RankMixup: Ranking-Based Mixup Training for Network Calibration

Jongyoun Noh Hyekang Park Junghyup Lee Bumsub Ham*
School of Electrical and Electronic Engineering, Yonsei University
https://cvlab.yonsei.ac.kr/projects/RankMixup

Abstract

Network calibration aims to accurately estimate the level of confidences, which is particularly important for employing deep neural networks in real-world systems. Recent approaches leverage mixup to calibrate the network's predictions during training. However, they do not consider the problem that mixtures of labels in mixup may not accurately represent the actual distribution of augmented samples. In this paper, we present RankMixup, a novel mixup-based framework alleviating the problem of the mixture of labels for network calibration. To this end, we propose to use an ordinal ranking relationship between raw and mixup-augmented samples as an alternative supervisory signal to the label mixtures for network calibration. We hypothesize that the network should estimate a higher level of confidence for the raw samples than the augmented ones (Fig.1). To implement this idea, we introduce a mixup-based ranking loss (MRL) that encourages lower confidences for augmented samples compared to raw ones, maintaining the ranking relationship. We also propose to leverage the ranking relationship among multiple mixup-augmented samples to further improve the calibration capability. Augmented samples with larger mixing coefficients are expected to have higher confidences and vice versa (Fig.1). That is, the order of confidences should be aligned with that of mixing coefficients. To this end, we introduce a novel loss, M-NDCG, in order to reduce the number of misaligned pairs of the coefficients and confidences. Extensive experimental results on standard benchmarks for network calibration demonstrate the effectiveness of RankMixup.

1. Introduction

Deep neural networks (DNNs) have achieved remarkable advances in numerous computer vision tasks, including image classification, object detection, and semantic segmentation. Despite the impressive performance, DNNs trained using a softmax cross-entropy (CE) loss and one-hot encoded labels often output overconfident and/or underconfident probabilities, which do not reflect the actual likelihood [10, 32, 41]. This suggests that the networks are not able to estimate the level of confidence, leading to inaccurate and unreliable predictions, which restricts practical applications in real-world systems, particularly for safety-sensitive domains, such as medical diagnosis [18] and autonomous driving [9].

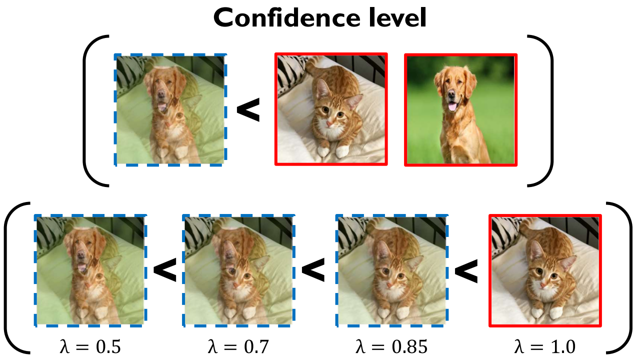

Figure 1: Motivation of RankMixup. We show raw samples of a cat and dog in red boxes, and mixup-augmented samples in blue boxes with dashed lines. The mixup-augmented samples are generated by linearly interpolating the raw samples with a mixing coefficient λ. We expect that confidences of the raw samples to be higher than that of the augmented sample (top). The augmented samples with larger coefficients λ would have higher confidences than the samples with smaller coefficients (bottom). Best viewed in color.

To improve the reliability of DNNs, many methods propose to calibrate the output distribution after or during training. Post-hoc approaches adjust miscalibrated probabilities directly from pretrained networks [7, 10, 12, 21, 27]. Although these approaches are straightforward to apply, calibration performance depends heavily on datasets and networks used for training [25], and they are not robust to distribution shifts between training and test samples [33]. Training-time approaches [4, 8, 14, 25, 28, 29, 30, 41, 51], on the other hand, exploit an additional regularization term along with the CE loss to calibrate the output probabilities

[*] Corresponding author.

during training. Early works [29, 30, 41] have explored the effects of regularization techniques, including label smoothing (LS) [40], focal loss (FL) [24], and mixup [49], on network calibration, and shown that the networks trained with these techniques are better calibrated than the ones trained with the CE loss alone. Recently, the works of [8, 25, 29] highlight the limitations of LS and FL in terms of network calibration, and introduce calibration-aware LS and FL. On the contrary, few studies have explored integrating the mixup technique into network calibration. Current mixup-based methods [35, 41, 51] simply interpolate input images and training labels with a mixing coefficient to soften output probabilities, while not considering network calibration itself. Since the interpolated labels may not represent accurately the distribution of the label mixture in augmented images [3, 19, 39, 43], training networks with such uncertain labels degrades the calibration performance.

In this work, we present a mixup-based approach to network calibration, dubbed RankMixup, that addresses the problem of directly using the mixture of labels for calibrating network predictions [35, 41, 51]. To this end, we conjecture that networks should give higher confidences for the raw sample, not augmented by the mixup technique, than the augmented one (Fig. 1). To implement this idea, we leverage an ordinal ranking relationship between raw and augmented samples (Fig. 1) and use it as a supervisory signal for network calibration, instead of using the label mixtures. In particular, we introduce a mixup-based ranking loss (MRL) that encourages the confidences of augmented samples to be lower than those of raw samples by a margin, in order to preserve the ranking relationship. To further strengthen the capability of network calibration, we also propose to exploit the ranking relationship for multiple mixup-augmented samples. We expect that higher confidences are favorable for augmented samples with larger mixing coefficients λ, and vice versa (Fig. 1), such that the orders of confidences and mixing coefficients are aligned to each other. To achieve this, we introduce a novel loss, M-NDCG, based on normalized discounted cumulative gain (NDCG) in information retrieval [17], which penalizes the number of misaligned pairs of the coefficients and confidences. Extensive experimental results on standard benchmarks demonstrate the effectiveness of our approach to leveraging the ordinal ranking relationships on network calibration. The main contributions of our work can be summarized as follows:

- We introduce a novel mixup-based framework for network calibration, RankMixup, that alleviates the problem of exploiting label mixtures for calibrating network predictions. To the best of our knowledge, we are the first to handle this issue in the context of confidence calibration.
- We propose to exploit ordinal ranking relationships between confidences from raw and augmented samples, as well as those among confidences from multiple augmented samples, and use them as supervisory signals, instead of the mixture of labels. To implement this idea, we introduce MRL and M-NDCG that facilitate aligning the confidences and mixing coefficients to preserve the relationships.
- We demonstrate that our approach provides better results in terms of expected calibration error (ECE), compared to other mixup-based methods. We also achieve competitive performances on standard benchmarks on network calibration, including CIFAR10/100 [20], Tiny-ImageNet [23], and ImageNet [6] with various network architectures.

