Dual-Supervised Uncertainty Inference of MoGMM-FC Layer for Image Recognition

Jiyang Xie, *Student Member, IEEE,* Zhanyu Ma, *Senior Member, IEEE,* Jing-Hao Xue, *Member, IEEE,* Guoqiang Zhang, *Member, IEEE,* Jian Sun, Yinhe Zheng, and Jun Guo

***Abstract*—This paper proposes a dual-supervised uncertainty inference (DS-UI) framework for improving Bayesian estimation-based UI in DNN-based image recognition. In the DS-UI, we combine the classifier of a DNN, *i.e.*, the last fully-connected (FC) layer, with a mixture of Gaussian mixture models (MoGMM) to obtain an MoGMM-FC layer. Unlike existing UI methods for DNNs, which only calculate the means or modes of the DNN outputs' distributions, the proposed MoGMM-FC layer acts as a probabilistic interpreter for the features that are inputs of the classifier to directly calculate the probabilities of them for the DS-UI. In addition, we propose a dual-supervised stochastic gradient-based variational Bayes (DS-SGVB) algorithm for the MoGMM-FC layer optimization. Unlike conventional SGVB and optimization algorithms in other UI methods, the DS-SGVB not only models the samples in the specific class for each Gaussian mixture model (GMM) in the MoGMM, but also considers the negative samples from other classes for the GMM to reduce the intra-class distances and enlarge the inter-class margins simultaneously for enhancing the learning ability of the MoGMM-FC layer in the DS-UI. Experimental results show the DS-UI outperforms the state-of-the-art UI methods in misclassification detection. We further evaluate the DS-UI in open-set out-of-domain/-distribution detection and find statistically significant improvements. Visualizations of the feature spaces demonstrate the superiority of the DS-UI.**

***Index Terms*—Deep Learning, Image Recognition, Uncertainty Inference, Dual Supervised Framework, Mixture of Gaussian Mixture Models**

I. Introduction

In recent years, deep neural networks (DNNs) have achieved significant improvement in image recognition and other research fields [1], [2], [3], [4], [5], [6], [7]. However, robust image recognition is still challenging, as DNNs tend to output certain and even overconfident predictions, but without confidence intervals of the predictions [8], [9], and thus unable to assess the uncertainty of their outputs and detect abnormal samples. To address this issue, uncertainty inference (UI), containing misclassification and out-of-domain/-distribution detections (as shown in Figure 1), has been introduced for estimating how uncertain the outputs of a DNN are to further improve its reliability and applicability.

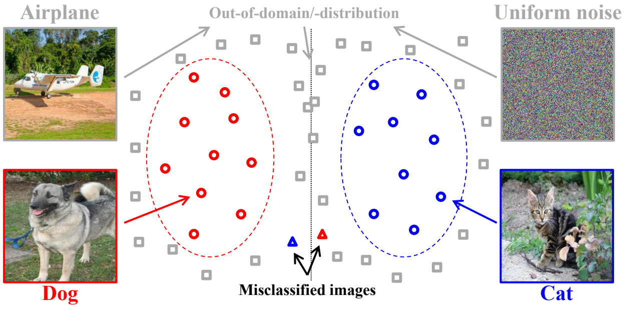


Fig. 1: Illustration of uncertainty inference. Taking a "dog (red circles) versus cat (blue circles)" image recognition task as an example, abnormal samples contains misclassified images (triangles) and out-of-domain/-distribution images (grey squares). The out-of-domain samples include other unknown classes (*e.g.*, airplane in this case) and the out-of-distribution samples include noisy data.

Recent works [9], [10], [11], [12], [13], [14] on the UI intended to model the outputs of DNNs by distributions, such as Gaussian [10], [12], [14], Dirichlet [9], [13], and softmax [11] distributions, and define uncertainty only on the outputs. In practice, they only calculate the means or modes of the distributions, although in different ways, for the UI.

In this paper, a dual-supervised uncertainty inference (DS-UI) framework is introduced for improving Bayesian estimation-based UI. In the DS-UI, we propose to combine a mixture of Gaussian mixture models (MoGMM) with a fully-connected (FC) layer into an MoGMM-FC layer to replace the classifier of a DNN and calculate probabilities of the outputs directly. The probabilities can be transformed to the representations of confidence or uncertainty. In general, a DNN architecture for image recognition can be divided into two cascaded parts, *i.e.*, a feature extractor that contains multiple convolutional and/or FC layers, and a classifier (the last FC layer), as shown in Figure 2(a) [15], [16]. In Figure 2(b), the output for one class is proportional to the projection of the extracted feature in the direction of the class center. As the Gaussian distribution is a simple and generic distribution, and a mixture of mixture models [17], [18] can better estimate large intra-class variability in complex scenes, we adopt an MoGMM to model both the intra-class variability and the inter-class difference, and accordingly extend the DNN model to a new model with the proposed MoGMM-FC layer for modeling the features *w.r.t.* the class centers (*i.e.*, the row

J. Xie, Z. Ma, and J. Guo are with the Pattern Recognition and Intelligent Systems Lab., Beijing University of Posts and Telecommunications, China. E-mail: {xiejiyang2013, mazhanyu, guojun}@bupt.edu.cn

J. -H. Xue is with the Department of Statistical Science, University College London, United Kingdom. E-mail: jinghao.xue@ucl.ac.uk

G. Zhang is with the School of Electrical and Data Engineering, University of Technology Sydney, Australia. E-mail: guoqiang.zhang@uts.edu.au

J. Sun, E-mail: jiansun_china@hotmail.com

Y. Zheng, E-mail: zhengyinhe1@163.com

(Corresponding author: Zhanyu Ma)

vectors of the parameter matrix of the classifier) and enhancing the learning ability of the classifier. In the MoGMM-FC layer, each Gaussian mixture model (GMM) is learned for one class, which is a common setting in probabilistic model-based image recognition [19], [20]. Each class center is shared with the weighted summation of the means of the components in a associated GMM and optimized with the MoGMM.

Moreover, traditional stochastic gradient-based variational Bayes (SGVB) algorithms generally supervise the optimization of mixture models by using only positive samples of each class and aim at reducing the distances between samples and their corresponding class centers [19], [20]. However, the margin between different classes might be undesirably compressed (see Figure 2(d)), as also found in the optimizations of other UI methods. In this paper, we propose to improve the SGVB for the DS-UI by comprehensively considering both the positive samples (in the class) and the negative samples (in other classes) for each GMM, a strategy defined hereafter as dual-supervised optimization, to reduce the intra-class distances and enlarge the inter-class margins simultaneously, as shown in Figure 2(c).

The contributions of this paper are four-fold:

- A new DS-UI framework is introduced for image recognition. We propose an MoGMM-FC layer with a parameter-shared and jointly-optimized MoGMM to act as a probabilistic interpreter for the features of DNNs to calculate probabilities for the DS-UI directly.
- We propose a dual-supervised SGVB (DS-SGVB) for the MoGMM-FC layer optimization. The DS-SGVB can enhance the learning ability of the MoGMM-FC layer for the DS-UI.
- The proposed DS-UI outperforms the state-of-the-art UI methods in the misclassification detection. Statistically significant improvement compared with the referred UI methods can be found.
- We extend the evaluation of the DS-UI to open-set out-of-domain/-distribution detection (detecting unknown samples from unknown classes or noisy samples) and find statistically significant improvements.

II. Related Work

A. Uncertainty Inference

Most of the recently proposed UI methods define uncertainty only on the outputs and calculate the means or modes of the DNN outputs' distributions.

Blundell *et al.* [21] proposed a backpropagation-compatible algorithm with unbiased Monte Carlo (MC) gradients for estimating parameter uncertainty of a DNN, called Bayesian by backpropagation (BBP). MC dropout [10] utilized the standard dropout [22] as an MC sampler to study the dropout uncertainty properties. Variance of different output values with the same input and different dropout masks were used for presenting the uncertainty of the output. Li at al. [23] gave theoretical properties on asymptotic convergence and predictive risk of stochastic gradient descent (SGD), and sampled the posterior by combining adaptive preconditioners with stochastic gradient Langevin dynamics (SGLD), shorten as p-SGLD. Lakshminarayanan *et al.* [8] proposed a deep ensemble method to yield predictive uncertainty estimations, which can be considered as an Markov chain Monte Carlo (MCMC)-based alternative to Bayesian neural networks. However, as the aforementioned MC sampling-based methods cannot satisfy the requirement of inference speed [15], more explicit distributional assumption-based methods have been proposed in recent years to address this problem.

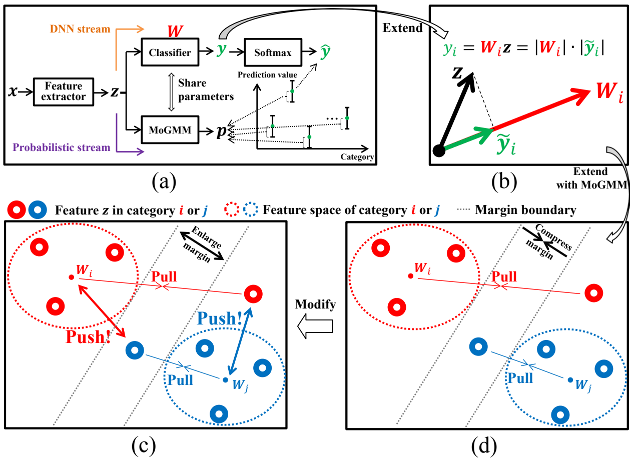


Fig. 2: Illustration of the DS-UI. A conventional DNN architecture for image recognition with cross-entropy loss (orange stream in (a)) can be divided into a feature extractor and a classifier (*i.e.*, the last FC layer) with parameter matrix $\boldsymbol{W}$. Here, $\boldsymbol{W}_i$, the i^{th} row vector of $\boldsymbol{W}$, is assumed to be the center of the i^{th} class. Given a feature vector $\boldsymbol{z}$, the output value y_i of the classifier is proportional to the projection length $|\tilde{\boldsymbol{y}}_i|$ of $\boldsymbol{z}$ in the direction of $\boldsymbol{W}_i$ in (b). We extend the DNN to a model with a parameter-shared MoGMM by adding a probabilistic stream to model $\boldsymbol{z}$ *w.r.t.* the class centers (purple stream in (a)) and obtain uncertainty representations by $\boldsymbol{p}$. Traditional SGVB for mixture models and optimization algorithms in other UI methods aim at decreasing the distances between each sample and its corresponding class center, which may undesirably compress the margin between two classes in the case in (d). We modify it by "pulling" positive samples and "pushing" negative samples simultaneously for the class centers (in (c)) to reduce intra-class distances and enlarge inter-class margins simultaneously by proposing a dual-supervised SGVB, which benefits the DS-UI.

