Improving Medical Imaging Model Calibration through Probabilistic Embedding

Bonian Han
Dept. of Statistics
Hangzhou Dianzi University
Hangzhou, China
bonian985@gmail.com

Gongbo Liang
Dept. of Computational, Engineering, and Mathematical Sciences
Texas A&M University-San Antonio
San Antonio, TX, USA
gliang@tamusa.edu

***Abstract*—Neural network model calibration is crucial in medical imaging, where accurate probabilistic predictions are essential for informed decision-making. Existing calibration techniques often introduce additional complexity and may not fully capture the inherent uncertainty associated with the tasks. To address these challenges, we propose a novel approach based on probabilistic embedding that models uncertainty through a Gaussian distribution. By embedding the model's predictions into a probabilistic space, the proposed method enables effective uncertainty quantification. We demonstrate the effectiveness of our approach on multiple medical imaging tasks. The experimental result shows our method outperforms existing techniques in terms of both calibration and accuracy.**

***Index Terms*—predictive modeling, classification, neural network, deep learning, uncertainty estimation, trustworthy ai.**

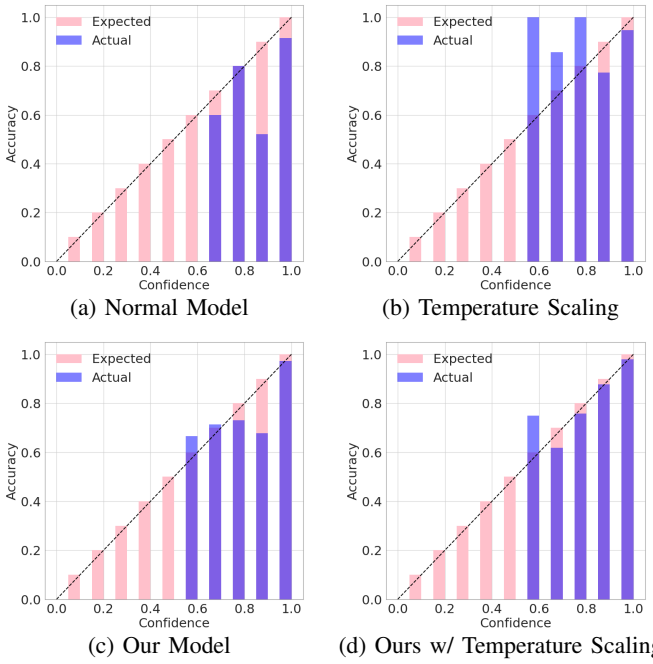
I. Introduction

Medical imaging, a cornerstone of modern healthcare, is heavily used for diagnosis, treatment planning, and prognosis estimation [1]–[4]. Over the past decade, numerous deep neural network models have been developed for medical imaging tasks, ranging from image processing [5]–[7] to diagnosis [8]–[10]. While neural network models achieve impressive performance in virtually every domain, such as cybersecurity [11]–[13], public transportation [14]–[16], and astrophysics [17]–[19], they often suffer from overconfidence or underconfidence, leading to potential misinterpretations [20]–[23]. This issue is known as model miscalibration, particularly critical in medical contexts where accurate probabilistic predictions are essential for informed decision-making [24], [25].

Existing methods for improving model calibration in medical imaging typically focus on post-hoc techniques, such as temperature scaling [20] or Dirichlet calibration [26], or auxiliary loss function, such as MMCE [22] and DCA [25]. While these methods are effective to some extent, they often introduce additional complexity and may not fully address the underlying calibration issues. Moreover, they may not adequately capture the inherent uncertainty associated with medical imaging tasks, which can arise from factors such as image quality, patient variability, and model limitations.

To address these challenges, we propose a novel calibration method that based on probabilistic embedding. Our method models uncertainty utilizing Gaussian distributions. Instead of predicting a point estimate for a given sample, our method predicts a Gaussian blob, with the mean representing the predicted label and the variance indicating the uncertainty. By embedding the model's predictions into a probabilistic space, our method effectively quantifies the uncertainty associated with each prediction, leading to more calibrated and interpretable results.

The rest of this paper is organized as follows. We first provide the necessary background of neural network miscalibration (Section II), followed by a detailed introduction of our proposed probabilistic embedding approach and its theoretical underpinnings (Section III). We then present the evaluation results and compare our method to existing calibration tech-



(a) Normal Model (b) Temperature Scaling

(c) Our Model (d) Ours w/ Temperature Scaling

Fig. 1: Reliability diagrams on the Mendeley dataset show (a) the Normal model is overconfident in its predictions; (b) temperature scaling over-compensated model calibration by making it underconfident in its predictions; (c) and (d) significantly improves the model calibration.

The code will be released upon acceptance.

niques (Section IV). Finally, we conclude with a discussion and outline future research directions (Section V).

