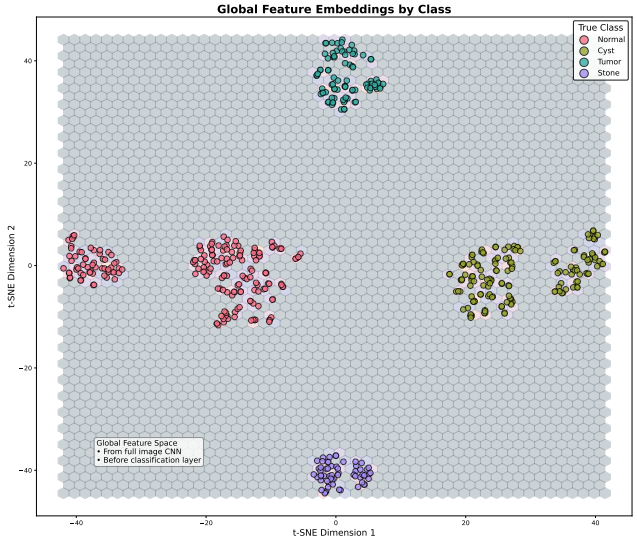
$$\begin{aligned}\mathcal{L}_{\text{total}} = {} & \lambda_f \mathcal{L}_{\text{fused}} + \lambda_g \mathcal{L}_{\text{global}} + \lambda_l \mathcal{L}_{\text{local}} \\ & + \lambda_u \mathcal{L}_{\text{uncertainty}} + \lambda_c \mathcal{L}_{\text{consistency}} + \lambda_{\text{conf}} \mathcal{L}_{\text{confidence}} \\ & + \lambda_d \mathcal{L}_{\text{diversity}}\end{aligned} \tag{21}$$

where $\lambda_f, \lambda_g, \lambda_l, \lambda_u, \lambda_c, \lambda_{\text{conf}}$, and λ_d are weighting coefficients for each loss component.

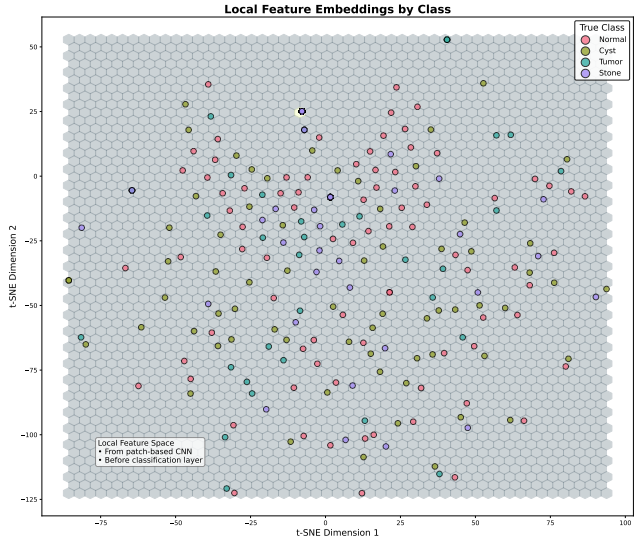
Classification Losses. We apply cross-entropy loss to the predictions from each component of our framework:

$$\mathcal{L}_{\text{fused}} = -\sum_{i=1}^{C} y_i \log(\text{softmax}(\mathbf{z}_f)_i) \tag{22}$$

$$\mathcal{L}_{\text{global}} = -\sum_{i=1}^{C} y_i \log(\text{softmax}(\mathbf{z}_g)_i) \tag{23}$$



(a) Global feature embeddings showing clear class separation with distinct clusters for each kidney condition (Normal, Cyst, Tumor, Stone). The global model learns discriminative whole-image representations that establish strong decision boundaries between classes.



(b) Local feature embeddings exhibiting significant class mixing without distinct clusters. The local model focuses on fine-grained details within uncertain regions, capturing complementary information not directly aligned with class boundaries.

Figure 1. Comparison of t-SNE visualizations for feature spaces in the kidney dataset. (a) Global features from the full-image CNN form well-separated clusters by class, demonstrating effective overall classification capability. (b) Local features from patch-based analysis show substantial mixing across classes, indicating their focus on subtle variations and uncertainty resolution rather than direct class discrimination. This complementary representation underscores why adaptive fusion of both feature types produces superior performance.

$$\mathcal{L}_{\text{local}} = \frac{1}{K}\sum_{k=1}^{K} -\sum_{i=1}^{C} y_i \log(\text{softmax}(\mathbf{z}_{l,k})_i) \tag{24}$$

where y_i is the ground truth label for class i (one-hot encoded), and $\text{softmax}(\mathbf{z})_i$ denotes the softmax probability for class i given logits $\mathbf{z}$.