Lee *et al.* [24] introduced a simple yet effective method for detecting any abnormal samples, including out-of-distribution samples and adversarial attacks. The method applies the class conditional Gaussian distributions *w.r.t.* shallower- and deeper-features of the DNNs under Gaussian discriminant analysis and obtains confidence scores with Mahalanobis distances. Maddox *et al.* [12] proposed stochastic weight averaging Gaussian (SWAG) method, which estimates a Gaussian distribution for the first-order moment of the SGD iterations as an approximate posterior distribution over DNN parameters and

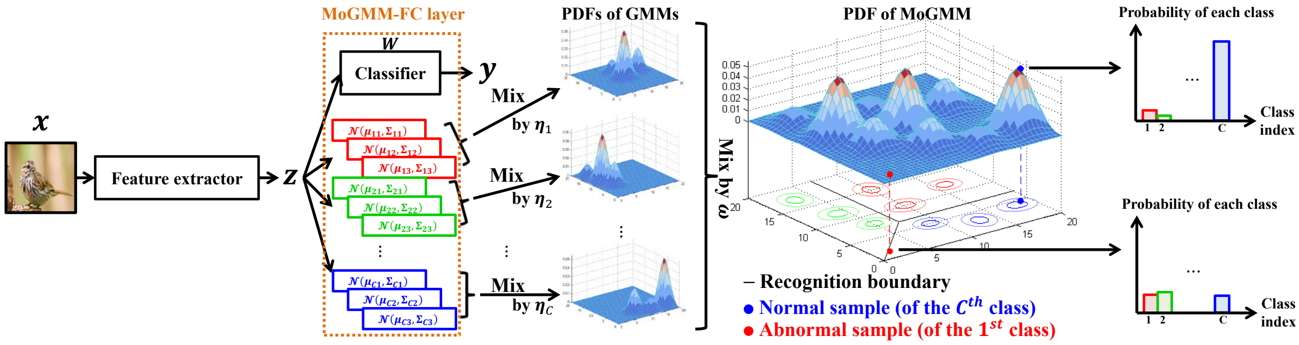

$\boldsymbol{x}$
Feature extractor
$\boldsymbol{z}$
MoGMM-FC layer
$\boldsymbol{W}$
Classifier
$\boldsymbol{y}$
$\mathcal{N}(\mu_{11},\Sigma_{11})$
$\mathcal{N}(\mu_{12},\Sigma_{12})$
$\mathcal{N}(\mu_{13},\Sigma_{13})$
$\mathcal{N}(\mu_{21},\Sigma_{21})$
$\mathcal{N}(\mu_{22},\Sigma_{22})$
$\mathcal{N}(\mu_{23},\Sigma_{23})$
$\mathcal{N}(\mu_{C1},\Sigma_{C1})$
$\mathcal{N}(\mu_{C2},\Sigma_{C2})$
$\mathcal{N}(\mu_{C3},\Sigma_{C3})$
Mix by $\boldsymbol{\eta}_1$
Mix by $\boldsymbol{\eta}_2$
Mix by $\boldsymbol{\eta}_C$
PDFs of GMMs
Mix by $\boldsymbol{\omega}$
PDF of MoGMM
– Recognition boundary
● Normal sample (of the C^{th} class)
● Abnormal sample (of the 1^{st} class)
Probability of each class
Class index
1 2 C
Probability of each class
Class index
1 2 C

Fig. 3: Structure of the MoGMM-FC layer. The MoGMM is paralleled with the classifier (*i.e.*, the final FC layer). Both of them are cascaded after the feature extractor. In the MoGMM-FC layer, C GMMs (one GMM for each class) are mixed and their output probabilities are used for the UI. Here, we take three Gaussian components ($K = 3$) for each GMM as an example. According to the PDF of the MoGMM, which mixes those of all the GMMs, abnormal samples, including misclassified and out-of-domain/-distribution ones, can be easily detected and distinguished from normal samples, as the probabilities of an abnormal sample belonging to individual classes are all small (bottom-right), unlike the pattern of a normal sample (top-right). Thus, the probabilities can be transformed to the representations of confidence or uncertainty.

samples from this Gaussian distribution to perform Bayesian model averaging and UI. Distributional estimation was utilized for learning feature and uncertainty simultaneously in face recognition [14].

Among UI methods based on explicit distributional assumptions, Hendrycks and Gimpel [11] introduced a baseline model with cross-entropy (CE) loss training that assumes the outputs following softmax distributions and detects the misclassified or the out-of-distribution samples with maximum softmax probabilities in multiple tasks. In [9], the authors presented Dirichlet prior network (DPN) using Kullback-Leibler (KL) divergence loss from the Dirichlet distributions of the smooth labels to those of model outputs to introduce Dirichlet distributions into the DNNs for modeling the uncertainty, especially distributional uncertainty (*i.e.*, dataset shift). Following the DPN, a reverse Kullback-Leibler (RKL) divergence loss between the aforementioned Dirichlet distributions was introduced for prior network training to improve the UI and adversarial robustness [13]. Posch and Pilz [25] trained DNNs using Bayesian techniques which allow an easy evaluation of model uncertainty. Counterfactual latent uncertainty explanations (CLUE) [26] was proposed for interpreting the uncertainty estimates from differentiable probabilistic models. The above methods calculate only the means or the modes of the predictions for the UI.

In addition to the above conventional methods, Guo *et al.* [27] proposed three scaling methods, namely temperature, vector, and matrix scaling methods, for prediction calibrations on DNN architectures with different depth and width. The authors found that the temperature scaling is surprisingly effective at calibrating predictions. An inhibited softmax function, which extends the softmax layer with an additional constant input, was proposed for uncertainty estimation in DNNs with softmax outputs [28]. Neural stochastic differential equation (SDE) network (SDE-Net) in [15], a non-Bayesian method, brought the concept of SDE into the UI. The SDE-Net contains a drift net that controls the system to fit the predictive function and a diffusion net that captures the model uncertainty. Van Amersfoort *et al.* [29] proposed deterministic uncertainty quantification (DUQ) based on radial basis function (RBF) networks to enforce detectability of changes in the input using a gradient penalty and reliably detect out of distribution data. A multi-input multi-output (MIMO) framework [30] was proposed that independently trains the input and the output layers for each class and shares hidden layers in order to implement a partially weight-sharing ensemble model for robust prediction and uncertainty estimation.

The aforementioned methods only consider the positive samples for the corresponding class centers and jointly train with both the in-domain samples and the out-of-domain samples, which is a close-set out-of-domain detection. They mainly applied the CE loss function for training, except for some methods adapting the KL and the RKL.

B. *Mixture Models and SGVB*

Several works [31], [32], [33] have incorporated GMMs into the DNNs but not for the UI. Variani *et al.* [32] first proposed a GMM layer, which is jointly optimized within a DNN using asynchronous stochastic gradient descent (ASGD). The GMM layer is cascaded after the DNN and models deep features extracted from it. In [33], an unsupervised deep learning framework was proposed to combine deep representations and GMM-based deep modeling. Later on, a temporal Gaussian mixture (TGM) layer was introduced for capturing longer-term temporal information in videos [31]. However, no mixtures of mixture models have been explored for jointly modeling the outputs of DNNs or estimating uncertainty.

In addition to some of the aforementioned works, which introduced their own optimization algorithms, various SGVB algorithms [34], [35], [36], [37], [38] have been proposed in recent years for probabilistic model optimization. However, these SGVB algorithms do not consider the negative samples in other classes for the mixture model of a class.

III. Dual-supervised Uncertainty Inference

As the UI usually requires stronger learning ability than other tasks, it is desirable to comprehensively consider both the positive and the negative samples for optimization of each class during training. In this section, we introduce the so-called dual-supervised uncertainty inference (DS-UI) framework to achieve this goal.

Although the classifier of a DNN, commonly the top FC layer, can model the correlation to describe the membership of a feature $\boldsymbol{z}$ belonging to a class, it cannot obtain the uncertainty of $\boldsymbol{z}$ directly. To this end, we propose an MoGMM-FC layer, which can be treated as a probabilistic interpreter for modeling $\boldsymbol{z}$, as shown in Figure 3. For each GMM in the MoGMM, we propose a dual-supervised SGVB (DS-SGVB) algorithm, which *not only* models the positive samples in the class as the conventional SGVB and the optimization algorithms in other UI methods, *but also* considers the negative samples from other classes. The DS-SGVB can enhance the learning ability of the MoGMM and improve the UI performance by reducing the intra-class distances and enlarging the inter-class margins simultaneously.

A. MoGMM-FC Layer

We propose to use an MoGMM to model the extracted feature vector $\boldsymbol{z} \in R^{M\times 1}$, where M is the dimension of $\boldsymbol{z}$. Assuming a recognition task with C classes, we assign a GMM in the MoGMM to each class. The probability density function (PDF) of the MoGMM is defined as

$$\text{MoGMM}(\boldsymbol{z};\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\eta},\boldsymbol{\omega}) = \sum_{i=1}^{C}\omega_i \underbrace{\sum_{j=1}^{K}\eta_{ij}\mathcal{N}(\boldsymbol{z};\boldsymbol{\mu}_{ij},\boldsymbol{\Sigma}_{ij})}_{\text{GMM}_i(\boldsymbol{z})}, \quad (1)$$

with Gaussian distributions $\mathcal{N}(\boldsymbol{z};\boldsymbol{\mu}_{ij},\boldsymbol{\Sigma}_{ij})$, where K is the number of components in each GMM and $\boldsymbol{\mu} = \{\boldsymbol{\mu}_{ij}\}$, $\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma}_{ij}\}$, and $\boldsymbol{\eta} = \{\eta_{ij}\}$ are the parameter sets of means, covariances, and mixing weights, respectively. $\boldsymbol{\mu}_{ij}$ ($M \times 1$ dimensions), $\boldsymbol{\Sigma}_{ij}$ ($M \times M$ dimensions), and η_{ij} are means, covariances, and mixing weight of the j^{th} Gaussian component in the i^{th} GMM. For high-dimensional $\boldsymbol{z}$ in practice, $\boldsymbol{\Sigma}_{ij}$ can be defined as a non-singular diagonal matrix for simplicity, which is employed in this paper. Meanwhile, $\boldsymbol{\omega} = [\omega_1, \cdots, \omega_C]^{\text{T}}$ contains C nonnegative mixing weights of the C GMMs and $\sum_{i=1}^{C}\omega_i = 1$. In the recognition task, ω_i can be roughly estimated by the proportions of each class in the training set beforehand [39].

The distribution of μ_{ijm} is defined as a Gaussian distribution with mean a_{ijm} and variance b_{ijm}, where a_{ijm} and b_{ijm} are elements of their corresponding hyperparameter sets $\boldsymbol{A} = \{a_{ijm}\}$ and $\boldsymbol{B} = \{b_{ijm}\}$, respectively. Meanwhile, Σ_{ijmm} follows a Dirac delta distribution $\delta(b_{ijm})$ where the value of the PDF is equal to one if $\Sigma_{ijmm} = b_{ijm}$, zero otherwise. We define the parameter set of the MoGMM as $\boldsymbol{\Phi} = \{\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{V}\}$, where the latent variable matrix $\boldsymbol{V}$ is a $C\times K$-dimensional matrix and each row $\boldsymbol{v}_i$ is a one-hot vector following $p(v_{ij} = 1) = \eta_{ij}$, and the hyperparameter set as $\boldsymbol{\theta} = \{\boldsymbol{A},\boldsymbol{B},\boldsymbol{\eta}\}$ for optimization.