2. Related work

Confidence calibration. The seminal work of [10] provides empirical evidence that DNNs are prone to making overconfident predictions due to miscalibrated output probabilities. In particular, they are typically trained with one-hot encoded labels, which fails to impose any entropy on the supervisory signal. The entropy of output probabilities for the network is thus encouraged to be zero, leading to the overconfident predictions [25]. To address this problem, several approaches have been proposed to augment the entropy of output probabilities, which can be divided into two categories: post-hoc and training-time methods. First, post-hoc approaches adjust the miscalibrated probabilities directly after training using a temperature scaling technique [7, 10, 42] or ranking statistics [27]. The temperature scaling technique, also effective to knowledge distillation [16], raises the entropy of output probabilities using a temperature parameter, optimized on additional validation samples. Although post-hoc approaches have shown the effectiveness on in-distribution datasets, they perform poorly under distribution shifts [33], and inevitably lower the confidences of correct predictions [22, 29]. Second, training-time approaches implicitly or explicitly impose regularization on outputs probabilities during training, usually in combination with post-hoc methods to further improve the calibration accuracy. The implicit methods [4, 4, 8, 25, 30, 35, 41] smooth the output distribution by increasing the entropy of training labels, rather than adjusting output probabilities directly. Several works [29, 30, 41, 51] have demonstrated empirically and theoretically that regularization methods such as LS [40], FL [24], and Mixup [49], which have originally designed to learn discriminative features, can also be effective in network calibration. However, recent works of [8, 25, 35] explore that directly using these methods for network calibration is problematic, and they introduce improved versions to overcome the drawback. For example, MbLS [25] exploits a margin-based LS to address the problem of LS providing non-informative predictions to obtain a better-calibrated network. The works of [8, 35] suggest that finding optimal parameters of FL and mixup is crucial for network calibration. The explicit methods [14, 22, 28, 34] adjust confidences of predictions directly by maximizing the entropy of output probabilities [34], using an ordinal ranking relationship [28] or minimizing calibration errors [14, 22]. For example, the ordinal ranking between output probabilities of networks has been proven to be effective to estimate the levels of confidence, providing better calibrated models [28]. It assumes that the predictions with higher confidences are more likely to be accurate than the ones with lower confidences. We explore the effects of mixup [49] in the context of network calibration, similar to other mixup-based methods [35, 41, 51]. They use interpolated labels to train networks for calibrating output probabilities, but the labels do not provide the distribution of the label mixture for the augmented sample accurately [3, 19, 39, 43]. We exploit confidences of augmented samples directly as supervisory signals for network calibration using ordinal ranking relationships between the samples, boosting the calibration performance significantly. The work of [28] is closely related to ours in that it uses an ordinal ranking relationship for raw samples. In contrast to this, our approach exploits more diverse and complex ordinal relationships for confidences. It considers not only the relationship between raw and augmented samples, but also among the augmented samples themselves.

Mixup. Data augmentation has become a standard technique for training DNNs that regularizes the networks to prevent overfitting and improve generalization. Mixup [49] linearly interpolates between pairs of raw data samples. Specifically, it takes a weighted average of two images to obtain a new one. The training label for the augmented sample is also interpolated with the same weight. Many works extend this idea to improve the generalization ability. Manifold-Mixup [45] performs the mixup process in the feature space, while CutMix [47] introduces a regional mixing scheme to preserve fine-grained information of original images. It has recently been explored that using label mixtures in mixup and its variants could provide misleading supervisory signals for training networks, which can be alleviated by saliency-guided mixup schemes [3, 19, 36, 43] or label perturbations [39]. In addition to data augmentation, various applications leverage mixup for, *e.g.*, self-supervised learning [37, 38] and network calibration [35, 41]. We mainly investigate the effect of the vanilla mixup [49] on network calibration. However, as will be shown in our experiments, the variants of mixup, including CutMix [47] and manifold-Mixup [45], can also be readily adapted to our approach.

3. Approach

We present in this section preliminaries for network calibration and mixup [49] in Sec. 3.1, and provide a detailed description of RankMiup and training losses in Secs. 3.2 and 3.3, respectively.

3.1. Preliminaries

Network calibration. Let us suppose a dataset $D = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ of size N, where $\mathbf{x}_i \in \mathcal{X}$ and $\mathbf{y}_i \in \mathcal{Y}$ are an input sample and a ground-truth label with K classes, respectively. DNNs can be considered as a function, parameterized by θ, that yields a logit $\mathbf{z}_i$ of size $K \times 1$ from the input sample: $\mathbf{f}_\theta : \mathcal{X} \rightarrow \mathbb{R}^K$. For multiclass classification, the networks compute class probabilities using the softmax function as follows:

$$p_{i,k} = \text{softmax}(\mathbf{z}_i) = \frac{\exp z_{i,k}}{\sum_j^K \exp z_{i,j}}, \tag{1}$$

where we denote by $p_{i,k}$ a probability of the k-th class for the i-th input. Based on the probabilities, the networks predict a class for the input sample $\mathbf{x}_i$ as

$$\hat{y}_i = \arg\max_k p_{i,k}. \tag{2}$$

The corresponding confidence can then be obtained as follows:

$$\hat{p}_i = \max_k p_{i,k}. \tag{3}$$

During training, the networks are optimized using a cross-entropy (CE) loss as follows:

$$\mathcal{L}_{CE}(\mathbf{x}_i, \mathbf{y}_i) = -\sum_{k=1}^K y_{i,k} \log p_{i,k}, \tag{4}$$

where $y_{i,k}$ is the k-th element of the one-hot encoded label $\mathbf{y}_i$. Recent works [10, 29, 30, 41] have shown that DNNs trained with the CE loss tend to be overconfident, providing higher confidences than accuracies. To address this problem, confidence calibration methods encourage that the confidence for each sample and the accuracy are the same as follows:

$$\hat{p}_i = P(\hat{y}_i = y_i | \hat{p}_i), \tag{5}$$

where y_i is an index of the true class for the ground-truth label $\mathbf{y}_i$.

Mixup. Mixup [49] is a data-augmentation technique based on the vicinal risk minimization principle [2]. It introduces the mixup vicinal distribution, which measures the likelihood of finding a pair of an augmented input and a target label, $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i)$, from the vicinity of the original pair, $(\mathbf{x}_i, \mathbf{y}_i)$. Mixup samples the vicinal distribution to construct a new dataset, $D_v = \{(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i)\}_{i=1}^N$. Here, we denote by D_v the vicinal set consisting of pairs of augmented training samples and corresponding target labels. Specifically, it augments training pairs by linearly interpolating the original ones as follows:

$$\begin{aligned} \tilde{\mathbf{x}} &= \lambda \mathbf{x}_i + (1-\lambda)\mathbf{x}_j, \\ \tilde{\mathbf{y}} &= \lambda \mathbf{y}_i + (1-\lambda)\mathbf{y}_j, \end{aligned} \tag{6}$$

where two pairs of input samples and target labels, $(\mathbf{x}_i, \mathbf{y}_i)$ and $(\mathbf{x}_j, \mathbf{y}_j)$, are randomly drawn from the training set D, and $\lambda \in [0, 1] \sim \text{Beta}(\alpha, \alpha)$ is a mixing coefficient, sampled from a Beta distribution. Mixup assumes that target labels for the vicinal set should also be interpolated with the same mixing coefficient as input samples during the mixing process. The hyperparameter α controls the strength of interpolation between the pairs of input samples and target labels. According to [49], the value of α within the range of $[0.1, 0.4]$ provides better classification performance, while higher values cause significant underfitting. We will also examine the effect of α in terms of confidence calibration within our framework.