Uncertainty Calibration Loss. To ensure that the uncertainty map accurately reflects prediction errors, we introduce an uncertainty calibration loss:

$$\mathcal{L}_{\text{uncertainty}} = \text{MSE}(\hat{\mathbf{U}}, 1 - \mathbf{C}) \tag{25}$$

where $\mathbf{C} \in \{0,1\}^{h \times w}$ is a correctness map derived from the global predictions. For each spatial location (i,j), $\mathbf{C}_{i,j} = 1$ if the predicted class at that location matches the ground truth, and $\mathbf{C}_{i,j} = 0$ otherwise. This loss encourages high uncertainty in regions where the global model makes errors and low uncertainty where predictions are correct.

Consistency Loss. To promote consistency between global and local predictions, we use a Kullback-Leibler (KL) divergence loss:

$$\mathcal{L}_{\text{consistency}} = \frac{1}{K}\sum_{k=1}^{K} \text{KL}(\text{softmax}(\mathbf{z}_{l,k}) \| \text{softmax}(\mathbf{z}_g)) \cdot c_k \tag{26}$$

where $\text{KL}(P\|Q) = \sum_i P_i \log(P_i/Q_i)$ is the KL divergence, and c_k is the confidence score for patch k. This loss is weighted by the patch confidence, reducing the penalty for inconsistency in low-confidence patches.

Confidence Regularization Loss. To align patch confidence scores with prediction accuracy, we introduce a confidence regularization loss:

$$\mathcal{L}_{\text{confidence}} = \frac{1}{K}\sum_{k=1}^{K} \text{MSE}(c_k, a_k) \tag{27}$$

where $a_k \in \{0,1\}$ indicates whether the prediction for patch k is correct ($a_k = 1$) or incorrect ($a_k = 0$). This loss encourages high confidence for correct predictions and low confidence for incorrect predictions.

Diversity Loss. To encourage diversity among patch predictions, we include a diversity loss:

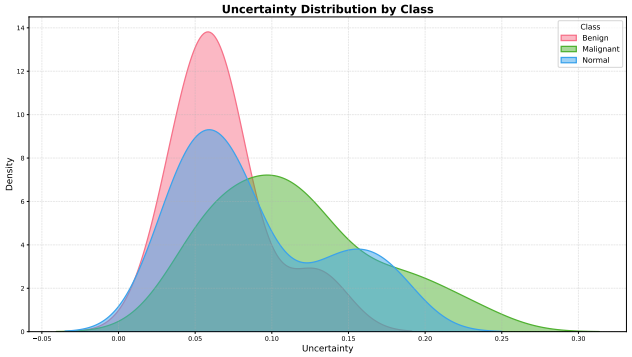


Figure 2. Uncertainty distribution by class for lung cancer detection. Malignant cases (green) exhibit significantly higher average uncertainty and broader distribution compared to benign cases (pink), which show a tighter, lower-uncertainty distribution. Normal cases (blue) display a distinctive bimodal distribution with peaks at both low and moderate uncertainty levels.

$$\mathcal{L}_{\text{diversity}} = \frac{1}{K(K-1)/2} \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} \cos\left(\text{softmax}(\mathbf{z}_{l,i}),\ \text{softmax}(\mathbf{z}_{l,j})\right) \tag{28}$$

where $\cos(a, b) = \frac{a \cdot b}{||a|| \cdot ||b||}$ is the cosine similarity between vectors. This loss penalizes similarity between patch predictions, encouraging each patch to contribute unique information.

A2. Implementation Details

All models are trained for 100 epochs with early stopping based on validation loss with a patience of 7 epochs on a single NVIDIA RTX 3090 GPU. We employ an Adam optimizer [**?**] with a learning rate of 1×10^{-4} and weight decay of 1×10^{-4}, with a batch size of 96 and a cosine decay learning rate scheduler [**?**]. For data augmentation [**?**] during training, we apply random horizontal and vertical flips, random rotation (±10°), random affine transformations (±5% translation), and contrast/brightness adjustments (±10%). Images are normalized to the [0,1] range after applying appropriate windowing for CT images. We do not use EMA [**?**] since it does not improve performance.

Model configurations are adapted for each dataset as follows: the Kidney dataset uses a ResNet-18 [**?**] backbone with a patch size of 64 and 3 patches per image, the Lung dataset uses a ResNet-50 [**?**] backbone with a patch size of 64 and 2 patches per image, and the COVID dataset uses a ResNet-18 [**?**] backbone with a patch size of 64 and 4 patches per image. The multi-component loss function assigns weights of 1.0 for the fused loss, 0.5 for global and local losses, 0.3 for the uncertainty loss, 0.2 for the consistency loss, and 0.1 for both the confidence and diversity losses.