Here, the mean parameters in $\boldsymbol{\mu}$ are shared with the classifier. As each row $\boldsymbol{W}_i$ of the parameter matrix $\boldsymbol{W}$ of the classifier is described as a class center, we introduce an approximation of $\boldsymbol{W}$ by $\boldsymbol{\mu}$ to align their dimensions. For the i^{th} GMM (representing the i^{th} class) in the MoGMM, the mean $\boldsymbol{W}_i$ of the whole GMM can be calculated as $\boldsymbol{W}_i = \sum_{j=1}^{K}\eta_{ij}\boldsymbol{\mu}_{ij}$ (Please find the derivation in Section III-A1). We define that $\boldsymbol{y} = [y_1, \cdots, y_C]^{\text{T}}$ is the output vector of the classifier. Thus, the i^{th} output y_i of the classifier for the i^{th} class can be determined by

$$y_i = \sum_{j=1}^{K}\eta_{ij}\boldsymbol{\mu}_{ij}^{\text{T}}\boldsymbol{z}, \quad (2)$$

assuming the bias vector of the classifier is removed.

As $\boldsymbol{\eta}_i = [\eta_{i1}, \cdots, \eta_{iK}]^{\text{T}}$ is normalized, which is a hard regularization in stochastic gradient-based optimization, we define an alternative $\tilde{\boldsymbol{\eta}}_i \in R^{K\times 1}$ to implicitly optimize $\boldsymbol{\eta}_i$ by $\boldsymbol{\eta}_i = \text{softmax}(\tilde{\boldsymbol{\eta}}_i)$. Similarly, an alternative $\tilde{b}_{ijm}$ is introduced for the positive b_{ijm} by $b_{ijm} = e^{\tilde{b}_{ijm}}$.

1) Derivation of the Mean of A GMM: The mean of the i^{th} GMM, $\text{GMM}_i(\boldsymbol{z})$, in the MoGMM can be obtained by

$$\begin{aligned} mean\,(\text{GMM}_i(\boldsymbol{z})) &= \int \boldsymbol{z}\sum_{j=1}^{K}\eta_{ij}\mathcal{N}(\boldsymbol{z};\boldsymbol{\mu}_{ij},\boldsymbol{\Sigma}_{ij})d\,\boldsymbol{z} \\ &= \sum_{j=1}^{K}\eta_{ij}\int \boldsymbol{z}\mathcal{N}(\boldsymbol{z};\boldsymbol{\mu}_{ij},\boldsymbol{\Sigma}_{ij})d\,\boldsymbol{z} \\ &= \sum_{j=1}^{K}\eta_{ij}\boldsymbol{\mu}_{ij}. \end{aligned} \quad (3)$$

B. Optimization for the MoGMM-FC Layer

1) Conventional SGVB: In variational inference (VI), the common approach [40] is to optimize the hyperparameters of a probability model by maximizing the lower bound $L(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi});\boldsymbol{D})$ with the approximated posterior distribution $q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})$, where $\boldsymbol{D} = \{\boldsymbol{Z},\boldsymbol{T}\}$ is the dataset, $\boldsymbol{Z} = \{\boldsymbol{z}_i\}_{i=1}^{N}$ and $\boldsymbol{T} = \{t_i\}_{i=1}^{N}$ are the inputs and labels, respectively, and N is the number of samples in $\boldsymbol{D}$. $L(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi});\boldsymbol{D})$, which can be considered as the negative KL divergence from $q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})$ to the joint distribution $p(\boldsymbol{D},\boldsymbol{\Phi})$= $p(\boldsymbol{D}|\boldsymbol{\Phi})p(\boldsymbol{\Phi})$ (where $p(\boldsymbol{D}|\boldsymbol{\Phi})$ is likelihood and $p(\boldsymbol{\Phi})$ is prior distribution) is defined as

$$\begin{aligned} L(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi});\boldsymbol{D}) &= \int q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})\ln\frac{p(\boldsymbol{D},\boldsymbol{\Phi})}{q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})}d\boldsymbol{\Phi} \\ &= \underbrace{\int q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})\ln p(\boldsymbol{D}|\boldsymbol{\Phi})d\boldsymbol{\Phi}}_{L_D(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))} - \underbrace{\int q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})\ln\frac{q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})}{p(\boldsymbol{\Phi})}d\boldsymbol{\Phi}}_{D_{\text{KL}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})||p(\boldsymbol{\Phi}))}, \end{aligned} \quad (4)$$

where the first term $L_D(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$ is the expected log-likelihood and the second term $D_{\text{KL}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})||p(\boldsymbol{\Phi}))$ is the KL divergence from $q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})$ to the prior distribution $p(\boldsymbol{\Phi})$.

For the SGVB algorithm, we usually approximate the expected log-likelihood $L_D(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$ by

$$\begin{aligned} L_D(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})) &= \frac{1}{N} \sum_{\boldsymbol{z}\in\boldsymbol{Z}, t\in\boldsymbol{T}} \mathrm{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})} \left[\ln p(\boldsymbol{z}, t|\boldsymbol{\Phi})\right] \\ &\approx L_D^{\mathrm{SGVB}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})) \\ &= \frac{1}{B} \sum_{b=1}^{B} \ln\left(\omega_{t_b} \mathrm{GMM}_{t_b}(\boldsymbol{z}_b)\right), \end{aligned} \tag{5}$$

where B is batch size and t_b is the label of the b^{th} sample $\boldsymbol{z}_b$. To be able to use the SGVB, the next step is to consider optimizing $\{-L_D^{\mathrm{SGVB}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))+\gamma D_{\mathrm{KL}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})||p(\boldsymbol{\Phi}))\}$ with nonnegative multiplier γ, where the KL divergence is seen as a regularization term. Note that the previous methods [10], [40] are computationally expensive by making use of the MC estimation approaches. We propose to derive a generalized form for the KL divergence in Section III-B3 as a regularization term to constrain the hyperparameters in $\boldsymbol{\theta}$.

Note that although the closed-form solution of the MoGMM optimization under the VI can be found, it is infeasible to be extended to an SGVB solution, which makes it difficult to jointly optimize the MoGMM together with the classifier.

2) Dual-supervised SGVB: In this section, we propose the DS-SGVB algorithm to reduce the intra-class distances and enlarge the inter-class margins simultaneously. Recall that the approximated expected log-likelihood in (5) undertakes "pull" operation between the class centers and their corresponding positive samples, we define a dual-supervised expected log-likelihood $L_D^{\mathrm{DS}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$ as

$$L_D^{\mathrm{DS}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})) = L_D^{\mathrm{SGVB}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})) - \rho L_D^{\mathrm{NSGVB}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})), \tag{6}$$

where ρ is a nonnegative multiplier, $L_D^{\mathrm{SGVB}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$ is given by (5), and $L_D^{\mathrm{NSGVB}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$ is the negative-sample expected log-likelihood as

$$L_D^{\mathrm{NSGVB}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})) = \frac{1}{B} \sum_{b=1}^{B} \sum_{i\neq t_b} \ln\left(\omega_i \mathrm{GMM}_i(\boldsymbol{z}_b)\right), \tag{7}$$

which minimizes the log-likelihood of each GMM *w.r.t.* negative samples and undertakes "push" operation between the class centers and the negative samples belonging to other classes. By minimizing $-L_D^{\mathrm{DS}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$, the learning ability of the MoGMM can be further enhanced, as it not only models the positive samples in the class for a GMM as the conventional SGVB, but also considers the negative samples from other classes.

3) Generalized Form of $D_{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})||p(\boldsymbol{\Phi}))$: In this section, a regularization term $\mathrm{Reg}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$, which is related to $D_{\mathrm{KL}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})||p(\boldsymbol{\Phi}))$ and performs as a generalized form of it, is applied to constrain the hyperparameters in $\boldsymbol{\theta}$ of the MoGMM.

Definition 1. *Let the prior distributions of μ_{ijm}, Σ_{ijmm}, and $\boldsymbol{v}_i$ be standard normal distribution, uniform distribution in the interval of $(0, \infty)$ and categorical distribution with equal probabilities, respectively. The generalized form Reg$(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$ of $D_{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})||p(\boldsymbol{\Phi}))$ is defined to be*

$$\begin{aligned} &\mathrm{Reg}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})) \\ &= \sum_{i=1}^{C} \omega_i^* \sum_{j=1}^{K} \eta_{ij}^* \sum_{m=1}^{M} D_{\mathrm{KL}}(q(\mu_{ijm}|a_{ijm}, b_{ijm})||p(\mu_{ijm})) \\ &\quad + \sum_{i=1}^{C} \omega_i^* \sum_{j=1}^{K} \eta_{ij}^* \sum_{m=1}^{M} D_{\mathrm{KL}}(q(\Sigma_{ijmm}|b_{ijm})||p(\Sigma_{ijmm})) \\ &\quad + \sum_{i=1}^{C} \omega_i^* D_{\mathrm{KL}}(q(\boldsymbol{v}_i|\boldsymbol{\eta}_i)||p(\boldsymbol{v}_i)) \\ &= \sum_{i=1}^{C} \omega_i^* \left\{ \sum_{j=1}^{K} \left[\eta_{ij} \ln(\eta_{ij} \cdot K) \right.\right. \\ &\quad \left.\left. + \frac{\eta_{ij}^*}{2} \sum_{m=1}^{M} \left(b_{ijm} + a_{ijm}^2 - \ln b_{ijm} - 1\right) \right] \right\}, \end{aligned} \tag{8}$$

where $\boldsymbol{\Sigma}_{ij}$ is assumed to be a non-singular diagonal matrix. ω_i^* and η_{ij}^* are nonnegative sub-multipliers and set equal to ω_i and η_{ij}, respectively, in this paper. Note that Reg$(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$ is equivalent to the original $D_{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})||p(\boldsymbol{\Phi}))$ when ω_i^* and η_{ij}^* have equal values for different i or/and j, respectively.

Derivation of $\mathrm{Reg}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$ can be found in Section III-B4. Here, we discuss the motivation and advantages of $\mathrm{Reg}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$ in (8). Firstly, the KL divergence in (4) is commonly utilized as regularization terms and loss functions, *e.g.*, in [9], [13]. Furthermore, using $\mathrm{Reg}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$ can be seen as adaptive hyperparameter (*i.e.*, multipliers in the loss) optimization of the KL divergence terms for different parameters. This can avoid not only the rigid constraint of equal multipliers, but also manual hyperparameter tuning. In the experiments discussed in Section IV, the learned η_{ij}^* vary in $[0.05, 0.25]$ instead of being equal, which confirms $\mathrm{Reg}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$ has generalized learning capacity. In addition, the sub-multipliers in $\mathrm{Reg}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$ can also reflect the contributions of the KL divergences of μ_{ijm} and Σ_{ijmm} in the MoGMM optimization.