3.2. RankMixup

Mixup provides smooth decision boundaries that gradually shift from one class to another, which leads to a regularized estimate of uncertainty (or confidence) [49]. There have been lots of studies to discover underlying principles behind the mixup method [15, 41, 50, 51, 51]. For example, it has proven to give better output probabilities in terms of model calibration, alleviating overconfident predictions [35, 41, 51]. The interpolated labels of augmented samples in the vicinal set do not always model the actual distribution of label mixtures in augmented samples [3, 19, 39, 43], which however has not been considered for designing a network calibration method. To alleviate this problem, we leverage ordinal ranking relationships between confidences from raw and augmented samples, as well as those among confidences from multiple augmented samples. We then propose to use them as supervisory signals for model calibration, rather than using the mixture of labels directly. In the following, we describe our framework with a single mixup-augmented sample, and present a generalized version that can handle multiple augmented samples.

Single mixup-augmented sample. The sample augmented by mixup $\tilde{\mathbf{x}}$ in Eq. (6) can be interpreted as a combination of the raw sample $\mathbf{x}_i$ and structured noise from another one $\mathbf{x}_j$, where the degree of noise is determined by the mixing parameter λ. This suggests that DNNs predict the class of the augmented sample with uncertainty imposed by the additional one, making it more difficult to recognize the augmented sample than the raw one. To this end, mixup imposes uncertainty on the label mixtures by adjusting confidences of target labels for the augmented samples by λ (or $1-\lambda$). Albeit simple, directly using the interpolated labels may provide misleading supervisory signals to train DNNs. To avoid this, we leverage an ordinal ranking relationship between confidences of raw and augmented samples. Specifically, the ordinal ranking holds for the relationship that confidence levels of easy-to-classify samples should be higher than the ones for hard samples [28]. Accordingly, we conjecture that the confidence levels of raw samples should be greater than those of the augmented samples in the mixing process (Fig.1). Namely, every pair of samples $(\mathbf{x}_i, \tilde{\mathbf{x}}_i)$ should satisfy the following relationship:

$$\max_k p_{i,k} \geq \max_k \tilde{p}_{i,k}, \tag{7}$$

where $\tilde{p}_{i,k}$ is a predicted probability of the class k for the augmented sample $\tilde{\mathbf{x}}_i$. We devise MRL that encourages the network to learn this relationship without the mixture of labels, which will be described in Sec. 3.3.

Multiple mixup-augmented samples. We extend our framework to exploit multiple mixup-augmented samples. As the mixing ratio λ decreases, augmented samples become more difficult to recognize, which suggests that we can exploit more complex ranking relationships among the multiple augmented samples. Analogous to the single case, we expect that the samples augmented with larger mixing coefficients λ show higher confidences compared to those with smaller coefficients (Fig. 1). For every pair of augmented samples $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)$ from the vicinal set D_v, the confidences of these samples should satisfy the following relationship:

$$\lambda_i \geq \lambda_j \Leftrightarrow \max_k \tilde{p}_{i,k} \geq \max_k \tilde{p}_{j,k}, \tag{8}$$

where (λ_i, λ_j) is a pair of mixing ratios for the corresponding pair of augmented samples $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)$. Note that we describe the relationship with two samples for simplicity, but our framework can adopt the relationship with any number of augmented samples.

We then integrate both the relationships in Eqs. (7) and (8) to obtain a general version of RankMixup. With the mixing coefficient of 1.0 for the raw sample, the following relationship should hold for every triplet of samples $(\mathbf{x}_i, \tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)$[1]:

$$1.0 \geq \lambda_i \geq \lambda_j \Leftrightarrow \max_k p_{i,k} \geq \max_k \tilde{p}_{i,k} \geq \max_k \tilde{p}_{j,k}. \tag{9}$$

This allows the network to estimate the levels of confidence better by considering diverse ordinal ranking relationships among the confidences of the raw and multiple mixup-augmented samples. To implement this idea, we align the orders of confidences and mixing ratios during training using M-NDCG (Sec. 3.3).

[1] Note that Eq. (9) holds only when $\lambda \geq 0.5$.

3.3. Training loss

To train our model, we use a CE loss for classification, together with either MRL or M-NDCG for network calibration, balanced by the parameter of w. As supervisory signals, we exploit one-hot encoded labels for classification, and ordinal ranking relationships for input and augmented samples for network calibration.

MRL. We introduce MRL facilitating the ordinal relationship in Eq. (7). It encourages confidences of augmented samples to be lower than those of raw samples by a margin, defined as follows:

$$\mathcal{L}_{\text{MRL}} = \max(0, \max_k \tilde{p}_{i,k} - \max_k p_{i,k} + m), \tag{10}$$

where the margin m determines acceptable discrepancies between the confidences of raw and augmented samples. The intuition behind using a margin is twofold: (1) It enables maintaining the ranking relationship between confidences, rather than penalizing the exact value of the discrepancies itself. (2) Controlling the confidence discrepancies with a margin allows to consider the uncertainty of inaccurate label mixtures. The vanilla mixup method continues to penalize the confidences of augmented samples, until they reach overconfident values, determined by the label mixtures. In contrast to this, MRL penalizes the confidences, only when the absolute discrepancies between them are below the margin.

M-NDCG. MRL is not sufficient to model the relationship in Eq. (9), where the comparison involves more than one augmented sample. M-NDCG addresses this problem that enables multiple confidence values to hold the ordinal ranking relationships. M-NDCG is based on NDCG used for evaluating ranking quality in information retrieval [17, 44]. It measures how many of retrieved results are aligned with the ground-truth ranking order, which can be calculated by normalizing a discounted cumulative gain (DCG) with an ideal DCG (IDCG):

$$NDCG = \frac{DCG}{IDCG}. \tag{11}$$

DCG accumulates a relevance score of each retrieval result, weighted by a corresponding rank, and IDCG is computed similarly but with the ground-truth score, as follows:

$$\begin{aligned} DCG &= \sum_{q=1}^{Q} \frac{r_q}{\log_2(q+1)}, \\ IDCG &= \sum_{q=1}^{Q} \frac{r_q^{\text{gt}}}{\log_2(q+1)}, \end{aligned} \tag{12}$$

where r_q and r_q^{gt} are the relevance and ground-truth scores, respectively, at position q in the result list of size Q.