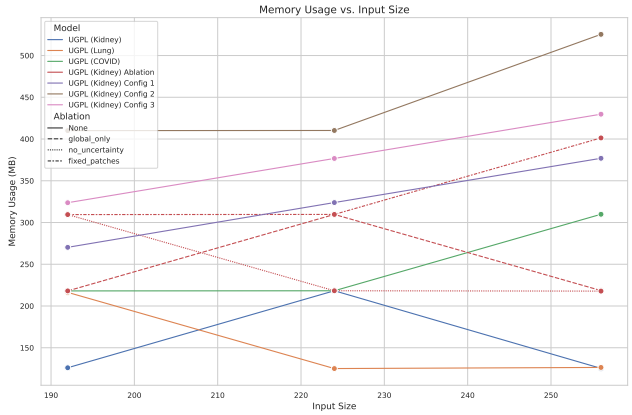


Figure 3. Memory usage scaling with input dimensions across UGPL variants. Lines represent different model configurations and ablations. Config 2 (brown line) consistently demonstrates the highest memory requirements due to its ResNet-50 [**?**] backbone variant. Some configurations show counterintuitive scaling behavior, particularly at larger input sizes, highlighting complex interactions between model architecture and GPU memory management.

A3. Additional Experiments and Results

A3.1. Feature Space Analysis

To better understand how UGPL learns different representations at global and local scales, we visualize the feature embeddings from both network components using t-SNE. Figure 1 demonstrates the contrast between global and local feature spaces for the kidney CT dataset [**?**].

The global feature embeddings (Figure 1a) display remarkably clear separation between classes, with distinct clusters forming for each pathological condition. This indicates that the global network successfully learns discriminative features that establish strong decision boundaries at the whole-image level. In contrast, the local feature embeddings (Figure 1b) exhibit substantial mixing between classes with no clear cluster formation, suggesting that the local network captures different characteristics altogether.

The global network provides robust overall classification by learning class-separable features, while the local network focuses on fine-grained details within uncertain regions that may not align directly with class boundaries but capture subtle variations critical for resolving ambiguous cases. When these complementary features are combined through our adaptive fusion mechanism, the model effectively leverages both the discriminative power of global features and the detailed analysis of local features, particularly in challenging regions where global analysis alone might be

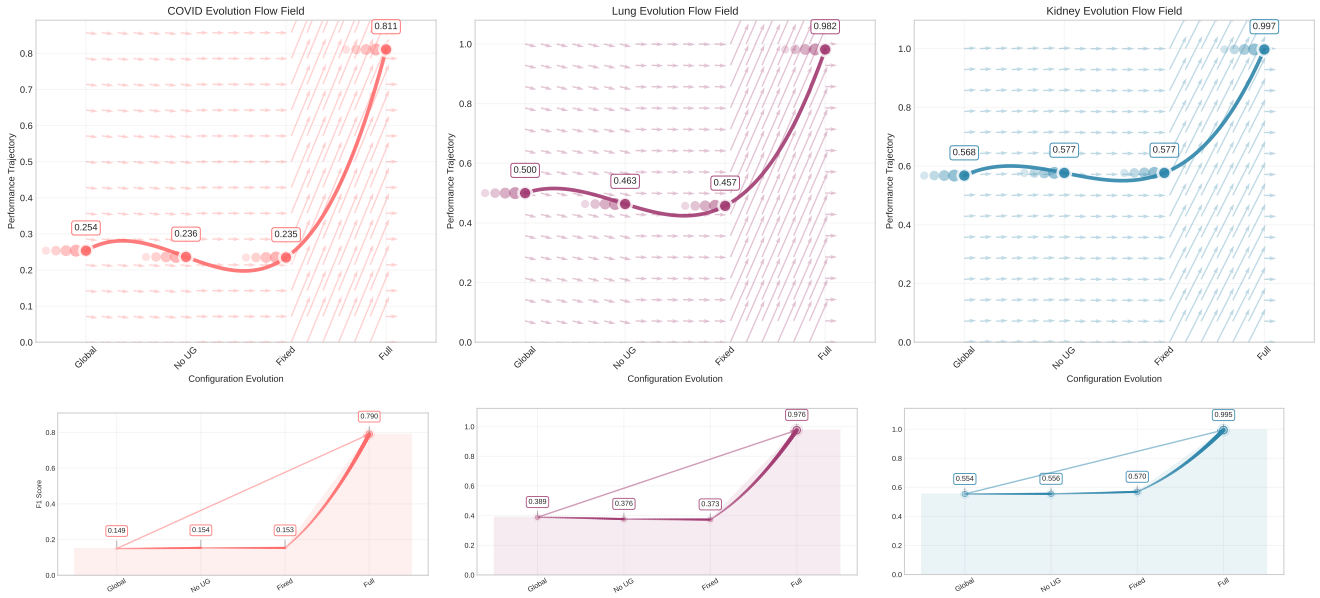


Figure 4. Evolution of model performance across different configurations. Top: Flow field visualization showing performance trajectories from simplified to complete model configurations for each dataset. Bottom: F1 score progression across configurations for COVID (left), Lung (middle), and Kidney (right) datasets, highlighting the dramatic improvement when all components are integrated in the full model.

insufficient.