II. Background

A. Neural Network Calibration

1) Problem Definition: Mathematically, the problem of model calibration can be defined in the following way. Let the input $x \in X$ and label $y \in Y = \{1, ..., k\}$ be random variables that follow a joint distribution $\pi(x, y) = \pi(y|x)\pi(x)$. Let $h(\cdot)$ be a deep neural network with $h(x) = (\hat{y}, \hat{p})$, where $\hat{y}$ is the predicted class label and $\hat{p}$ is the associated confidence. We would like the confidence estimate $\hat{p}$ to be calibrated, which intuitively means that $\hat{p}$ represents a true probability, p. The perfect calibration can be defined as:

$$\mathbb{P}\left(\hat{y} = y|\hat{p} = p\right) = p, \forall p \in [0, 1]. \tag{1}$$

For instance, given 100 predictions with the average $\hat{P} = 0.95$ from a perfect calibrated model, we expect that 95 predictions should be correct. In reality, the average confidence of a deep neural network is often higher than its accuracy [20]–[22]. The difference in expectation between confidence and accuracy (i.e., the calibration error) can be defined as:

$$\mathbb{E}_{\hat{p}}\left[|(\hat{y} = y|\hat{p} = p) - p|\right]. \tag{2}$$

2) Measurement: Expected Calibration Error (ECE) [27] is a commonly used criterion for measuring neural network calibration error that approximates Equation (2) by partitioning predictions into M bins and taking a weighted average of the difference between the accuracy and confidence for each bin. To calculate ECE, all the samples need to be grouped into M interval bins according to the predicted probability. Let B_m be the set of indices of samples whose predicted confidence falls into the interval $I_m = (\frac{m-1}{M}, \frac{m}{M}]$, $m \in M$. The accuracy of B_m is

$$\text{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} 1(\hat{y}_i = y_i), \tag{3}$$

where $\hat{y}_i$ and y_i are the predicted and ground-truth labels for sample i. The average predicted confidence of bin B_m can be defined as

$$\text{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i, \tag{4}$$

where $\hat{p}_i$ is the confidence of sample i. ECE can be defined with $\text{acc}(B_m)$ and $\text{conf}(B_m)$

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left|\text{acc}(B_m) - \text{conf}(B_m)\right|, \tag{5}$$

where n is the number of samples.

Maximum Calibration Error (MCE) [27] is another common criterion for measuring neural network calibration error that partitions predictions into M equally-spaced bins and estimates the worst-case scenario. MCE can be computed as:

$$\text{MCE} = \max_{m \in \{1,...,m\}} |\text{acc}(B_m) - \text{conf}(B_m)|. \tag{6}$$

Unlike ECE, MCE is known as sensitive to the number of bins [20], [25].

B. Existing Method

The existing calibration methods may be grouped into three broader categories, namely post-hoc processing, auxiliary regularization loss, and uncertainty estimation. Our proposed method can be considered as the third category. This section provides the basic introduction to all the three categories.

1) Post-hoc Processing: Post-hoc calibration techniques are methods designed to improve the calibration of machine learning models after training is complete. These approaches aim to adjust the model's predicted probabilities to better align with the true probabilities of the target classes.

One of the most widely-used post-hoc approaches for neural network model calibration is temperature scaling [20], [28], which addresses miscalibration by dividing the logits by a temperature parameter, T.

Temperature scaling typically involves two steps: 1) model training and 2) learning the T. Once the model is trained, T is added and optimized on a validation set while all other model parameters remain fixed. This optimization process aims to find the optimal temperature value that improves the calibration of the model's predictions [20].

Once the optimal temperature value is found, the temperature parameter will be used for calibration at the testing time. The calibrated confidence, $\hat{q}_i$, using temperature scaling is

$$\hat{q}_i = \max_k \theta_{SM}(\frac{z_i}{T})^{(k)}, \tag{7}$$

where k is the class label ($k = 1, ..., K$), $\theta_{SM}(z_i)$ is the predicted confidence. As $T \to \infty$, the confidence $\hat{q}_i$ approaches the minimum, which indicates maximum uncertainty.

Temperature scaling is easy to use and performs well. However, as a post-processing technique, temperature scaling does not directly contribute to feature learning. Ideally, a neural network model should be capable of self-calibration without requiring external adjustments like temperature scaling [29].

2) Auxiliary Losses: To improve model calibration during training, auxiliary regularization losses can be incorporated into the negative log-likelihood (NLL) loss function. The combined loss can be expressed as:

$$\text{Loss} = \text{NLL} + \beta\text{Auxiliary\_Loss}, \tag{8}$$

where β is a weight scalar and `Auxiliary_Loss` is the regularization loss for calibration, which may have multiple options, such as entropy [21], MMCE [22], DCA [25], etc.

MMCE is computed in a reproducing kernel Hilbert space (RKHS) [30]. The completely loss function can be written as:

$$\text{MMCE} = \sum_{i,j \in D} \frac{(\hat{y}_i - \hat{p}_i)(\hat{y}_j - \hat{p}_j)k(\hat{p}_i, \hat{p}_j)}{m^2}, \tag{9}$$

where D denotes a dataset and $k(.,.)$ is a universal kernel [31]. While MMCE can be effective, its performance may be limited by imbalanced predictions from the neural network.