We consider confidences of predictions and mixing coefficients as the relevance and ground-truth scores, respectively, enabling multiple confidences to satisfy the relationship in Eq. (9). Given Q pairs of confidences and mixing coefficients, we compute DCG_{M} and $IDCG_{\text{M}}$, as follows:

$$\begin{aligned} DCG_{\text{M}} &= \frac{\max_k p_{t,k}}{\log_2(t+1)} + \sum_{q=1, q\neq t}^{Q} \frac{\max_k \tilde{p}_{q,k}}{\log_2(q+1)}, \\ IDCG_{\text{M}} &= 1 + \sum_{q=2}^{Q} \frac{\lambda_q^{\text{gt}}}{\log_2(q+1)}, \end{aligned} \tag{13}$$

where t is the position of the confidence value for the raw sample, which is set to 1 in order to ensure that its confidence is always ranked first in a descending order, following the ranking relationship in Eq. (9). Q is a hyper-parameter that determines the number of augmented samples, *i.e.*, $Q-1$ augmented samples. λ_q^{gt} is the mixing coefficient sorted based on its magnitude, with a value of 1.0 for the raw sample appearing at position of the top rank, *i.e.*, $\lambda_1^{\text{gt}} = 1$. We then define M-NDCG loss as follows:

$$\mathcal{L}_{\text{M-NDCG}} = 1 - \frac{DCG_{\text{M}}}{IDCG_{\text{M}}}, \tag{14}$$

which is minimized when the ranking orders of the confidences and mixing coefficients are aligned to each other after sorting in a descending ordeer. The network is thus trained to encourage the confidences to hold the ordinal ranking relationship of RankMixup in Eq. (9). M-NDCG also discounts lower-ranked confidences more, preventing the uncertain predictions from being overconfident.

4. Experiments

4.1. Implementation details

Datasets. We mainly perform experiments on CIFAR10/100 [20], Tiny-ImageNet [23] for network calibration. The CIFAR10/100 datasets consist of 50,000 training and 10,000 test images of size 32×32, and provide 10/100 object categories, respectively. Tiny-ImageNet includes 120,000 images of size 64×64 for 200 object categories. Each category in Tiny-ImageNet contains 600 images, which are divided into 500, 50 and 50 images for training, validation and test, respectively. Tiny-ImageNet is more challenging, compared to CIFAR10/100, as it contains lots of object classes with a smaller number of samples per category. We further validate the generalization ability of our method on ImageNet [6], which comprises around 1.2M training and 50K validation images across 1K classes. For out-of-distribution (OOD) detection, we train a model

Table 1: Quantitative comparison with the state of the art in terms of ECE (%) and AECE (%) with 15 bins on the validation splits of CIFAR10/100 [20] and Tiny-ImageNet [23]. Numbers in bold indicate the best performance and underscored ones indicate the second best. Numbers in parentheses represent the results obtained using a TS post-hoc technique [10].

	CIFAR10 [20]						CIFAR100 [20]						Tiny-ImageNet [23]					
Methods	ResNet-50 [13]			ResNet-101 [13]			ResNet-50 [13]			ResNet-101 [13]			ResNet-50 [13]			ResNet-101 [13]		
	Acc↑	ECE↓	AECE↓	Acc↑	ECE↓	AECE↓	Acc↑	ECE↓	AECE↓	Acc↑	ECE↓	AECE↓	Acc↑	ECE↓	AECE↓	Acc↑	ECE↓	AECE↓
CE	**95.38**	3.75(0.97)	2.98(1.01)	94.46	3.61(0.92)	3.55(0.85)	77.81	13.59(2.93)	13.54(2.86)	77.48	12.94(2.63)	12.94(2.66)	64.34	3.18(3.18)	2.87(2.87)	66.04	3.50(3.50)	3.52(3.52)
MMCE [22]	95.18	3.88(0.97)	3.88(1.12)	94.99	3.88(1.15)	3.88(12.9)	77.56	12.72(2.83)	12.71(2.86)	77.82	13.43(3.06)	13.42(2.80)	64.80	2.03(2.03)	1.97(1.97)	**66.44**	3.40(3.40)	3.38(3.38)
ECP [34]	94.75	4.01(1.06)	3.99(1.53)	93.97	4.41(1.72)	4.40(1.70)	76.20	12.29(2.08)	12.28(2.22)	76.81	13.43(2.92)	13.42(3.04)	64.88	1.94(1.94)	1.95(1.95)	66.20	2.72(2.72)	2.70(2.70)
LS [30]	94.87	3.27(1.58)	3.67(3.02)	94.18	3.35(1.51)	3.85(3.10)	76.45	6.73(4.23)	6.54(4.26)	76.91	7.99(4.38)	7.87(4.55)	65.46	3.21(2.51)	3.23(2.51)	65.52	3.11(2.51)	2.92(2.72)
FL [24]	94.82	3.42(1.07)	3.41(0.87)	93.59	3.27(1.12)	3.23(1.37)	76.41	**2.83**(1.66)	2.88(1.73)	76.12	**3.10**(2.58)	**3.22**(2.51)	63.08	2.03(2.03)	1.94(1.94)	64.02	2.18(2.18)	2.09(2.09)
Mixup [41]	94.76	2.86(1.37)	2.81(2.00)	95.50	6.87(1.18)	6.79(2.33)	**78.47**	8.68(2.14)	8.68(2.19)	**78.74**	8.92(3.69)	8.91(3.65)	**65.81**	1.92(1.92)	1.96(1.96)	66.41	2.41(1.97)	2.43(1.95)
FLSD [29]	94.77	3.86(0.83)	3.74(0.96)	93.26	3.92(0.93)	3.67(0.94)	76.20	2.86(2.86)	**2.86**(2.86)	76.61	3.29(2.04)	3.25(1.78)	63.56	1.93(1.93)	1.98(1.98)	64.02	1.85(1.85)	1.81(1.81)
CRL [28]	95.08	3.14(0.96)	3.11(1.25)	95.04	3.74(1.12)	3.73(2.03)	77.85	6.30(3.43)	6.26(3.56)	77.60	7.29(3.32)	7.14(3.31)	64.88	1.65(2.35)	1.52(2.34)	65.87	3.57(1.60)	3.56(**1.52**)
CPC [4]	95.04	5.05(1.89)	5.04(2.60)	**95.36**	4.78(1.52)	4.77(2.37)	77.23	13.29(3.74)	13.28(3.82)	77.50	13.32(2.96)	13.28(3.23)	65.70	3.41(3.41)	3.42(3.42)	**66.44**	3.93(3.93)	3.74(3.74)
MbLS [25]	95.25	1.16(1.16)	3.18(3.18)	95.13	**1.38**(1.38)	3.25(3.25)	77.92	4.01(4.01)	4.14(4.14)	77.45	5.49(5.49)	6.52(6.52)	64.74	1.64(1.64)	1.73(1.73)	65.81	1.62(1.62)	**1.68**(1.68)
RegMixup [35]	94.68	2.76(0.98)	2.67(0.92)	95.03	4.75(0.92)	4.74(0.94)	76.76	5.50(1.98)	5.48(1.98)	76.93	4.20(1.36)	4.15(1.92)	63.58	3.04(1.89)	3.04(1.81)	63.26	3.35(1.86)	3.32(1.68)
RankMixup (MRL)	94.59	**1.01**(0.84)	**1.03**(0.80)	94.18	2.86(1.08)	**2.87**(1.58)	76.65	3.52(2.40)	3.39(2.05)	77.26	4.41(1.68)	4.24(1.76)	64.54	1.70(1.70)	1.69(1.69)	64.80	2.26(1.78)	2.28(1.60)
RankMixup (M-NDCG)	94.88	2.59(**0.57**)	2.58(**0.52**)	94.25	3.24(**0.65**)	3.21(**0.56**)	77.11	3.46(**1.49**)	3.45(**1.42**)	76.46	3.49(**1.10**)	3.49(**1.40**)	64.97	**1.49(1.49)**	**1.44(1.44)**	64.89	**1.57(1.57)**	1.94(1.94)