The dispersed nature of local embeddings also validates our patch selection approach - these patches represent precisely those regions where additional analysis is most beneficial, as they contain ambiguous features that the global model finds difficult to classify confidently. This feature space analysis provides concrete evidence for why progressive refinement is more effective than single-pass approaches for medical image classification.

A3.2. Uncertainty Calibration Analysis

Figure 2 visualizes the distribution of pixel-wise uncertainty values across diagnostic classes in the lung cancer dataset [**? ? ?**]. The distinct separation between uncertainty profiles demonstrates the model's ability to calibrate uncertainty in a clinically meaningful way. Malignant cases consistently show higher uncertainty (mean 0.14, standard deviation 0.07) compared to benign cases (mean 0.06, standard deviation 0.03), reflecting the inherently more complex and variable presentation of malignant lesions. Normal cases exhibit an intriguing bimodal distribution, suggesting the existence of two distinct subgroups within what radiologists classify as normal tissue. This aligns with clinical practice, where some normal cases closely resemble benign findings (first mode) while others contain subtle variations that warrant closer inspection (second mode). The UGPL framework effectively leverages these uncertainty patterns to guide computational resource allocation, focusing detailed analysis precisely where diagnostic ambiguity is highest.

A3.3. Ablation Evolution

Figure 4 visualizes performance evolution across configurations. All datasets show minimal variations among simplified configurations followed by dramatic jumps with the full model - COVID F1 scores improve 5.3× (0.15 to 0.79), lung dataset by 2.6× (0.37 to 0.98), and kidney dataset by 1.7× (0.57 to 0.99).

A3.4. Computational Efficiency Analysis

We analyze computational efficiency of UGPL across different configurations and ablations to understand tradeoffs between model complexity and performance. Figure 5 shows the relationship between computational complexity (measured in GFLOPs) and inference time. The full UGPL model requires approximately 3-5 GFLOPs depending on the dataset and configuration, with inference times between 4.5-6.7ms on an NVIDIA P100 (we use a lightweight GPU for inference to better reflect real-world deployment settings). The global-only ablation (without patch extraction and local refinement) reduces inference time by 27-36% across all datasets, demonstrating the computational cost of the progressive analysis components. Higher-capacity backbones (Config 2 with ResNet-50 variant) increase both GFLOPs and inference time by approximately 45% compared to the standard configurations.

Memory efficiency is another critical factor for medical imaging applications. Figure 3 illustrates how memory us-

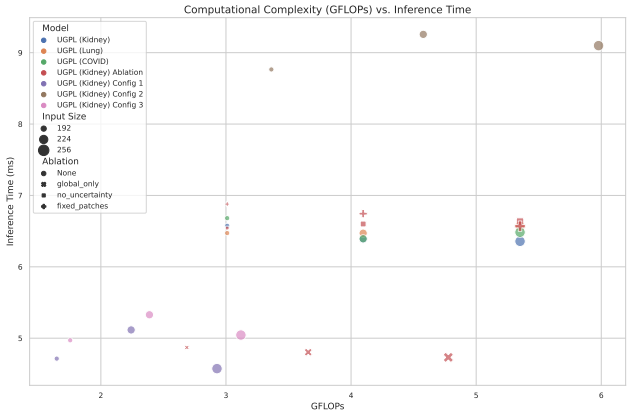


Figure 5. Computational complexity (GFLOPs) versus inference time (ms) for UGPL variants. Points are colored by dataset, with marker style indicating ablation type and size representing input dimensions.

age scales with input image dimensions. We observe nonlinear scaling patterns that vary significantly across configurations. The ResNet-50 backbone (Config 2) requires 1.4-1.7× more memory than ResNet-18 configurations. Interestingly, ablations demonstrate dataset-specific memory profiles: for the COVID dataset, memory usage increases linearly with input size, while the Kidney dataset shows more complex patterns. The global-only ablation demonstrates inconsistent memory scaling, suggesting that optimizations in GPU memory management affect different architectural components differently.

UGPL model requires more computational resources than simplified variants, and the progressive learning approach maintains reasonable efficiency for clinical deployment. The additional cost of uncertainty estimation and local refinement is justified by the significant performance improvements, particularly for challenging cases.