DCA is based on the expected calibration error (Equation 2) and directly minimizes the difference between predicted confi-

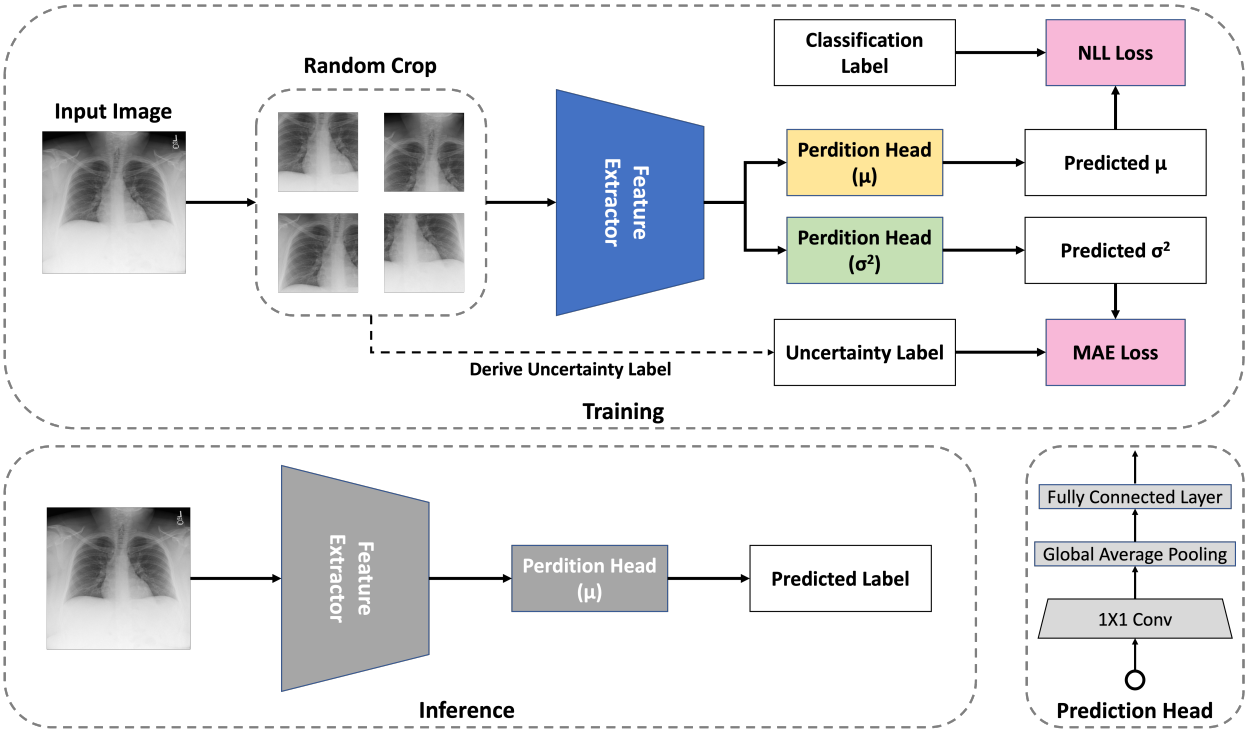


Fig. 2: Illustration of the proposed method for both training (top) and inference (bottom).

dence and accuracy. The DCA term can be computed for each mini-batch using the following equation:

$$\text{DCA} = \left| \frac{1}{N} \sum_{i=1}^{N} c_i - \frac{1}{N} \sum_{i=1}^{N} p(\hat{y}_i) \right|, \tag{10}$$

where $c_i = 1$, if $\hat{y}_i = y_i$; otherwise, $c_i = 0$.

While auxiliary losses can enhance both calibration and overall model performance, it is crucial to carefully select the weight scalar (β) associated with the auxiliary loss. An inappropriate weight can lead to suboptimal calibration or even hinder the model's performance. The optimal weight scalar may vary depending on the specific task and dataset, requiring intensive experimentation and hyperparameter tuning.

3) Uncertainty Estimation: Enhancing a model's uncertainty estimation ability can also contribute to improved calibration. Label smoothing [32] was initially proposed to enhance the classification performance of the Inception architecture. Müller et al. demonstrated that label smoothing can implicitly calibrate models by introducing a degree of uncertainty into the training process [33].

Instead of targeting a hard probability of 1.0 for the correct class, label smoothing aims to predict a softer version:

$$y_k^{LS} = y_k(1 - \alpha) + \frac{\alpha}{K}, \tag{11}$$

where y_k is original targeting probability (i.e., $y_k = 1.0$ for the correct class and $y_k = 0.0$ for the rest), K is the number of class labels, α is a hyperparameter that smooths the target.

Mixup [34] is another method that encourages softer target predictions by randomly mixing training samples. During training, two samples from different classes are combined, and the network is required to predict the combined probability of the corresponding labels. The target probabilities are proportional to the pixel contributions from each image. Thulasidasan et al. [35] have shown that Mixup can also be beneficial for neural network calibration.