using CIFAR10, and use CIFAR100, Tiny-ImageNet, and SVHN [31] as OOD datasets, following the experimental protocol in [26, 35]. Similarly, CIFAR100 is used to train a model, and evaluate it on CIFAR10, Tiny-ImageNet, and SVHN.

Evaluation protocols. Following the standard protocol [4, 14, 25, 29, 30, 41], we use expected calibration error (ECE) and adaptive ECE (AECE) as evaluation metrics for network calibration. ECE measures the expected absolute difference between confidences and corresponding accuracies:

$$\mathbb{E}\left[|P(\hat{y}_i = y_i|\hat{p}_i) - \hat{p}_i|\right]. \tag{15}$$

To compute ECE, we divide a range $(0, 1]$ into equidistance bins, and compute an average of the absolute differences, weighted by the size of each bin. We set the number of bins for both ECE and AECE as 15, following the protocol. We also report top-1 classification accuracies to compare discriminative capabilities. We provide results obtained without/with applying a post-hoc technique. In particular, we use the TS method [10], where the temperature is optimized on the validation sets. For OOD detection, we compare the area under the receiver operating characteristics (AUROC), following the standard evaluation protocol [26, 35].

Methods for comparison. We compare our method with various training-time approaches. In addition to the baseline model using the CE loss, we include implicit methods (LS [30], FL/FLSD [29], Mixup [41], mbLS [25], and RegMixup [35]) and explicit methods (ECP [34], MMCE [22], and CPC [4]). We have reproduced results using official codes with hyperparameter settings provided by the authors, if they are not available. Detailed descriptions for the hyperparameters can be found in the supplement.

Training details. Following [25], we train our network for 200 epochs for CIFAR10/100 [20], with an initial learning rate of 1e-1 and a batch size of 128. We also follow the training setting in [25] for Tiny-ImageNet [23] except for a batch size of 32 due to memory limits. All networks are trained using the SGD optimizer with a momentum of 0.9. The learning rate decreases by a tenth at every fixed intervals. We use ResNet50 [13], and ResNet101 [13] for both CIFAR10/100 [20] and Tiny-ImageNet [23], and ResNet50 for ImageNet [6]. For more comprehensive experiments, we also exploit WideResNet [48] for CIFAR10 and Tiny-ImageNet. We fix the balancing parameter of training loss w to 0.1. We set margin m to 2.0 and 1.0 for CIFAR10/100 and Tiny-ImageNet, respectively, using each validation dataset. Other hyperparameters, such as α and Q, are chosen similarly. For OOD detection, we train the network using M-NDCG and in-distribution (ID) datasets with the same training settings used for network calibration.

Table 2: Quantitative comparison with the state of the art in terms of ECE (%) and AECE (%) with 15 bins on the validation split of ImageNet [6]. Numbers in bold indicate the best performance and underscored ones indicate the second best.

	ImageNet [6]		
Methods	ResNet-50 [13]		
	Acc↑	ECE↓	AECE↓
CE	73.96	9.10	9.24
Mixup [41]	**75.84**	7.07	7.09
CRL [28]	73.83	8.47	8.47
MbLS [25]	75.39	4.07	4.14
RegMixup [35]	75.64	5.34	5.42
RankMixup (MRL)	75.27	5.70	5.71
RankMixup (M-NDCG)	74.86	**3.93**	**3.92**

4.2. Results

Comparison with the state of the art. We compare in Table 1 our method with state-of-the-art calibration approaches. From the tables, we observe four things: (1) Our models outperform other mixup-based methods [35, 41] using label mixtures as supervisions, demonstrating the effectiveness of our approach to leveraging an ordinal ranking relationship for confidences. (2) Our model using M-NDCG achieves the best performance in terms of both ECE and AECE on the challenging Tiny-ImageNet [23] dataset, while providing competitive results on CIFAR10/100. This suggests that the ordinal ranking

Table 3: Quantitative comparison in terms of ECE (%) and AECE (%) with 15 bins on the validation splits of CIFAR10 [20] and Tiny-ImageNet [23]. Numbers in bold indicate the best performance and underscored ones are the second best.

Methods	CIFAR10 [20]			Tiny-ImageNet [23]		
	Wide-ResNet-26-10 [48]					
	Acc↑	ECE↓	AECE↓	Acc↑	ECE↓	AECE↓
CE	95.80	2.70	2.66	65.18	6.08	6.06
Mixup [41]	**96.53**	3.14	3.08	**66.36**	3.77	3.75
MbLS [25]	95.70	**1.45**	2.78	65.30	**2.57**	**2.32**
RegMixup [35]	95.44	4.18	3.99	63.40	3.87	3.93
RankMixup (MRL)	95.57	1.70	**1.38**	65.34	3.30	3.28
RankMixup (M-NDCG)	95.73	1.62	1.53	65.56	3.83	3.94

Table 4: Quantitative comparison with the state of the art in terms of AUROC (%). Numbers in bold indicate the best performance and underscored ones indicate the second best.

ID	C10 [20]			C100 [20]		
OOD	C100 [20]	Tiny [23]	SVHN [31]	C10 [20]	Tiny [23]	SVHN [31]
	AUROC↑			AUROC↑		
TS [10]	86.73	88.06	92.14	76.38	80.12	82.41
LS [30]	75.23	76.29	74.23	72.92	77.54	72.33
FL [29]	85.63	86.84	92.66	78.37	81.23	**86.37**
Mixup [41]	82.07	84.09	79.72	78.46	**81.55**	76.21
CPC [4]	85.28	85.15	**92.79**	74.68	76.66	71.72
MbLS [25]	86.20	87.55	89.98	73.88	79.19	73.19
RegMixup [35]	**87.82**	87.54	83.15	76.00	80.72	75.05
RankMixup (M-NDCG)	**87.82**	**88.94**	92.62	**78.64**	80.67	85.42

relationships among confidences is crucial for network calibration, and M-NDCG effectively encourages our model to provide confidences maintaining the relationships. We can also see that our method generalizes well to the large-scale dataset. (3) Our method outperforms another ranking-based method (CRL)[28] significantly, suggesting that more complex ranking relationship based on augmented samples enables estimating better confidences. (4) Ours with the post-hoc method [10] gives the best results in most settings, which indicates that they are complementary to each other, and exploiting both further boosts the calibration accuracy.