In the end, the total loss function $\mathcal{L}$, which is used during training, is defined as

$$\mathcal{L} = L_{\mathrm{CE}} - L_D^{\mathrm{DS}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})) + \gamma \mathrm{Reg}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})), \tag{9}$$

where L_{CE} is the cross-entropy (CE) loss for the classifier in Figure 3 and γ is a nonnegative multiplier. We should note that this loss function is an extension of the lower bound in variational Bayes and the Bayesian optimization is carried out by a gradient-based method.

4) Derivation of the Regularization Term Reg$(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$: We start with the derivation of the Kullback-Leibler (KL) divergence $D_{\mathrm{KL}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})||p(\boldsymbol{\Phi}))$. As we define the parameter set of the mixture of Gaussian mixture models (MoGMM) as $\boldsymbol{\Phi} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{V}\}$ and the hyperparameter set of it as $\boldsymbol{\theta} = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\eta}\}$ for optimization, the approximated posterior distribution $q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})$ can be factorized as

$$\begin{aligned} q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}) =& q(\boldsymbol{\mu}|\boldsymbol{A}, \boldsymbol{B}) \cdot q(\boldsymbol{\Sigma}|\boldsymbol{B}) \cdot q(\boldsymbol{V}|\boldsymbol{\eta}) \\ =& \prod_{i=1}^{C} \prod_{j=1}^{K} \prod_{m=1}^{M} q(\mu_{ijm}|a_{ijm}, b_{ijm}) \\ & \cdot \prod_{i=1}^{C} \prod_{j=1}^{K} \prod_{m=1}^{M} q(\Sigma_{ijmm}|b_{ijm}) \cdot \prod_{i=1}^{C} q(\boldsymbol{v}_i|\boldsymbol{\eta}_i), \end{aligned} \tag{10}$$

where $\boldsymbol{\Sigma}_{ij}$ is assumed to be a non-singular diagonal matrix, and

$$q(\mu_{ijm}|a_{ijm}, b_{ijm}) = \mathcal{N}(a_{ijm}, b_{ijm}), \tag{11}$$

$$\begin{aligned} q(\Sigma_{ijmm}|b_{ijm}) &= \delta(b_{ijm}) \\ &= \begin{cases} 1, \Sigma_{ijmm} = b_{ijm} \\ 0, \text{otherwise} \end{cases}, \end{aligned} \tag{12}$$

$$q(\boldsymbol{v}_i|\boldsymbol{\eta}_i) = \text{Categorical}(\boldsymbol{\eta}_i), \tag{13}$$

with

$$q(v_{ij} = 1) = \eta_{ij}, j = 1, \cdots, K, \tag{14}$$

where $q(\mu_{ijm}|a_{ijm}, b_{ijm})$ is Gaussian distribution with mean a_{ijm} and variance b_{ijm}, $q(\Sigma_{ijmm}|b_{ijm})$ is Dirac delta distribution with parameter b_{ijm}[1], and $q(\boldsymbol{v}_i|\boldsymbol{\eta}_i)$ is categorical distribution with parameter $\boldsymbol{\eta}_i$. For the Dirac delta distribution $\delta(b_{ijm})$, the value of its PDF is equal to one if $\Sigma_{ijmm} = b_{ijm}$ and zero otherwise. A categorical distributed vector $\boldsymbol{v}_i$ is a one-hot vector with the probability η_{ij} that $v_{ij} = 1$. We set $\boldsymbol{\eta}$ as hyperparameters of the MoGMM, and the elements η_{ij} in $\boldsymbol{\eta}$ are not further assumed to be random variables.

Furthermore, we choose the prior distributions of the parameters in the MoGMM as

$$p(\mu_{ijm}) = \mathcal{N}(0, 1), \tag{15}$$

$$p(\Sigma_{ijmm}) = \text{Uniform}(0, \infty), \tag{16}$$

$$p(\boldsymbol{v}_i) = \text{Categorical}(\underbrace{\left[\frac{1}{K}, \cdots, \frac{1}{K}\right]}_{K \text{ elements}}), \tag{17}$$

with

$$p(v_{ij} = 1) = \frac{1}{K}, j = 1, \cdots, K, \tag{18}$$

where $p(\mu_{ijm})$ is a standard normal distribution with zero mean and unit variance, $p(\Sigma_{ijmm})$ is a uniform distribution in the interval of $(0, \infty)$[2], and $p(\boldsymbol{v}_i)$ is a categorical distribution with equal probabilities $\frac{1}{K}$. Here, note that $p(\Sigma_{ijmm}) = \text{Uniform}(0, \infty) = 1$ is an improper prior distribution, as the integral over its support is not equal to one. However, this does not affect the optimization of the MoGMM.

Then, the KL divergence $D_{\text{KL}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})||p(\boldsymbol{\Phi}))$ from the posterior $q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})$ to the prior $p(\boldsymbol{\Phi})$ can be presented by

$$\begin{aligned} & D_{\text{KL}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})||p(\boldsymbol{\Phi})) \\ =& \int q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}) \ln \frac{q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})}{p(\boldsymbol{\Phi})} d\boldsymbol{\Phi} \\ =& \int q(\boldsymbol{\mu}|\boldsymbol{A}, \boldsymbol{B}) q(\boldsymbol{\Sigma}|\boldsymbol{B}) q(\boldsymbol{V}|\boldsymbol{\eta}) \\ & \cdot \ln \frac{q(\boldsymbol{\mu}|\boldsymbol{A}, \boldsymbol{B}) q(\boldsymbol{\Sigma}|\boldsymbol{B}) q(\boldsymbol{V}|\boldsymbol{\eta})}{p(\boldsymbol{\mu}) p(\boldsymbol{\Sigma}) p(\boldsymbol{V})} d\boldsymbol{\mu}\, d\boldsymbol{\Sigma}\, d\boldsymbol{V} \\ =& \int q(\boldsymbol{\mu}|\boldsymbol{A}, \boldsymbol{B}) q(\boldsymbol{\Sigma}|\boldsymbol{B}) q(\boldsymbol{V}|\boldsymbol{\eta}) \\ & \cdot \left(\ln \frac{q(\boldsymbol{\mu}|\boldsymbol{A}, \boldsymbol{B})}{p(\boldsymbol{\mu})} + \ln \frac{q(\boldsymbol{\Sigma}|\boldsymbol{B})}{p(\boldsymbol{\Sigma})} + \ln \frac{q(\boldsymbol{V}|\boldsymbol{\eta})}{p(\boldsymbol{V})} \right) d\boldsymbol{\mu}\, d\boldsymbol{\Sigma}\, d\boldsymbol{V} \\ =& \underbrace{\int q(\boldsymbol{\mu}|\boldsymbol{A}, \boldsymbol{B}) \ln \frac{q(\boldsymbol{\mu}|\boldsymbol{A}, \boldsymbol{B})}{p(\boldsymbol{\mu})} d\boldsymbol{\mu}}_{D_{\text{KL}}(q(\boldsymbol{\mu}|\boldsymbol{A}, \boldsymbol{B})||p(\boldsymbol{\mu}))} + \underbrace{\int q(\boldsymbol{\Sigma}|\boldsymbol{B}) \ln \frac{q(\boldsymbol{\Sigma}|\boldsymbol{B})}{p(\boldsymbol{\Sigma})} d\boldsymbol{\Sigma}}_{D_{\text{KL}}(q(\boldsymbol{\Sigma}|\boldsymbol{B})||p(\boldsymbol{\Sigma}))} \\ & + \underbrace{\int q(\boldsymbol{V}|\boldsymbol{\eta}) \ln \frac{q(\boldsymbol{V}|\boldsymbol{\eta})}{p(\boldsymbol{V})} d\boldsymbol{V}}_{D_{\text{KL}}(q(\boldsymbol{V}|\boldsymbol{\eta})||p(\boldsymbol{V}))}, \end{aligned} \tag{19}$$

with

$$\begin{aligned} & p(\boldsymbol{\Phi}) \\ =& p(\boldsymbol{\mu}) \cdot p(\boldsymbol{\Sigma}) \cdot p(\boldsymbol{V}) \\ =& \prod_{i=1}^{C} \prod_{j=1}^{K} \prod_{m=1}^{M} p(\mu_{ijm}) \cdot \prod_{i=1}^{C} \prod_{j=1}^{K} \prod_{m=1}^{M} p(\Sigma_{ijmm}) \cdot \prod_{i=1}^{C} p(\boldsymbol{v}_i). \end{aligned} \tag{20}$$

The KL divergence in (19) can be decomposed into three KL divergence terms for $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$, and $\boldsymbol{V}$, respectively. These three terms can be further decomposed as

$$\begin{aligned} & D_{\text{KL}}(q(\boldsymbol{\mu}|\boldsymbol{A}, \boldsymbol{B})||p(\boldsymbol{\mu})) \\ =& \sum_{i=1}^{C} \sum_{j=1}^{K} \sum_{m=1}^{M} D_{\text{KL}}(q(\mu_{ijm}|a_{ijm}, b_{ijm})||p(\mu_{ijm})), \end{aligned} \tag{21}$$

$$\begin{aligned} & D_{\text{KL}}(q(\boldsymbol{\Sigma}|\boldsymbol{B})||p(\boldsymbol{\Sigma})) \\ =& \sum_{i=1}^{C} \sum_{j=1}^{K} \sum_{m=1}^{M} D_{\text{KL}}(q(\Sigma_{ijmm}|b_{ijm})||p(\Sigma_{ijmm})), \end{aligned} \tag{22}$$

$$\begin{aligned} & D_{\text{KL}}(q(\boldsymbol{V}|\boldsymbol{\eta})||p(\boldsymbol{V})) \\ =& \sum_{i=1}^{C} D_{\text{KL}}(q(\boldsymbol{v}_i|\boldsymbol{\eta}_i)||p(\boldsymbol{v}_i)), \end{aligned} \tag{23}$$

respectively, where

$$\begin{aligned} & D_{\text{KL}}(q(\mu_{ijm}|a_{ijm}, b_{ijm})||p(\mu_{ijm})) \\ =& \frac{1}{2} \left(b_{ijm} + a_{ijm}^2 - \ln b_{ijm} - 1 \right), \end{aligned} \tag{24}$$

$$\begin{aligned} & D_{\text{KL}}(q(\Sigma_{ijmm}|b_{ijm})||p(\Sigma_{ijmm})) \\ =& 0, \end{aligned} \tag{25}$$

$$\begin{aligned} & D_{\text{KL}}(q(\boldsymbol{v}_i|\boldsymbol{\eta}_i)||p(\boldsymbol{v}_i)) \\ =& \sum_{j=1}^{K} \eta_{ij} \ln(\eta_{ij} \cdot K). \end{aligned} \tag{26}$$

As $D_{\text{KL}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})||p(\boldsymbol{\Phi}))$ is treated as a regularization term in the SGVB and the nonnegative multiplier γ is assigned for the term, each sub-term in $D_{\text{KL}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})||p(\boldsymbol{\Phi}))$ for

[1] As distinct distribution forms can be introduced for Σ_{ijmm}, we select a simple and easily implemented form in the paper. Here, we assume the mean variance of each component is equal to the corresponding data variance to reflect the variations of the data in modeling. In addition to the Dirac delta distribution, other distributional forms can be empirically chosen as well. The different choices of the prior distributions do not majorly affect the performance of the whole model, since the influence of prior distributions can be ignored when we have enough training data.