Our proposed method can be considered a member of this category, as it also focuses on improving calibration by introducing uncertainty into the training process.

III. Method

A. Architecture Overview

This section provides a detailed introduction of the proposed method that enhances calibration through modeling neural network's uncertainty using Gaussian distribution (Section III-B). Figure 2 shows the overview of the proposed method for both training (top) and inference (bottom).

Given an input image, x, with a target label y, we first perform a random crop, x', of the image during training. The cropped image x' is then used as input to a deep neural network, $h(\cdot)$. Since the x' is derived from x, it inherits the target label of x. However, due to the partial nature of the crop, $h(x')$ should exhibit higher uncertainty than $h(x)$. An uncertainty label, y_c, for x' can be determined based on the random crop size and location (Section III-C).

The model $h(\cdot)$ comprises a feature extractor and two prediction heads: one for estimating the mean (μ) and the other for estimating the variance (σ^2) of a Gaussian distribution. The optimization of $h(\cdot)$ aims to minimize the distance

between μ and y while simultaneously pushing σ^2 towards y_c (Section III-D).

During inference, the entire input image, without cropping, is used as input to $h(\cdot)$. The prediction head for σ^2 is removed, and the predicted μ is used as the final predicted label.

B. Uncertainty Modeling through Gaussian Distribution

Traditionally, deep neural networks are trained to predict a point estimate for a given sample. For example, when presented with an image of a cat, a model is expected to predict whether the image indeed depicts a cat. While these models are optimized for accuracy, they often neglect to explicitly estimate the uncertainty associated with their predictions, leading to potentially inaccurate confidence scores. Instead of predicting a point estimation, we propose a novel approach that models the neural network outputs as Gaussian blobs centered around the predicted label. The size of the Gaussian blob represents the model's uncertainty, with smaller blobs indicating lower uncertainty in the prediction.

Given an input x and the target y, our goal is to let the model $h(\cdot)$ output a Gaussian distribution, $\mathcal{N}(\mu, \sigma^2)$, where μ and σ^2 are the mean and variance, respectively. Ideally, the distribution should be centered around the target label, i.e., the output of $h(x)$ should be $\mathcal{N}(y, \sigma^2)$. The variance indicates the model's uncertainty, with a smaller σ^2 suggesting lower uncertainty in the prediction.

A perfect model would have $\mu = y$ and $\sigma^2 = 0$, resulting in a point distribution $\mathcal{N}(y, 0)$. As σ increases, the Gaussian distribution becomes wider and flatter, indicating higher uncertainty. When σ^2 approaches infinity, the distribution becomes completely flat, suggesting a random model that is unable to make informative predictions beyond random guessing.

To simplify the problem and ensure a practical range of uncertainty, we impose a constraint on the estimated variance, limiting it between 0 and 1. This constraint helps to prevent the model from producing overly uncertain predictions that might be difficult to interpret or use for decision-making.

C. Uncertainty Label Generation

For a random crop x' of the input image x, the uncertainty label can be determined based on the crop size and location. Let h and w are the width and height of x, respectively, while h' and w' are the width and height of x'.

The uncertainty regarding crop size, σ_s can be assessed relative to the proportion of the crop. We first define the certainty of the crop, denoted as $certainty_s$:

$$certainty_s = \frac{h' \times w'}{h \times w}. \tag{12}$$

Consequently, the uncertainty σ_s can be calculated as:

$$\sigma_s = 1 - certainty_s = 1 - \frac{h' \times w'}{h \times w}. \tag{13}$$

Assuming all inputs are resized to a square before cropping, and the crop is always square, Equation 13 simplifies to:

$$\sigma_s = 1 - \frac{s'^2}{s^2}, \tag{14}$$

where s is the side length of the resized x and s' is the side length x'.

The σ_s calculation assumes that all pixels in an image contribute equally to decision-making. However, this might not be accurate, as pixels near the edges in medical images may contain less information or even no information (e.g., edge pixels in x-ray images might be outside the patient's body). Therefore, we need to consider the crop's location to generate a more accurate uncertainty label.

We assume that pixels closer to the center of the image carry more information. Thus, the certainty of the crop considering location, $certainty_l$ can be defined as:

$$certainty_l = (1 - \frac{d}{s/2}) \times certainty_s, \tag{15}$$

where d is the Chebyshev distance from the pixel to the center of x.

The overall uncertainty (σ) can be calculated as:

$$\begin{aligned} \sigma =& 1 - certainty_l \\ & 1 - (1 - \frac{d}{s/2}) \times certainty_s \\ & 1 - (1 - \frac{d}{s/2}) \times (1 - \sigma_s) \\ & \frac{d}{s/2} + (1 - \frac{d}{s/2}) \times \sigma_s, \end{aligned} \tag{16}$$

which is bounded between 0 and 1.