We also provide in Tables 2 and 3 quantitative results on ImaegeNet [6] and for the WideResNet backbone [48], respectively. They show similar trends to those in Table 1, validating the effectiveness of our approach across various network architectures and large-scale datasets. Specifically, our models provide competitive results with the state of the art [25], while outperforming mixup-based methods [35, 41] and the ranking-based one [28].

OOD detection. We compare in Table 4 the OOD detection accuracy using the ResNet-50 [13] architecture. We use the entropy of the softmax output as an uncertainty measure for binary classification and report AUROC scores for comparison. From the table, we have two findings. Our model trained with M-NDCG outperforms other mixup-based methods and achieves the highest AUROC scores in most settings. This indicates that ordinal ranking relationship of confidences is also crucial for OOD detection, suggesting that models trained with M-NDCG are not only better calibrated but also perform better than other calibration methods under the distributional shifts.

Table 5: Quantitative comparison by varying the margin m. We report the ECE (%) for the models trained with MRL. Numbers in parentheses represent the results obtained using a TS post-hoc technique [10].

	CIFAR10 [20]	Tiny-ImageNet [23]
	ECE↓	ECE↓
$m = 1.0$	1.54(1.09)	1.49(1.49)
$m = 2.0$	1.01(0.84)	1.70(1.70)
$m = 3.0$	2.12(0.93)	1.54(1.54)
$m = 4.0$	3.12(1.67)	2.37(1.77)
$m = 5.0$	13.13(0.48)	2.42(1.48)

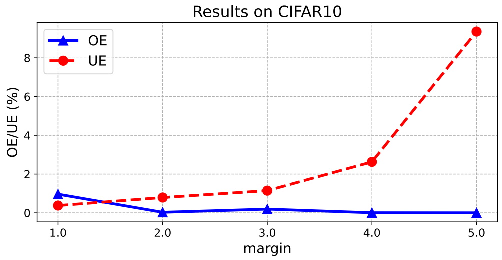

Figure 2: Quantitative results by varying the margin m. We plot OE (%) and UE (%) on the validation split of CIFAR10 [20]. Best viewed in color.

4.3. Discussion

In this section, we analyze the effects of the hyperparameters in our framework, and show generalization ability by incorporating other mixup techniques into our framework. For the analyses, we mainly adopt the ResNet-50 [13] models trained with M-NDCG, and report ECE and AECE on the validation sets of CIFAR10 [20] and Tiny-ImageNet [23], unless otherwise specified. We use 15 bins to measure the ECE and AECE.

Margin. We compare in Table 5 the calibration results of the models trained with MRL and various margins. We can see that small margins provide better performances for both datasets. Furthermore, we find that increasing the margin degrades the classification accuracy. A reason is that a large margin could make a significant difference in confidence levels between the raw and augmented samples. That is, with a larger margin, the augmented sample is more likely to have a lower confidence than its actual value.

To demonstrate this, we compute overconfidence errors (OE) [41] that measure the expected difference between confidence values of overconfident predictions and corresponding accuracies. Given H bins of equal size, OE is defined as follows [41]:

$$OE = \sum_{h=1}^{H} \frac{|B_h|}{N} \left[\mathrm{conf}(B_h) \times \max(\mathrm{conf}(B_h) - \mathrm{acc}(B_h), 0) \right], \tag{16}$$

where N is the total number of samples, and B_h is the set of samples, whose confidences fall into the h-th bin. We denote by $\mathrm{acc}(B_h)$ and $\mathrm{conf}(B_h)$ the accuracy and averaged confidence of the h-th bin, respectively. Similar to OE, we

Table 6: Quantitative results of our model with various numbers of the augmented samples. We report the ECE (%) for comparison. Numbers in parentheses represent the results obtained using a TS post-hoc technique [10].

	CIFAR10 [20]	Tiny-ImageNet [23]
	ECE↓	ECE↓
$Q = 2$	3.01(0.82)	3.25(1.44)
$Q = 3$	2.78(0.50)	2.18(1.12)
$Q = 4$	2.59(0.57)	1.49(1.49)
$Q = 5$	2.37(0.64)	1.52(1.52)
$Q = 6$	2.24(0.59)	1.30(1.30)

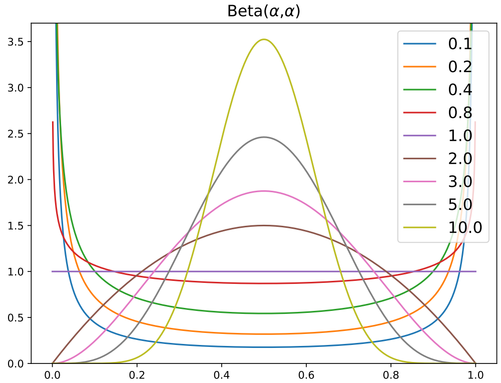

Figure 3: Probability density function (PDF) of *Beta* distribution with various αs.

define underconfidence errors (UE) computing the difference using underconfident predictions as follows:

$$UE = \sum_{h=1}^{H} \frac{|B_h|}{N} \left[\mathrm{conf}(B_h) \times \max(\mathrm{acc}(B_h) - \mathrm{conf}(B_h), 0) \right]. \tag{17}$$

We plot in Fig. 2 UE and OE for ResNet-50 with MRL on CIFAR10 [20], while varying margins. We can see that UE increases drastically, according to the margins, while OE decreases gradually. This suggests that exploiting a large margin causes underconfidence problems, where the model becomes less confident in its predictions. An underconfident model produces large calibration errors [41, 49], leading to the degraded performance.

Number of augmented samples. We compare in Table 6 the calibration performance in terms of ECE with varying number of augmented samples. We set α to 2.0 and 1.0 for CIFAR10 and Tiny-ImageNet, respectively. We can observe that ECE improves as more augmented samples are used, suggesting that learning with diverse ranking relationships between a number of augmented samples encourages the model to estimate confidence levels better. This further implies that the mixup coefficient λ is also an important factor, since it enables modeling the diverse ranking relationships. This leads us to analyze the mixup coefficient λ in detail in the following.