[2] Although it is better to choose the prior distribution in the same form as the approximated posterior distribution, for the purpose of applying non-informative prior distribution, we selected the simple yet easily implemented form, *i.e.*, the uniform distribution. In addition to the uniform distribution, other distributions can be chosen as well.

TABLE I: Ablation studies with VGG16 on the CIFAR-10 dataset for misclassification detection. The number of components in each GMM (*i.e.*, K) is discussed. The effectiveness of two key parts in the DS-SGVB algorithm, *i.e.*, $L_D^{\text{NSGVB}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$ and $\text{Reg}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$, are discussed as well. "✓" means the part is contained and "○" means replacing $\text{Reg}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$ by the original $D_{\text{KL}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})||p(\boldsymbol{\Phi}))$. The best results are highlighted in **bold**.

Optimizer	$L_D^{\text{NSGVB}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$	$\text{Reg}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$	K	Accuracy (%)	AUROC (%)		AUPR (%)	
					Max.P.	Ent.	Max.P.	Ent.
Adam	✓	✓	1	92.09 ± 0.15	91.27 ± 0.35	91.22 ± 0.36	46.14 ± 1.23	46.51 ± 1.64
Adam	✓	✓	2	92.12 ± 0.14	91.12 ± 0.38	91.11 ± 0.38	46.01 ± 0.83	46.76 ± 1.03
Adam	✓	✓	4	92.16 ± 0.09	91.97 ± 0.38	91.93 ± 0.38	49.76 ± 0.88	49.24 ± 1.05
Adam	✓	✓	8	$\mathbf{92.64 \pm 0.31}$	$\mathbf{93.51 \pm 0.27}$	$\mathbf{93.48 \pm 0.27}$	$\mathbf{53.60 \pm 0.85}$	53.25 ± 0.48
Adam	✓	✓	16	92.25 ± 0.26	93.12 ± 0.30	93.09 ± 0.31	53.28 ± 0.51	53.09 ± 0.16
Adam	✓	✓	32	92.63 ± 0.38	93.14 ± 0.21	93.12 ± 0.21	53.30 ± 0.50	$\mathbf{53.32 \pm 0.42}$
Adam	✓	○	8	92.28 ± 0.76	92.67 ± 0.21	92.74 ± 0.30	50.05 ± 0.53	50.30 ± 0.27
Adam	✓		8	92.36 ± 0.25	90.89 ± 0.47	90.88 ± 0.49	46.69 ± 2.30	47.24 ± 2.64
Adam		✓	8	92.57 ± 0.10	91.37 ± 0.26	91.32 ± 0.28	45.95 ± 0.56	46.81 ± 1.08
Adam			8	92.36 ± 0.08	91.09 ± 0.34	91.03 ± 0.36	45.52 ± 0.88	46.37 ± 0.93
SGD	✓	✓	8	92.34 ± 0.56	92.76 ± 0.23	92.84 ± 0.24	52.08 ± 0.43	52.30 ± 0.32

TABLE II: Discussion of the influence of the bias vector in the classifier. We compare the performance of the baseline and that without the bias vector ("Baseline w/o bias") with the VGG16 on the CIFAR-10 dataset for misclassification detection. "×" means no significance.

Method	Accuracy (%)	AUROC (%)		AUPR (%)	
		Max.P.	Ent.	Max.P.	Ent.
Baseline	91.76 ± 0.09 (×)	91.36 ± 0.38 (×)	91.30 ± 0.37 (×)	46.51 ± 2.60 (×)	46.90 ± 2.57 (×)
Baseline w/o bias	91.67 ± 0.25	91.63 ± 0.18	91.57 ± 0.18	46.91 ± 1.59	47.21 ± 1.89

each parameter can be also treated as a sub-regularization term, and their corresponding sub-multipliers can be empirically set, respectively. Here, we set the nonnegative sub-multipliers $\omega_i^*\eta_{ij}^*$ for both $D_{\text{KL}}(q(\mu_{ijm}|a_{ijm}, b_{ijm})||p(\mu_{ijm}))$ and $D_{\text{KL}}(q(\Sigma_{ijmm}|b_{ijm})||p(\Sigma_{ijmm}))$, and ω_i^* for $D_{\text{KL}}(q(\boldsymbol{v}_i|\boldsymbol{\eta}_i)||p(\boldsymbol{v}_i))$. In this case, we can obtain the generalized form $\text{Reg}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$ of $D_{\text{KL}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})||p(\boldsymbol{\Phi}))$ with the sub-multipliers as

$$\begin{aligned}
&\text{Reg}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})) \\
=&\sum_{i=1}^{C}\sum_{j=1}^{K}\sum_{m=1}^{M}\omega_i^*\eta_{ij}^*\left\{\frac{1}{2}\left(b_{ijm}+a_{ijm}^2-\ln b_{ijm}-1\right)\right\} \\
&+\sum_{i=1}^{C}\sum_{j=1}^{K}\sum_{m=1}^{M}\omega_i^*\eta_{ij}^*\cdot 0 \\
&+\sum_{i=1}^{C}\omega_i^*\left\{\sum_{j=1}^{K}\eta_{ij}\ln(\eta_{ij}\cdot K)\right\} \\
=&\sum_{i=1}^{C}\omega_i^*\left\{\sum_{j=1}^{K}\left[\eta_{ij}\ln(\eta_{ij}\cdot K)\right.\right. \\
&\left.\left.+\frac{\eta_{ij}^*}{2}\sum_{m=1}^{M}\left(b_{ijm}+a_{ijm}^2-\ln b_{ijm}-1\right)\right]\right\}, \qquad (27)
\end{aligned}$$

where ω_i^* and η_{ij}^* are sub-multipliers and set equal to ω_i and η_{ij}, respectively, in this paper.

After optimization, as we obtain the optimal hyperparameters $\boldsymbol{A}$ and $\boldsymbol{B}$, we set $\mu_{ijm} = a_{ijm}$ and $\Sigma_{ijmm} = b_{ijm}$ directly in the inference procedure, which is a common setting in variational inference.

IV. Experimental Results and Discussions

We conducted three different UI tasks, including misclassification detection, open-set out-of-domain detection, and open-set out-of-distribution detection in image recognition. The proposed DS-UI was evaluated with VGG16 [41] and ResNet18 [42] as backbone models on CIFAR-10/-100 [43], street view house numbers (SVHN) [44], and tiny ImageNet (TIM) [45] datasets. We compared the DS-UI with the baseline [11], the MC dropout [10], the DPN [9], the RKL [13], and the SDE-Net [15]. In addition to the UI methods, we also compared the DS-UI with some classic open-set recognition methods, G-OpenMax [46], C2AE [47], and GDOSR [16], for the open-set out-of-domain detection.

A. Implementation Details

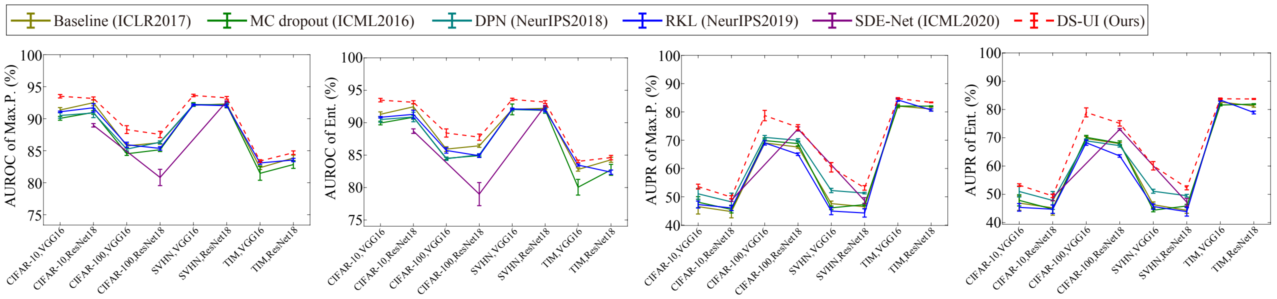
Following the settings in [11], [9], we introduced max probability (Max.P.) and entropy (Ent.) of output probabilities as metrics of uncertainty measurement and adopted the area under receiver operating characteristic curve (AUROC) and the area under precision-recall curve (AUPR) for evaluations. Large values of AUROC and AUPR indicate good performance.

In the training procedure, we applied Adam [48] optimizer by following [9], [11] with 100 epochs for CIFAR-10/-100, 40 epochs for SVHN, and 120 epochs for TIM. We used 1-cycle learning rate scheme, where we set initial learning rates as 7.5×10^{-4} for each dataset and cycle length as 70 epochs for CIFAR-10/-100, 30 epochs for SVHN, and 80 epochs for TIM. Weight decay values were set as 5×10^{-4}. γ (in (9)) and ρ (in (6)) were set as 1×10^{-4} and 4, respectively. We performed the same training strategy to the referred methods. Following [11], [9], the FC layers of VGG16 and ResNet18 are replaced by a three-layer FC net with 2048 hidden units for each hidden layer. Leaky ReLU [49] was used as the activation function. Hyperparameters of the referred methods were set the same as those in the original papers.

For all the methods, we conducted five runs and report the means and the standard deviations of recognition accuracies, the AUROC and the AUPR. The SDE-Net can be implemented with the ResNet structure only and the DPN does not work

TABLE III: Means and standard deviations of image recognition accuracies (%) on the four datasets. Note that "-" means that the model do not work in the case of reimplementation, "✓" means statistically significant difference between the accuracies of the DS-UI and those of the referred methods, "×" means no significance, and "N/A" means inapplicable. The best results are highlighted in **bold**.