D. Model Training and Loss Functions

Given an input image, x, and the target label y, our goal is to optimize the model $h(\cdot)$ to minimize the distance between the predicted mean (μ) of the Gaussian distribution and the target label (y) while simultaneously pushing the predicted variance (σ^2) towards the uncertainty label (y_c). This can be achieved by combining two loss functions:

$$\mathcal{L}_{NNL} = -\sum_{i=1}^{N} y_i \log(\mu_i) + (1 - y_i) \log(1 - \mu_i) \tag{17}$$

and

$$\mathcal{L}_{MAE} = \frac{1}{N} \sum_{i=1}^{N} |y_{c_i} - \sigma_i^2|, \tag{18}$$

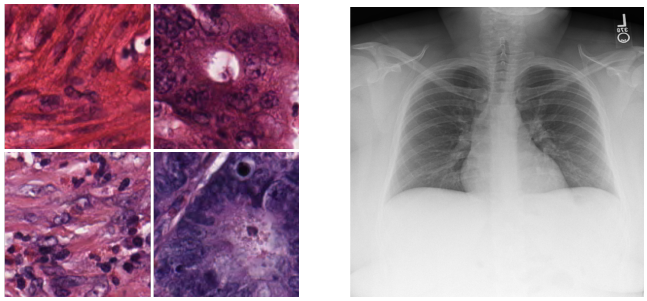
where N is the number of training samples.

The Equation 17 is the NLL loss that encourages the model to predict a mean that is centered around the ground-truth label, while Equation 19 is the mean absolute error (MAE) loss that pushes the variance towards the uncertainty label.

The overall loss can be written as:

$$\mathcal{L} = \alpha \mathcal{L}_{NLL} + \beta \mathcal{L}_{MAE}, \tag{19}$$

where α and β are two weighting scalars. By jointly minimizing NLL and MAE loss functions, we can effectively train the model to both accurately predict the target label and provide reliable uncertainty estimates.



(a) (b)

Fig. 3: Examples from the Kather (left) and Mendeley (right) datasets.

IV. Experiments

We evaluate the performance of the proposed model using publicly available medical imaging datasets for two types of images—histological images and radiology images. The former one is for RGB images, and the latter one is for gray-scale images. Thirty-two neural network models of six methods for two types of classification tasks—binary classification and multi-class classification—are trained and compared over the two datasets. We denote the six methods as follows:

- **Normal**: the normal CNN models trained with NLL;
- **TS**: the Normal model with temperature scaling;
- **Ours$_{\mathbf{R}}$**: the proposed method with the fixed α and β weighting the two components of the loss function;
- **Ours$_{\mathbf{R\text{-}TS}}$**: Ours$_{\text{R}}$ model with temperature scaling;
- **Ours$_{\mathbf{DW}}$**: the proposed method with an automatic weighting strategy for selecting α and β automatically;
- **Ours$_{\mathbf{DW\text{-}TS}}$**: applying temperature scaling to Ours$_{\text{DW}}$.

A. Experiment Setup

1) Dataset: The Kather dataset [36] contains 5000 histological images of 150×150 pixels (Figure 3a). Each image belongs to exactly one of eight tissue categories: tumor epithelium, simple stroma, complex stroma, immune cells, debris, normal mucosal glands, adipose tissue, and background (no tissue). All images are RGB, $0.495\mu m$ per pixel, digitized with an Aperio ScanScope (Aperio/Leica biosystems), magnification $20\times$. Histological samples are fully anonymized images of formalin-fixed paraffin-embedded human colorectal adenocarcinomas (primary tumors) from the Institute of Pathology, University Medical Center Mannheim, Heidelberg University, Mannheim, Germany. The dataset was randomly partitioned into training and testing datasets with a $4:1$ ratio by us.

The Mendeley dataset [37] contains both the optical coherence tomography (OCT) images of the retina and pediatric chest X-ray images. We used the pediatric chest X-ray images (Figure 3b) in this study. The dataset includes 4273 pneumonia images and 1583 normal images. We used the provided training and testing sets in this study.

2) Implementation: We implemented the proposed method using ResNet-50 [38] as the backbone that was pre-trained on ImageNet [39]. Specifically, all the convolutional (Conv) layers of the ResNet-50 model were used as the feature extractor. Two prediction heads were added on top of the feature extractor, each had a 1×1 Conv layer, a global average pooling layer, and a fully connected layer. The batch size was set as 64. The stochastic gradient descent (SGD) optimizer with a learning rate of $5e-3$ and a momentum of 0.9 was used to optimize the model parameters. The learning rate was reduced by half when the model was plateaued for 5 epochs.

The weights for both loss components were 1 (i.e., $\beta=1$ and $\alpha=1$) for Ours$_{\text{R}}$. The α and β were dynamically selected for Ours$_{\text{WD}}$ and Ours$_{\text{WDTS}}$ that balance the loss of the two loss components (i.e., NLL and MAE) to the same range.

Two random cropping strategies were used for all of our models, namely *Mild* strategy and *Radical*. The *Mild* strategy generated the crop x' as the 65% to 95% of x randomly, while *Radical* generated the x' as the 15% to 45% of x randomly.