Mixup coefficient. Following [49], we sample the mixup coefficient λ from a *Beta* distribution, making it possible to generate various augmented samples. Note that the degree of the diversity is determined by how we set the shape parameter for the *Beta* distribution. For an in-depth analysis, we also show in Fig. 3 the probability density functions (PDFs) of different $Beta(\alpha, \alpha)$ distributions, while varying the shape parameter α. We evaluate the calibration errors of the models trained with M-NDCG along with different shape parameters α of the *Beta* distribution, and show the results using 3 augmented images ($Q = 4$) in Fig. 4. From the figures, we observe two things: (1) Our models achieve the best performances with $\alpha = 2$ and $\alpha = 1$ on CIFAR10 and Tiny-ImageNet, respectively, where the corresponding PDFs are relatively flat, as shown in Fig. 3. This suggests that diverse mixup coefficients λ drawn from such distributions can enhance the calibration performances, as it allows us to augment the samples in a more diverse way. (2) Our model performs well with large α, which is in contrast to the vanilla mixup [41, 49], where the value of α within the range of $[0.1, 0.4]$ yields better classification and calibration performances. When α is too large, it suffers from an underconfidence problem [41, 49] and is also affected by manifold intrusion [11], leading to significant performance drops in both the classification and calibration. The manifold intrusion happens when the label mixture of an augmented sample differs from the label of the real sample [11]. However, in our framework, we do not explicitly use the label mixtures as supervision, suggesting that our framework is less affected by those problems. Therefore, our framework is more robust to the variation of the parameter α compared to the vanilla mixup. To further investigate the importance of diverse samples, we also perform experiments with various combinations of α and Q and provide the results in the supplementary material.

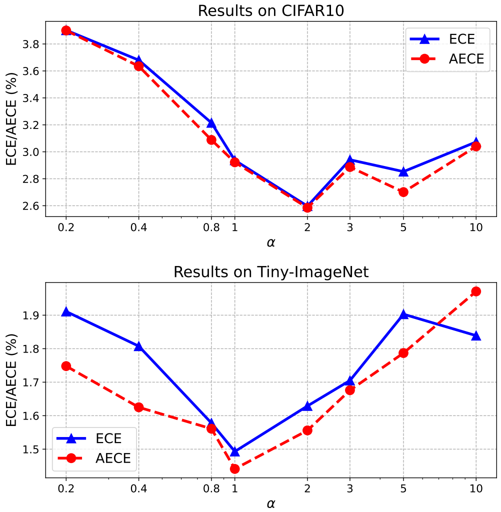

Figure 4: Quantitative results by varying the parameter of the Beta distribution (α). We plot ECE (%) and AECE (%) on the validation split of CIFAR10 [20] (*top*) and Tiny-ImageNet [23] (*bottom*). Best viewed in color.

Table 7: Quantitative comparison of calibration performances using other mixup techniques on the validation splits of CIFAR10 [20] and Tiny-ImageNet [23].

	Mixup [49]	Manifold [45]	CutMix [47]	CIFAR10 [20]		Tiny-ImageNet [23]	
				ECE↓	AECE↓	ECE↓	AECE↓
RegMixup [35]		✓		3.63	3.57	4.76	4.88
			✓	3.56	3.54	2.97	2.93
RankMixup		✓		2.92	2.98	1.20	1.22
			✓	3.26	3.21	1.80	1.66
	✓			2.59	2.58	1.49	1.44

Table 8: Quantitative comparison of classification and calibration performances on the test splits of CIFAR10-LT [1] and CIFAR100-LT [1]. We report the Top-1 accuracy (%) and ECE (%) with 15 bins.

Methods	CIFAR10-LT [1]		CIFAR100-LT [1]	
	$\rho = 10$	$\rho = 100$	$\rho = 10$	$\rho = 100$
	Acc↑/ECE↓	Acc↑/ECE↓	Acc↑/ECE↓	Acc↑/ECE↓
CE	86.39/6.60	70.36/20.53	55.70/22.85	38.32/38.23
Mixup [49]	87.10/6.55	73.06/19.20	58.02/19.69	39.54/32.72
Remix [5]	88.15/6.81	75.36/15.38	59.36/20.17	41.94/33.56
UniMix [46]	89.66/6.00	82.75/12.87	61.25/19.38	45.45/27.12
RankMixup (MRL)	88.83/2.16	74.16/ 2.03	62.84/13.55	42.56/ 5.28
RankMixup (M-NDCG)	89.80/5.94	75.41/14.10	63.83/ 9.99	43.00/18.74

Other mixup techniques. RankMixup employs a vanilla mixup technique that interpolates the input images and the corresponding labels for data augmentation, but other variants can be readily adopted in our framework. We have also tried using Manifold-Mixup [45] and CutMix [47] for the experiments. We show in Table 7 the results of our models using these techniques together with M-NDCG, while setting the shape parameter α to 1 for the both techniques. For comparison, we report the result of our model with the vanilla mixup, which exploits 3 augmented samples, and additionally show the performances obtained using RegMixup [35] equipped with these variants. For both datasets, our method achieves better calibration errors than RegMixup, indicating that our method has better generalization capability. While our model with vanilla mixup provides the best performance on CIFAR10, employing Manifold-Mixup offers better results on Tiny-ImageNet in terms of calibration errors. Increasing the entropy of supervision is crucial for better-calibrated models [25, 29, 30, 41]. RankMixup exploits the augmented samples as supervision for network calibration, indicating that samples with higher entropies can yield better performances. Manifold-Mixup [45] produces flattened representations that have higher entropies than the mixup-augmented samples, which may explain the reason why our model with Manifold-Mixup performs better on Tiny-ImageNet.

Long-tailed datasets. We evaluate our method on long-tailed (LT) datasets, where the miscalibration issues of DNNs are more pronounced [46, 52]. Following the experimental settings in [46, 52], we adopt the ResNet32 [13] architecture as our backbone, and exploit CIFAR10/100-LT datasets [1] with two imbalanced factors (ρ). We compare in Table 8 our method against the vanilla mixup [49] and two adaptive mixup techniques tailored for LT scenarios [5, 46]. We can see that our models significantly outperform other mixup methods in terms of ECE, while providing competitive accuracies, although they are not specially designed for LT scenarios. This suggests that leveraging the ordinal ranking relationship between confidences is also effective for addressing miscalibration problems in LT scenarios, instead of using label mixtures for other mixup-based augmentations [46, 52]. Note that the number of samples for head and tail classes is highly skewed, suggesting that augmented samples are not diverse in LT scenarios. As this problem is more severe for M-NDCG, it yields worse results, compared to MRL, in the LT datasets.

5. Conclusion

We have presented a novel mixup-based framework for network calibration, dubbed RankMixup, alleviating the problem that interpolated labels do not necessarily model the actual distribution of label mixtures. To this end, we have proposed a new supervisory signal based on the ordinal ranking relationship that a higher confidence is more favorable for a raw sample than the one augmented by mixup. We have proposed MRL, which encourages a network to yield confidences that preserve the ranking relationships between the raw and mixup-augmented samples. We have also extended this idea to multiple augmented samples with different mixup coefficients via M-NDCG, enabling a network to predict outputs with higher confidences for augmented samples that are closer to a raw one. We have demonstrated the effectiveness of RankMixup through the extensive experiments and analysis, achieving the best results among mixup-based network calibration approaches.

Acknowledgments. This work was supported in part by the NRF and IITP grants funded by the Korea government (MSIT) (No.2023R1A2C2004306, No.2022-0-00124, Development of Artificial Intelligence Technology for Self-Improving Competency-Aware Learning Capabilities, No.2021-0-02068, Artificial Intelligence Innovation Hub), and the Yonsei Signature Research Cluster Program of 2023 (2023-22-0008).