Dataset	CIFAR-10		CIFAR-100		SVHN		TIM	
Method	VGG16	ResNet18	VGG16	ResNet18	VGG16	ResNet18	VGG16	ResNet18
Baseline (ICLR2017)	91.76 ± 0.09 (✓)	92.56 ± 0.14 (✓)	70.46 ± 0.24 (✓)	71.04 ± 0.19 (✓)	95.25 ± 0.11 (✓)	95.23 ± 0.13 (✓)	46.13 ± 0.41 (✓)	50.37 ± 0.45 (✓)
MC dropout (ICML2016)	91.76 ± 0.09 (✓)	92.56 ± 0.14 (✓)	70.46 ± 0.24 (✓)	71.04 ± 0.19 (✓)	95.25 ± 0.11 (✓)	95.23 ± 0.13 (✓)	46.13 ± 0.41 (✓)	50.37 ± 0.45 (✓)
DPN (NeurIPS2018)	90.73 ± 0.35 (✓)	91.98 ± 0.32 (✓)	67.56 ± 0.28 (✓)	69.80 ± 0.11 (✓)	93.51 ± 0.42 (✓)	93.98 ± 0.67 (✓)	-	-
RKL (NeurIPS2019)	92.25 ± 0.30 (×)	92.37 ± 0.23 (✓)	70.58 ± 0.26 (✓)	71.62 ± 0.76 (×)	94.95 ± 0.07 (✓)	95.09 ± 0.09 (✓)	45.21 ± 0.16 (✓)	50.31 ± 0.29 (✓)
SDE-Net (ICML2020)	-	92.15 ± 0.76 (✓)	-	52.52 ± 1.80 (✓)	-	95.00 ± 0.23 (✓)	-	-
DS-UI (Ours)	$\mathbf{92.64 \pm 0.31}$ (N/A)	$\mathbf{93.09 \pm 0.04}$ (N/A)	$\mathbf{71.39 \pm 0.38}$ (N/A)	$\mathbf{71.94 \pm 0.43}$ (N/A)	$\mathbf{95.71 \pm 0.11}$ (N/A)	$\mathbf{95.94 \pm 0.11}$ (N/A)	$\mathbf{46.83 \pm 0.28}$ (N/A)	$\mathbf{52.02 \pm 0.30}$ (N/A)



(a) AUROC (%) of Max.P. (b) AUROC (%) of Ent. (c) AUPR (%) of Max.P. (d) AUPR (%) of Ent.

Fig. 4: Performance of misclassification detection with the two backbones on the four datasets. Note that annotations in x-axis mean "dataset, backbone". The error bars represent standard deviations of the values of the metrics for the methods. The dashed lines in each subfigure present the DS-UI and the other solid lines present the referred methods.

on the TIM dataset in practice. In order to check if the DS-UI has statistically significant performance improvement compared with the referred methods, we conducted unpaired Student's t-tests between the values of the metrics of them. The significance level was set as 0.05.

B. Ablation Studies

We conducted ablation studies with VGG16 on the CIFAR-10 dataset under misclassification detection (Table I) to discuss the selection of number of components K in each GMM of the MoGMM, as well as the effectiveness of two key parts in the DS-SGVB, *i.e.*, $L_D^{\text{NSGVB}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$ and $\text{Reg}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$. Although the accuracies maintain steady in different cases, the AUROC and the AUPR change sharply in the full DS-SGVB after increasing K to eight and then obtain slight decreases when K rises to 16 and 32. Thus, we set K as eight in the following experiments. In addition, the results using $\text{Reg}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$ surpasses those using $D_{\text{KL}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi})||p(\boldsymbol{\Phi}))$. Meanwhile, only introducing $L_D^{\text{NSGVB}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$ or $\text{Reg}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$ cannot statistically significantly improve the performance. The AUROC and the AUPR of the full DS-SGVB can outperform those of removing one or two key parts, which means the two key parts are essential and should be combined in implementation.

We also discuss the influence of optimizer. As we used Adam for all the experiments, we compare its performance with those using SGD with momentum 0.9 (other parts in the training recipe are the same to those of the Adam) introducing $L_D^{\text{NSGVB}}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$ and $\text{Reg}(q_{\boldsymbol{\theta}}(\boldsymbol{\Phi}))$ with $K = 8$, which are also listed in Table I. The proposed DS-UI can also converge with SGD used, with slightly worse performance.

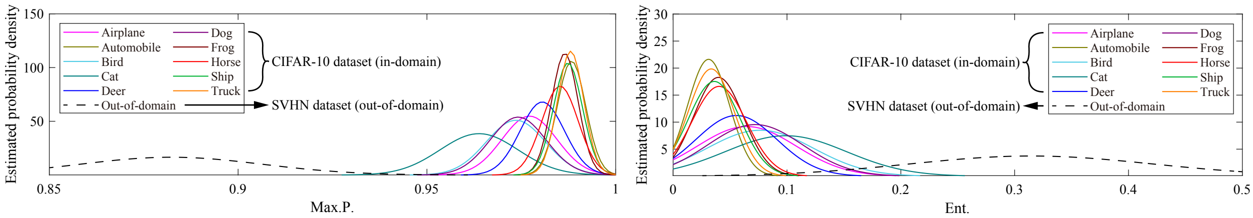
Here, we further discuss the influence of the bias vector in the classifier and conduct experiments, with the baseline as the uncertainty inference method and VGG16 as the base model, on the CIFAR-10 dataset under misclassification detection. Table II shows the comparison between the original baseline and that without the bias vector. It can be observed that removing the bias vector has no statistically significant effect on the performance. It is worthy to note that the proposed model can be only implemented with the bias vector removed, according to the model description in Section III-A.

C. Misclassification Detection

The first important task in the UI is misclassification detection, which aims at detecting mispredicted samples in the test sets with uncertainty. Table III lists the image recognition accuracies with two backbones on four datasets. According to Table III, the DS-UI leads to the best performance in each case and achieves statistically significant performance improvement in most of the cases except the RKL with VGG16 on the CIFAR-10 dataset and ResNet18 on the CIFAR-100 dataset. Figure 4 illustrates the experimental results in the misclassification detection. The DS-UI yields the best AUROC/AUPR in all the cases as well, and achieves statistically significant improvement in most of the cases. Therefore, we can conclude that the DS-UI is better for misclassification detection than the referred methods.

D. Open-set Out-of-domain Detection

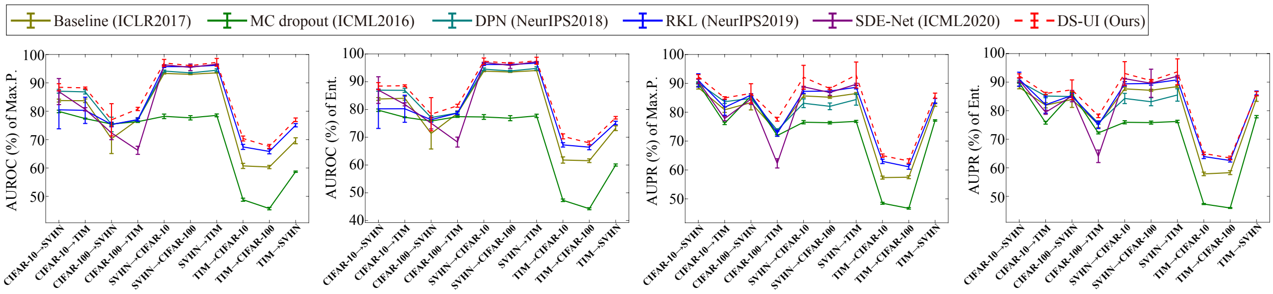
We further evaluated the DS-UI in open-set out-of-domain detection. Different in-domain and out-of-domain dataset pairs were applied for the task, and the out-of-domain set in each



(a) Distributions of Max.P.

(b) Distributions of Ent.

Fig. 5: Estimated Gaussian distributions of Max.P. and Ent. values in the open-set out-of-domain detection. Test sets of the "CIFAR-10 →SVHN" pair are chosen.



(a) AUROC (%) of Max.P.

(b) AUROC (%) of Ent.

(c) AUPR (%) of Max.P.

(d) AUPR (%) of Ent.

Fig. 6: Performance of open-set out-of-domain detection with ResNet18 on ten dataset pairs. Note that annotations in x-axis mean "in-domain dataset→out-of-domain dataset". The error bars represent standard deviations of the values of the metrics for the methods. The dashed lines in each subfigure present the DS-UI and the solid lines present the referred ones.

TABLE IV: Performance of open set out-of-domain detection on the CIFAR-10 and the TIM datasets (another setting). The datasets are divided into in-domain (ID) and out-of-domain (OoD) sets, respectively. "#ID" and "#OoD" mean the class numbers of the ID set and the OoD set, respectively. Note that "†" means the results in the row are obtained from [16], "✓" means statistically significant difference between the values of the evaluation metrics of the DS-UI and those of the referred methods, "×" means no significance, and "N/A" means inapplicable. The best results in each case are highlighted in **bold**.

Dataset	CIFAR-10 (#ID: 6, #OoD: 4)				TIM (#ID: 20, #OoD: 180)			
Metric	AUROC (%)		AUPR (%)		AUROC (%)		AUPR (%)	
Method	Max.P.	Ent.	Max.P.	Ent.	Max.P.	Ent.	Max.P.	Ent.
MC dropout (ICML2016)	66.88 ± 0.28 (✓)	66.61 ± 0.27 (✓)	54.89 ± 0.45 (✓)	54.22 ± 0.44 (✓)	65.09 ± 0.37 (✓)	64.60 ± 0.47 (✓)	93.28 ± 0.10 (✓)	92.98 ± 0.11 (✓)
RKL (NeurIPS2019)	76.47 ± 0.94 (✓)	76.58 ± 0.99 (✓)	65.49 ± 0.52 (×)	66.42 ± 0.49 (×)	70.30 ± 0.80 (✓)	70.89 ± 0.82 (✓)	93.85 ± 0.17 (✓)	93.97 ± 0.13 (✓)
SDE-Net (ICML2020)	77.02 ± 0.81 (✓)	77.94 ± 0.79 (✓)	64.57 ± 0.85 (✓)	66.53 ± 0.82 (×)	65.68 ± 0.99 (✓)	66.95 ± 1.06 (✓)	93.51 ± 0.31 (✓)	93.71 ± 0.31 (✓)
G-OpenMax (BMVC2017)†	67.50 ± 3.50 (✓)	-	-	-	58.00 ± N/A (✓)	-	-	-
C2AE (CVPR2019)†	71.10 ± 0.80 (✓)	-	-	-	58.10 ± 1.90 (✓)	-	-	-
GDOSR (CVPR2020)†	80.70 ± 3.90 (×)	-	-	-	60.80 ± 1.70 (✓)	-	-	-
DS-UI (Ours)	**81.02 ± 0.54** (N/A)	**81.34 ± 0.55** (N/A)	**66.26 ± 0.88** (N/A)	**67.26 ± 0.86** (N/A)	**72.27 ± 0.11** (N/A)	**73.10 ± 0.24** (N/A)	**94.90 ± 0.08** (N/A)	**95.11 ± 0.09** (N/A)

pair was not used for training. Figure 5 shows the distributions of Max.P. and Ent. on the test sets of the "CIFAR-10→SVHN" pair as an example. We can observe that distribution of out-of-domain samples is almost separated from those of in-domain classes, which means the DS-UI can effectively estimate uncertainty. Figure 6 shows that the DS-UI can surpass all the referred methods in most of the cases, except the AUPR of Ent. on the "TIM→SVHN" pair. Although the RKL outperforms the DS-UI in the case, there is no statistically significant difference between them, as the p-value of the unpaired Student's t-test is larger than 0.05. In addition, the DS-UI obtains statistically significant improvement in most of the other cases, which shows the superiority of the DS-UI in the task.