All models were trained to converge. The best checkpoint of each model was used to evaluate the model's performance.

3) Evaluation Metrics: Six evaluation metrics were used to assess the performance of each model, namely ECE, MCE, accuracy (Acc), F1 score (F1), precision, and recall. Among the six metrics, the first two were used to measure the calibration errors, with a lower value indicating a better calibration. The rest were used to evaluate the models' classification performance, with a higher value indicating a better performance. All the evaluated models were trained twice. The average performance of the two training trials is reported in this section.

B. Evaluation Results

1) Overall Performance: Table I presents the overall performance of the six evaluated methods. The best performance for each dataset is highlighted in **bold**, the second-best in blue, and the worst in red.

The table demonstrates that our proposed method significantly outperforms the baselines in terms of calibration while also achieving substantial improvements in classification performance. For example, Ours$_{\text{R}}$ achieved ECE scores of 0.0135 and 0.0334 on the Kather and Medeley datasets, respectively, representing improvements of 39.46% and 54.31% compared to the Normal models. Though applying temperature scaling to Normal significantly improves the calibration of the model, Ours$_{\text{R-TS}}$ further enhanced calibration by 3.73% and 74.28%, respectively. Additionally, with dynamic loss weighting, Ours$_{\text{DW-TS}}$ reduced the ECE on the Kather dataset to 0.0096, surpassing both Normal and TS by 56.95% and 28.36%, respectively.

It is worth noting that our proposed method not only improves model calibration but also boosts classification performance. For instance, Ours$_{\text{R}}$ increased classification accuracy on Kather by 1.7% from 96% to 97.70% and on Medely by 3.61% from 90.80% to 94.08%.

2) Calibration Visualization: Figure 1 presents the reliability diagrams for the Normal, TS, Ours$_{\text{R}}$, and Ours$_{\text{R-TS}}$ models on the Medeley dataset. These diagrams divide the predicted

TABLE I: Detailed performance of different models on the Kather and Medeley datasets

Dataset	Model	ImageNet Pre-Train	Temperature Scaling	Uncertainty Embedding	Dynamic Weighting	ECE (↓)	MCE (↓)	Acc (↑)	F1 (↑)	Precision (↑)	Recall (↑)
Kather	**Normal**	✓				0.0223	0.4437	0.9600	0.9600	0.9601	0.9600
	TS	✓				0.0134	0.6941				
	Ours$_{R}$	✓	✓	✓		0.0135	0.5925	**0.9770**	**0.9772**	**0.9777**	**0.9770**
	Ours$_{R\text{-}TS}$	✓		✓		0.0129	0.6560				
	Ours$_{DW}$	✓		✓	✓	0.0143	**0.4352**	0.9755	0.9756	0.9747	0.9755
	Ours$_{DW\text{-}TS}$	✓	✓	✓	✓	**0.0096**	0.5122				
Medeley	**Normal**	✓				0.0731	0.4279	0.9080	0.9296	0.8941	0.9692
	TS	✓	✓			0.0552	0.3347				
	Ours$_{R}$	✓		✓		0.0334	**0.2580**	**0.9408**	**0.9513**	**0.9158**	**0.9897**
	Ours$_{R\text{-}TS}$	✓	✓	✓		**0.0142**	0.3110				
	Ours$_{DW}$	✓		✓	✓	0.0389	0.3515	0.9296	0.9457	0.9149	0.9795
	Ours$_{DW\text{-}TS}$	✓	✓	✓	✓	0.0326	0.2987				

Note: All the proposed models (i.e., Ours$_{R}$, Ours$_{R\text{-}TS}$, Ours$_{DW}$, and Ours$_{DW\text{-}TS}$) in this table used the mild cropping strategy.

TABLE II: Effect of Weighting Strategy and Cropping Strategy

Dataset	Weighting Strategy	Cropping Strategy	ECE(↓)	MCE(↓)	F1(↑)
Kather	Fixed	Mild	**0.0135**	0.5925	**0.9772**
		Radical	0.0459	**0.3728**	0.9409
	Dynamic	Mild	0.0143	0.4352	0.9756
		Radical	0.0255	0.4957	0.8687
Medeley	Fixed	Mild	**0.0334**	0.2580	**0.9513**
		Radical	0.0402	0.2745	0.9412
	Dynamic	Mild	0.0389	0.3515	0.9457
		Radical	0.0440	**0.1206**	0.8710

probabilities into bins and plot the average predicted confidence against the empirical probability (the actual proportion of positive outcomes) for each bin. An ideal calibration is indicated by points closely aligned with the diagonal line.

The larger gaps below the diagonal line in Figure 1a reveal that the Normal model is overconfident in many of its predictions. For example, the accuracy of the bin with confidence of 0.9 is only about 50%, which is significantly higher than the expected confidence, i.e., 0.5. Temperature scaling (Figure 1b) improves the Normal model's calibration but may over-compensate it, leading to underconfidence as evidenced by the gaps above the diagonal line.