References

[1] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *NIPS*, 2019. 9

[2] Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. Vicinal risk minimization. In *NeurIPS*, 2000. 3

[3] Jie-Neng Chen, Shuyang Sun, Ju He, Philip HS Torr, Alan Yuille, and Song Bai. TransMix: Attend to mix for vision transformers. In *CVPR*, 2022. 2, 3, 4

[4] Jiacheng Cheng and Nuno Vasconcelos. Calibrating deep neural networks by pairwise constraints. In *CVPR*, 2022. 1, 2, 6, 7

[5] Hsin-Ping Chou, Shih-Chieh Chang, Jia-Yu Pan, Wei Wei, and Da-Cheng Juan. Remix: rebalanced mixup. In *ECCV Workshop*, 2020. 9

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 2, 5, 6, 7

[7] Zhipeng Ding, Xu Han, Peirong Liu, and Marc Niethammer. Local temperature scaling for probability calibration. In *ICCV*, 2021. 1, 2

[8] Arindam Ghosh, Thomas Schaaf, and Matthew R Gormley. AdaFocal: Calibration-aware adaptive focal loss. In *NeurIPS*, 2022. 1, 2, 3

[9] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 2020. 1

[10] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017. 1, 2, 3, 6, 7, 8

[11] Hongyu Guo, Yongyi Mao, and Richong Zhang. Mixup as locally linear out-of-manifold regularization. In *AAAI*, 2019. 8

[12] Kartik Gupta, Amir Rahimi, Thalaiyasingam Ajanthan, Thomas Mensink, Cristian Sminchisescu, and Richard Hartley. Calibration of neural networks using splines. In *ICLR*, 2021. 1

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6, 7, 9

[14] Ramya Hebbalaguppe, Jatin Prakash, Neelabh Madan, and Chetan Arora. A stitch in time saves nine: A train-time regularizing loss for improved neural network calibration. In *CVPR*, 2022. 1, 3, 6

[15] Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A simple data processing method to improve robustness and uncertainty. In *ICLR*, 2020. 4

[16] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2

[17] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002. 2, 5

[18] Xiaoqian Jiang, Melanie Osl, Jihoon Kim, and Lucila Ohno-Machado. Calibrating predictive model estimates to support personalized medicine. *Journal of the American Medical Informatics Association*, 2012. 1

[19] Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. Puzzle Mix: Exploiting saliency and local statistics for optimal mixup. In *ICML*, 2020. 2, 3, 4

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2, 5, 6, 7, 8, 9

[21] Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In *NeurIPS*, 2019. 1

[22] Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *ICML*, 2018. 2, 3, 6

[23] Ya Le and Xuan Yang. Tiny ImageNet visual recognition challenge. *CS 231N*, 2015. 2, 5, 6, 7, 8, 9

[24] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 2, 6

[25] Bingyuan Liu, Ismail Ben Ayed, Adrian Galdran, and Jose Dolz. The devil is in the margin: Margin-based label smoothing for network calibration. In *CVPR*, 2022. 1, 2, 6, 7, 9

[26] Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. In *NeurIPS*, 2020. 6

[27] Xingchen Ma and Matthew B Blaschko. Meta-Cal: Well-controlled post-hoc calibration by ranking. In *ICML*, 2021. 1, 2

[28] Jooyoung Moon, Jihyo Kim, Younghak Shin, and Sangheum Hwang. Confidence-aware learning for deep neural networks. In *ICML*, 2020. 1, 3, 4, 6, 7

[29] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating deep neural networks using focal loss. In *NeurIPS*, 2020. 1, 2, 3, 6, 7, 9

[30] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? In *NeurIPS*, 2019. 1, 2, 3, 6, 7, 9

[31] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 6, 7

[32] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, 2015. 1

[33] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In *NeurIPS*, 2019. 1, 2

[34] Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. In *ICLR Workshop*, 2017. 3, 6

[35] Francesco Pinto, Harry Yang, Ser-Nam Lim, Philip Torr, and Puneet K. Dokania. Using mixup as a regularizer can surprisingly improve accuracy & out-of-distribution robustness. In *NeurIPS*, 2022. 2, 3, 4, 6, 7, 9

[36] Jie Qin, Jiemin Fang, Qian Zhang, Wenyu Liu, Xingang Wang, and Xinggang Wang. ResizeMix: Mixing data with preserved object information and true labels. *arXiv preprint arXiv:2012.11101*, 2020. 3

[37] Sucheng Ren, Huiyu Wang, Zhengqi Gao, Shengfeng He, Alan Yuille, Yuyin Zhou, and Cihang Xie. A simple data mixing prior for improving self-supervised learning. In *CVPR*, 2022. 3

[38] Zhiqiang Shen, Zechun Liu, Zhuang Liu, Marios Savvides, Trevor Darrell, and Eric Xing. Un-mix: Rethinking image mixtures for unsupervised visual representation learning. In *AAAI*, 2022. 3

[39] Shuyang Sun, Jie-Neng Chen, Ruifei He, Alan Yuille, Philip Torr, and Song Bai. LUMix: Improving mixup by better modelling label uncertainty. *arXiv preprint arXiv:2211.15846*, 2022. 2, 3, 4

[40] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon

Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 2

[41] Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *NeurIPS*, 2019. 1, 2, 3, 4, 6, 7, 8, 9

[42] Christian Tomani, Sebastian Gruber, Muhammed Ebrar Erdem, Daniel Cremers, and Florian Buettner. Post-hoc uncertainty calibration for domain drift scenarios. In *CVPR*, 2021. 2

[43] A F M Shahab Uddin, Mst. Sirazam Monira, Wheemyung Shin, TaeChoong Chung, and Sung-Ho Bae. SaliencyMix: A saliency guided data augmentation strategy for better regularization. In *ICLR*, 2021. 2, 3, 4

[44] Hamed Valizadegan, Rong Jin, Ruofei Zhang, and Jianchang Mao. Learning to rank by optimizing ndcg measure. In *NeurIPS*, 2009. 5

[45] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold Mixup: Better representations by interpolating hidden states. In *ICML*, 2019. 3, 9

[46] Zhengzhuo Xu, Zenghao Chai, and Chun Yuan. Towards calibrated model for long-tailed visual recognition from prior perspective. In *NIPS*, 2021. 9

[47] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019. 3, 9

[48] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016. 6, 7

[49] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 2, 3, 4, 8, 9

[50] Linjun Zhang, Zhun Deng, Kenji Kawaguchi, Amirata Ghorbani, and James Zou. How does mixup help with robustness and generalization? In *ICLR*, 2021. 4

[51] Linjun Zhang, Zhun Deng, Kenji Kawaguchi, and James Zou. When and how mixup improves calibration. In *ICML*, 2022. 1, 2, 3, 4

[52] Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *CVPR*, 2021. 9