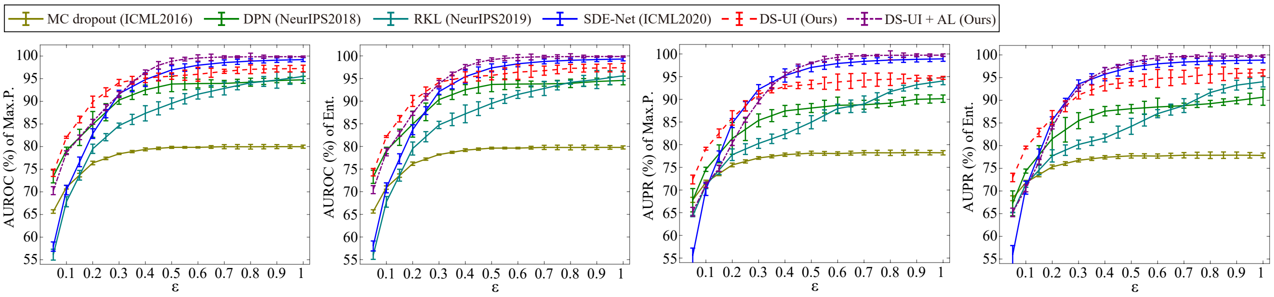
In addition, we also evaluated the DS-UI following the settings in [16]. The CIFAR-10 and the TIM datasets were divided into in-domain and out-of-domain sets, respectively. In Table IV, the best performance of the DS-UI under the metrics can be found on two datasets and the DS-UI achieves statistically significant improvement in all the cases on the TIM dataset and most of the cases on the CIFAR-10 dataset. The results show the remarkable ability of the DS-UI in the out-of-domain detection task.

E. Open-set Out-of-distribution Detection

We then evaluated the DS-UI in open-set out-of-distribution detection on a synthetic noise dataset. The dataset contains $10,000$ random images, where each pixel is independently sampled from a uniform distribution in $[0, 1]$. Table V shows the experimental results in open-set out-of-distribution detection between the CIFAR-100 dataset and the synthetic noise dataset. Although the DS-UI can only obtain the second best results under the metrics, no statistically significant difference is observed between the values of the evaluation metrics of

TABLE V: Performance of open-set out-of-distribution detection between CIFAR-100 dataset and the synthetic uniform noise dataset. Means and standard deviations of the metrics (%) are shown. Note that "✓" means statistically significant difference between the values of the metrics of the DS-UI and those of the referred methods, "×" means no significance, and "N/A" means inapplicable. The best and the second best results are highlighted in **bold** and <u>underline</u>. [3]The SDE-Net applied adversarial learning (AL) with noisy input samples during its training procedure. The training procedure of the AL is undertaken similarly to the test procedure of the open-set out-of-distribution detection task, as both of them add noises into the input samples. Thus, the AL can benefit the open-set out-of-distribution detection.

Method	AUROC (%)		AUPR (%)	
	Max.P.	Ent.	Max.P.	Ent.
Baseline (ICLR2017)	84.20 ± 4.56 (✓)	86.70 ± 5.75 (✓)	74.14 ± 6.97 (✓)	77.18 ± 8.40 (✓)
MC dropout (ICML2016)	82.39 ± 0.16 (✓)	84.11 ± 0.18 (✓)	78.24 ± 0.31 (✓)	79.07 ± 0.31 (✓)
DPN (NeurIPS2018)	88.99 ± 3.01 (✓)	90.35 ± 0.64 (✓)	82.67 ± 1.47 (✓)	83.55 ± 2.87 (✓)
RKL (NeurIPS2019)	87.64 ± 7.15 (✓)	88.61 ± 6.03 (✓)	79.44 ± 10.57 (✓)	80.17 ± 8.89 (✓)
SDE-Net (ICML2020)[3]	$\mathbf{97.44 \pm 2.39}$ (×)	$\mathbf{98.13 \pm 2.22}$ (×)	$\mathbf{93.68 \pm 6.82}$ (×)	$\mathbf{94.46 \pm 6.64}$ (×)
DS-UI (Ours)	$\underline{97.43 \pm 0.81}$ (N/A)	$\underline{97.50 \pm 0.76}$ (N/A)	$\underline{91.76 \pm 3.33}$ (N/A)	$\underline{91.72 \pm 3.40}$ (N/A)



(a) AUROC (%) of Max.P. (b) AUROC (%) of Ent. (c) AUPR (%) of Max.P. (d) AUPR (%) of Ent.

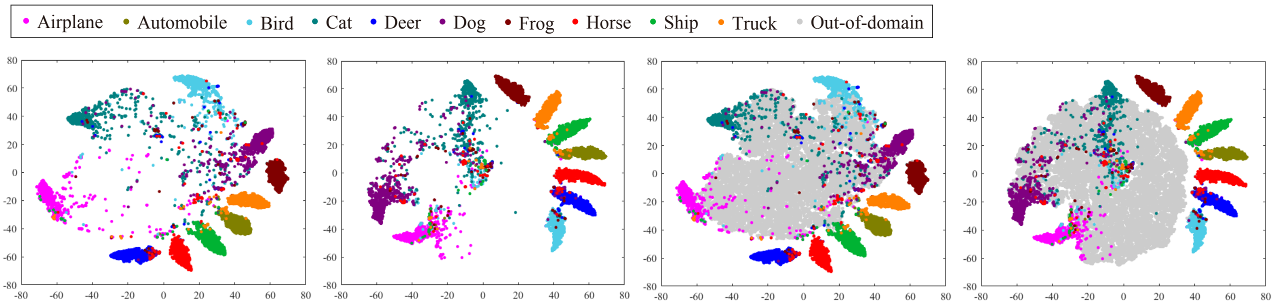
Fig. 7: Performance of open-set out-of-distribution detection under FGSM attacks with ResNet18 on the CIFAR-10 dataset. ε is the step size in the FGSM and selected in the set $\{\frac{n}{20}\}_{n=1}^{20}$. The error bars represent standard deviations of the values of the metrics. The dashed and solid lines in each subfigure present the DS-UI and the referred methods, respectively.

the DS-UI and those of the SDE-Net (the p-values of the unpaired Student's t-test are all larger than 0.05). The SDE-Net performs the best as it involves adversarial learning (AL) during its training procedure. This means it benefits from both the UI and the AL. In summary, the DS-UI works well and achieves comparable performance with the AL-based method (SDE-Net).

Furthermore, we evaluated the DS-UI under the adversarial attack, which can be considered as a distributional attack task, on the CIFAR-10 dataset. We introduced fast gradient-sign method (FGSM) [50] as the attacker in the original input images on the test set. Treating the attacked images as the out-of-distribution samples, the adversarial attack task can be seen as an open-set out-of-distribution detection task. Parameter ε in the FGSM presents the amplitude of the noises (or called the offset of distribution shift), which was selected in the set $\{\frac{n}{20}\}_{n=1}^{20}$ [15]. For different ε, independent experiments were conducted. Figure 7 shows the DS-UI obtains statistically significant improvement when ε is small ($\varepsilon \leq 0.2$, *i.e,*, adding minor noises) and even outperforms the AL-based SDE-Net. Although the SDE-Net can gain almost 99% on all the four metrics when ε is large ($\varepsilon \geq 0.4$, which are easier cases than the cases that minor noises are added), the DS-UI can perform comparably. Thus, the DS-UI can obtain superior ability in this task. In addition, we further combined the AL into the proposed DS-UI and the corresponding results are shown in Figure 7 as well. The AL can indeed improve the performance of the DS-UI in the open-set out-of-distribution detection task and the DS-UI with the AL can outperform the SDE-Net on all the evaluation metrics with each ε.

F. Visualizations

We conducted visualizations of the feature spaces of feature $\boldsymbol{z}$ of the baseline [11] and the proposed DS-UI by t-distributed stochastic neighbor embedding (t-SNE) [51], respectively, and show the results in Figure 8. The ResNet18 model was used as the backbone, and the test sets of the CIFAR-10 and the SVHN datasets were used as the in-domain and the out-of-domain datasets, respectively. For the baseline in Figure 8(a), all the classes are fused with each other and the inter-class margins are small, especially airplane, bird, cat, and dog (most confusing classes) mix with the other ones. In contrast, the DS-UI in Figure 8(b) obtains larger margins between most of the classes and the four most confusing classes are far away from the other ones, which is much better for the misclassification detection. Meanwhile, the intra-class distances of the DS-UI is also smaller than the baseline. More importantly, a clear and patent margin can be found between most of the in-domain classes and the out-of-domain samples in Figure 8(d), even though some in-domain classes are partly confused with the out-of-domain samples. In the baseline model, the in-domain samples and the out-of-domain samples are more confusing with each other (Figure 8(c)). It can be observed that the DS-UI can not only reduce intra-class distances, but also obtain



(a) Baseline (b) DS-UI (c) Baseline with OoD (d) DS-UI with OoD

Fig. 8: Visualizations of feature spaces of samples in the test sets of the baseline and the DS-UI with ResNet18 on the CIFAR-10 dataset as an example. The SVHN dataset is selected as the out-of-domain (OoD) dataset.

much wider inter-class margins than the baseline model for both the misclassification detection and the open-set out-of-domain detection.

V. Conclusions

In order to improve UI, DS-UI, a dual-supervised learning framework has been introduced to UI. Conventional UI methods commonly define uncertainty only on the outputs of DNNs. In the DS-UI, an MoGMM-FC layer that combines the classifier with an MoGMM was proposed to act as a probabilistic interpreter for the features of the DNNs. To enhance the learning ability of the MoGMM-FC layer, the DS-SGVB algorithm was proposed. It comprehensively considers both positive and negative samples to not only reduce the intra-class distances, but also enlarge the inter-class margins simultaneously. We evaluated the proposed DS-UI in the three UI tasks. Experimental results show the proposed DS-UI outperforms the state-of-the-art UI methods in misclassification detection. In addition, we found the DS-UI can achieve statistically significant improvements in open-set out-of-domain/-distribution detection. Visualizations also support the superiority of the DS-UI for the learning ability enhancement.

Acknowledgment

This work was supported in part by the National Key R&D Program of China under Grant 2019YFF0303300 and under Subject II No. 2019YFF0303302, in part by National Natural Science Foundation of China (NSFC) No. 61922015, 61773071, U19B2036, in part by Beijing Natural Science Foundation Project No. Z200002.

References

[1] Y. Wang, Y. Cao, Z.-J. Zha, J. Zhang, and Z. Xiong, "Deep degradation prior for low-quality image classification," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 049–11 058.