The Ours$_{R}$ model (Figure 1c) demonstrates a much closer alignment with the diagonal line compared to Normal and TS, indicating superior calibration. After applying temperature scaling to our model, Ours$_{R\text{-}TS}$ (Figure 1d) further brings the bins even closer to the diagonal line. This suggests that our models exhibit significantly better calibration overall.

3) Hyperparameter Effects: The proposed method involves two hyperparameters: 1) the random crop ratio of the image and 2) the weight scalars for the loss components. To investigate the effects of the hyperparameters, we conducted an evaluation of four training strategies, combining two cropping strategies and two weighting strategies.

- Two Cropping Strategies:
  - Mild Cropping: The cropped image, x', maintains a size between 65% and 95% of the original image.
  - Radical Cropping: The cropped image, x', maintains a size between 15% and 45% of the original image.
- Two Weighting Strategies:
  - Fixed Weighting: The weights of the loss components are fixed with $\alpha = 1$ and $\beta = 1$.
  - Dynamic Weighting: The β value is automatically updated to balance the two loss components.

Table II summarizes the effects of different weighting and cropping strategies on the two datasets, with the best performance of each dataset is highlighted in **bold**, the second best performance is highlighted in blue. Among the six evaluation tasks (ECE, MCE, and F1 score on the two datasets), the combination of fixed weighting and mild cropping (`Fixed+Mild`) achieved the best performance in four cases. Although the dynamic weighting with mild cropping (`Dynamic+Mild`) did not attain the absolute best performance, it secured five second-best results. The radical cropping strategy consistently underperformed, except in terms of MCE.

These findings suggest that carefully tuning the weights for the loss components in conjunction with mild cropping (`Fixed+Mild`) can yield the best calibration and performance for a given task. However, the `Dynamic+Mild` strategy also provides acceptable results, especially when considering the reduced tuning effort and the only marginally lower classification performance (0.38% on average) compared to the `Fixed+Mild` strategy.

V. Conclusion

In this paper, we proposed a novel approach to improving model calibration in medical imaging based on probabilistic embedding. By embedding the model's predictions into a probabilistic space through Gaussian distribution, the proposed

method effectively quantifies the uncertainty associated with each prediction, leading to more calibrated, interpretable, and reliable outputs.

Our experimental results demonstrate the effectiveness of the proposed method in improving model calibration on two medical imaging tasks. We have also compared our method to existing calibration techniques and highlighted its advantages in terms of calibration and accuracy.

Future research directions of this research may include exploring different probabilistic distributions for modeling uncertainty, investigating the impact of our approach on downstream tasks, and evaluating the performance of our method for large-scale medical imaging applications.

REFERENCES

[1] J. Tang, R. M. Rangayyan, J. Xu, I. El Naqa, and Y. Yang, "Computer-aided detection and diagnosis of breast cancer with mammography: recent advances," *IEEE transactions on information technology in biomedicine*, vol. 13, no. 2, pp. 236–251, 2009.

[2] K. Doi, "Computer-aided diagnosis in medical imaging: historical review, current status and future potential," *Computerized medical imaging and graphics*, vol. 31, no. 4-5, pp. 198–211, 2007.

[3] J. Yanase and E. Triantaphyllou, "A systematic survey of computer-aided diagnosis in medicine: Past and present developments," *Expert Systems with Applications*, vol. 138, p. 112821, 2019.

[4] Z. He, "Deep learning in image classification: A survey report," in *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*. IEEE, 2020, pp. 174–177.

[5] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[6] G. Liang, S. Fouladvand, J. Zhang, M. A. Brooks, N. Jacobs, and J. Chen, "Ganai: Standardizing ct images using generative adversarial network with alternative improvement," in *2019 IEEE International Conference on Healthcare Informatics (ICHI)*. IEEE, 2019, pp. 1–11.

[7] L. Liu, P. Zhang, G. Liang, S. Xiong, J. Wang, and G. Zheng, "A spatiotemporal correlation deep learning network for brain penumbra disease," *Neurocomputing*, vol. 520, pp. 274–283, 2023.

[8] Y. Zhang, X. Wang, H. Blanton, G. Liang, X. Xing, and N. Jacobs, "2d convolutional neural networks for 3d digital breast tomosynthesis classification," in *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2019, pp. 1013–1017.

[9] G. Liang, X. Xing, L. Liu, Y. Zhang, Q. Ying, A.-L. Lin, and N. Jacobs, "Alzheimer's disease classification using 2d convolutional neural networks," in *2021 43rd annual international conference of the ieee engineering in medicine & biology society (embc)*, 2021, pp. 3008–3012.

[10] X. Xing, G. Liang, C. Wang, N. Jacobs, and A.-L. Lin, "Self-supervised learning application on covid-19 chest x-ray image classification using masked autoencoder," *Bioengineering*, vol. 10, no. 8, p. 901, 2023.

[11] I. Alsmadi, K. Ahmad, M. Nazzal, F. Alam, A. Al-Fuqaha, A. Khreishah, and A. Algosaibi, "Adversarial nlp for social network applications: Attacks, defenses, and research directions," *IEEE Transactions on Computational Social Systems*, vol. 10, no. 6, pp. 3089–3108, 2022.

[12] J. Zulu, B. Han, I. Alsmadi, and G. Liang, "Enhancing machine learning based sql injection detection using contextualized word embedding," in *Proceedings of the 2024 ACM Southeast Conference*, 2024, pp. 211–216.

[13] G. Liang, J. Guerrero, F. Zheng, and I. Alsmadi, "Enhancing neural text detector robustness with μ attacking and rr-training," *Electronics*, vol. 12, no. 8, p. 1948, 2023.

[14] W. Song, S. Workman, A. Hadzic, X. Zhang, E. Green, M. Chen, R. Souleyrette, and N. Jacobs, "Farsa: Fully automated roadway safety assessment," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 521–529.

[15] M. Chen, A. Hadzic, W. Song, and N. Jacobs, "Applications of deep machine learning to highway safety and usage assessment," in *Transportation Research Board Workshop (Sponsored by AED50)*, Jan. 2021.

[16] G. Liang, J. Zulu, X. Xing, and N. Jacobs, "Unveiling roadway hazards: Enhancing fatal crash risk estimation through multiscale satellite imagery and self-supervised cross-matching," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 17, pp. 535–546, 2024.

[17] J. ZuHone, D. Barnes, N. Jacobs, W. Forman, P. Nulsen, R. Kraft *et al.*, "A deep learning view of the census of galaxy clusters in illustristng," *Monthly Notices of the Royal Astronomical Society*, vol. 498, no. 4, pp. 5620–5628, 2020.

[18] Y. Zhang, G. Liang, Y. Su, and N. Jacobs, "Multi-branch attention networks for classifying galaxy clusters," in *the 25th International Conference on Pattern Recognition*. IEEE, 2021, pp. 9643–9649.

[19] S.-C. Lin, Y. Su, G. Liang, Y. Zhang, N. Jacobs, and Y. Zhang, "Estimating cluster masses from sdss multiband images with transfer learning," *Monthly Notices of the Royal Astronomical Society*, vol. 512, no. 3, pp. 3885–3894, 2022.

[20] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International conference on machine learning*. PMLR, 2017, pp. 1321–1330.

[21] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton, "Regularizing neural networks by penalizing confident output distributions," *arXiv:1701.06548*, 2017.

[22] A. Kumar, S. Sarawagi, and U. Jain, "Trainable calibration measures for neural networks from kernel mean embeddings," in *Prco. ICML*, 2018, pp. 2810–2819.

[23] T. Popordanoska, R. Sayer, and M. Blaschko, "A consistent and differentiable lp canonical calibration error estimator," *Advances in Neural Information Processing Systems*, vol. 35, pp. 7933–7946, 2022.

[24] X. Jiang, M. Osl, J. Kim, and L. Ohno-Machado, "Calibrating predictive model estimates to support personalized medicine," *Journal of the American Medical Informatics Association*, vol. 19, no. 2, pp. 263–274, 2012.

[25] G. Liang, Y. Zhang, X. Wang, and N. Jacobs, "Improved trainable calibration method for neural networks on medical imaging classification," in *British Machine Vision Conference (BMVC)*, 2020.

[26] M. Kull, M. Perello Nieto, M. Kängsepp, T. Silva Filho, H. Song, and P. Flach, "Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration," *Advances in neural information processing systems*, vol. 32, 2019.

[27] M. P. Naeini, G. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using bayesian binning," in *Proc. AAAI*, 2015.

[28] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv:1503.02531*, 2015.

[29] D. Widmann, F. Lindsten, and D. Zachariah, "Calibration tests in multi-class classification: A unifying framework," in *Proc. NeurIPS*, 2019, pp. 12 236–12 246.

[30] A. Gretton, "Introduction to rkhs, and some simple kernel algorithms," *Adv. Top. Mach. Learn. Lecture Conducted from University College London*, 2013.

[31] L. Song, "Learning via hilbert space embedding of distributions," 2008.

[32] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. CVPR*, June 2016.

[33] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?" in *Proc. NeurIPS*, 2019, pp. 4694–4703.

[34] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Proc. ICLR*, 2018.

[35] S. Thulasidasan, G. Chennupati, J. A. Bilmes, T. Bhattacharya, and S. Michalak, "On mixup training: Improved calibration and predictive uncertainty for deep neural networks," in *Proc. NeurIPS*, 2019, pp. 13 888–13 899.

[36] J. N. Kather *et al.*, "Multi-class texture analysis in colorectal cancer histology," *Scientific reports*, vol. 6, p. 27988, 2016.

[37] D. Kermany and M. Goldbaum, "Labeled optical coherence tomography (oct) and chest x-ray images for classification," *Mendeley Data*, vol. 2, 2018.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.