[2] Z. Wang, S. Wang, S. Yang, H. Li, J. Li, and Z. Li, "Weakly supervised fine-grained image classification via Guassian mixture model oriented discriminative learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9749–9758.

[3] D. Chang, Y. Ding, J. Xie, A. K. Bhunia, X. Li, Z. Ma, M. Wu, J. Guo, and Y.-Z. Song, "The devil is in the channels: Mutual-channel loss for fine-grained image classification," *IEEE Transactions on Image Processing (TIP)*, vol. 29, pp. 4683–4695, 2020.

[4] Y. Ding, S. Wen, J. Xie, D. Chang, Z. Ma, Z. Si, and H. Ling, "Weakly supervised attention pyramid convolutional neural network for fine-grained visual classification," *IEEE Transactions on Image Processing (TIP)*, 2021.

[5] D. Chang, K. Pang, Y. Zheng, Z. Ma, Y.-Z. Song, and J. Guo, "Your "flamingo" is my "bird": Fine-grained, or not," in *Computer Vision and Pattern Recognition (CVPR)*, 2021.

[6] W. Sun and Z. Chen, "Learned image downscaling for upscaling using content adaptive resampler," *IEEE Transactions on Image Processing (TIP)*, vol. 29, pp. 4027–4040, 2020.

[7] W. Sun, Z. Chen, and F. Wu, "Visual scanpath prediction using IOR-ROI recurrent mixture density network," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 43, no. 6, pp. 2101–2118, 2021.

[8] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles." in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 6402–6413.

[9] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018, pp. 7047–7058.

[10] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *International Conference on Machine Learning (ICML)*, 2016, pp. 1050–1059.

[11] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[12] W. J. Maddox, T. Garipov, P. Izmailov, D. Vetrov, and A. G. Wilson, "A simple baseline for Bayesian uncertainty in deep learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[13] A. Malinin and M. Gales, "Reverse KL-divergence training of prior networks: Improved uncertainty and adversarial robustness," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 14 520–14 531.

[14] J. Chang, Z. Lan, C. Cheng, and Y. Wei, "Data uncertainty learning in face recognition," in *Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 5709–5718.

[15] L. Kong, J. Sun, and C. Zhang, "SDE-Net: Equipping deep neural networks with uncertainty estimates," in *International Conference on Machine Learning (ICML)*, 2020.

[16] P. Perera, V. I. Morariu, R. Jain, V. Manjunatha, C. Wigington, V. Ordonez, and V. M. Patel, "Generative-discriminative feature representations for open-set recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[17] G. Malsiner-Walli, S. Frühwirth-Schnatter, and B. Grün, "Identifying mixtures of mixtures using Bayesian estimation," *Journal of Computational and Graphical Statistics*, vol. 26, no. 2, pp. 285–295, 2017.

[18] M. D. Zio, U. Guarnera, and R. Rocci, "A mixture of mixture models for a classification problem: The unity measure error," *Computational Statistics & Data Analysis*, vol. 51, pp. 2573–2585, 2007.

[19] Z. Ma and A. Leijon, "Bayesian estimation of Beta mixture models with variational inference," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 33, no. 11, pp. 2160–2173, 2011.

[20] Z. Ma, J. Xie, Y. Lai, J. Taghia, J.-H. Xue, and J. Guo, "Insights into multiple/single lower bound approximation for extended variational inference in non-Gaussian structured data modeling," *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, vol. 31, no. 7, pp. 2240–2254, 2020.

[21] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, “Weight uncertainty in neural networks,” in *International Conference on Machine Learning (ICML)*, 2015, pp. 1613–1622.

[22] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *ArXiv preprint, arXiv:1207.0580*, 2012.

[23] C. Li, C. Chen, D. Carlson, and L. Carin, “Preconditioned stochastic gradient langevin dynamics for deep neural networks,” in *AAAI Conference on Artificial Intelligence*, 2016, pp. 1788–1794.

[24] K. Lee, K. Lee, H. Lee, and J. Shin, “A simple unified framework for detecting out-of-distribution samples and adversarial attacks,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[25] K. Posch and J. Pilz, “Correlated parameters to accurately measure uncertainty in deep neural networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 3, pp. 1037–1051, 2020.

[26] J. Antorán, U. Bhatt, T. Adel, A. Weller, and J. Hernández-Lobato, “Getting a CLUE: A method for explaining uncertainty estimates,” in *International Conference on Learning Representations*, 2021.

[27] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in *International Conference on Machine Learning (ICML)*, 2017.

[28] M. Mozejko, M. Susik, and R. Karczewski, “Inhibited softmax for uncertainty estimation in neural networks,” *ArXiv preprint, arXiv:1810.01861*, 2018.

[29] J. Van Amersfoort, L. Smith, Y. W. Teh, and Y. Gal, “Uncertainty estimation using a single deep deterministic neural network,” in *International Conference on Machine Learning*, 2020, pp. 9690–9700.

[30] M. Havasi, R. Jenatton, S. Fort, J. Z. Liu, J. Snoek, B. Lakshminarayanan, A. M. Dai, and D. Tran, “Training independent subnetworks for robust prediction,” in *International Conference on Learning Representations*, 2021.

[31] A. Piergiovanni and M. Ryoo, “Temporal Gaussian mixture layer for videos,” in *International Conference on Machine Learning (ICML)*, 2019, pp. 5152–5161.

[32] E. Variani, E. McDermott, and G. Heigold, “A Gaussian mixture model layer jointly optimized with discriminative features within a deep neural network architecture,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4270–4274.

[33] J. Wang and J. Jiang, “An unsupervised deep learning framework via integrated optimization of representation learning and GMM-based modeling,” in *Asian Conference on Computer Vision (ACCV)*, 2018, pp. 249–265.

[34] J. Altosaar, R. Ranganath, and D. M. Blei, “Proximity variational inference,” in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

[35] K. Fan, Z. Wang, J. M. Beck, J. T. Kwok, and K. A. Heller, “Fast second-order stochastic backpropagation for variational inference,” in *Advances in Neural Information Processing Systems (NIPS)*, 2015, pp. 1387–1395.

[36] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, “Stochastic variational inference,” *Journal of Machine Learning Research (JMLR)*, vol. 14, pp. 1303–1347, 2013.

[37] T. Plötz, A. S. Wannenwetsch, and S. Roth, “Stochastic variational inference with gradient linearization,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1566–1575.

[38] R. Ranganath, S. Gerrish, and D. M. Blei, “Black box variational inference,” in *International Conference on Articial Intelligence and Statistics (AISTATS)*, 2014, pp. 814–822.

[39] C. M. Bishop, *Pattern recognition and machine learning*. Springer Science+Business Media LLC., 2006.

[40] M. Teye, H. Azizpour, and K. Smith, “Bayesian uncertainty estimation for batch normalized deep networks,” in *International Conference on Machine Learning (ICML)*, 2018, pp. 4907–4916.

[41] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[42] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[43] A. Krizhevsky, “Learning multiple layers of features from tiny images,” CIFAR, techreport, 2009.

[44] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, “Reading digits in natural images with unsupervised feature learning,” in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[45] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet large scale visual recognition challenge,” *International Journal on Computer Vision (IJCV)*, 2015.

[46] Z. Ge, S. Demyanov, and R. Garnavi, “Generative OpenMax for multi-class open set classification,” in *British Machine Vision Conference (BMVC)*, 2017.

[47] P. Oza and V. M. Patel, “C2AE: Class conditioned auto-encoder for open-set recognition,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

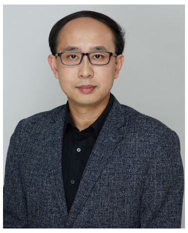
[48] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *International Conference on Learning Representations (ICLR)*, 2015.

[49] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in *International Conference on Machine Learning (ICML)*, vol. 30, no. 1, 2013, p. 3.

[50] I. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in *International Conference on Learning Representations (ICLR)*, 2015.

[51] L. v. d. Maaten and G. Hinton, “Visualizing data using t-SNE,” *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

Jiyang Xie received his B.E. degree in information engineering from Beijing University of Posts and Telecommunications (BUPT), China, in 2017, where he is currently pursuing the Ph.D. degree. His research interests include pattern recognition and machine learning fundamentals with a focus on applications in image processing, data mining, and deep learning.



Zhanyu Ma is currently a Professor at Beijing University of Posts and Telecommunications. He received the Ph.D. degree in electrical engineering from the KTH Royal Institute of Technology, Sweden, in 2011. From 2012 to 2013, he was a Postdoctoral Research Fellow with the School of Electrical Engineering, KTH Royal Institute of Technology. He has been an Associate Professor with the Beijing University of Posts and Telecommunications, Beijing, China, from 2014 to 2019. His research interests include pattern recognition and machine learning fundamentals with a focus on applications in computer vision, multimedia signal processing, data mining. He is a Senior Member of IEEE.

Jing-Hao Xue received the Dr.Eng. degree in signal and information processing from Tsinghua University in 1998 and the Ph.D. degree in statistics from the University of Glasgow in 2008. He is an associate professor in the Department of Statistical Science, University College London. His research interests include statistical machine learning, high-dimensional data analysis, pattern recognition and image analysis.

Guoqiang Zhang Guoqiang Zhang received the B. Eng. from University of Science and Technology of China (USTC) in 2003, M.Phil. degree from University of Hong Kong in 2006, and Ph.D. degree from Royal Institute of Technology in 2010. From the spring of 2011, he worked as a Postdoctoral Researcher at Delft University of Technology. From the spring of 2015, he worked as a senior researcher at Ericsson AB, Sweden. Since 2017, he has been a senior lecturer in the School of Electrical and Data Engineering, University of Technology Sydney, Australia. He is an IEEE member. His current research interests include distributed optimization, large scale optimization, deep learning, and multimedia signal processing.

Jian Sun holds a PhD in signal and information processing from Beijing University of Posts and Telecommunications. His research interests include NLP and conversational AI. He serves as a member in Chinese Association for Artificial Intelligence and Chinese Information Processing Society of China. He also serves as reviewers on various top conferences.

Yinhe Zheng received his Ph.D. degree from China University of Geosciences (Beijing) in 2017. He worked as a Post Doctor in the Department of Computer Science and Technology, Tsinghua University. His research interests include natural language processing and dialogue system, especially the tasks related to natural language understanding and natural language generation.

Jun Guo received the B.E. and M.E. degrees from the Beijing University of Posts and Telecommunications (BUPT), China, in 1982 and 1985, respectively, and the Ph.D. degree from the Tohuku Gakuin University, Japan, in 1993. He is currently a professor and a vice president with BUPT. He has authored over 200 papers in journals and conferences, including Science, Nature Scientific Reports, the IEEE Transactions on PAMI, Pattern Recognition, AAAI, CVPR, ICCV, and SIGIR. His research interests include pattern recognition theory and application, information retrieval, content-based information security, and bioinformatics.