This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

On the Pitfall of Mixup for Uncertainty Calibration

Deng-Bao Wang[1], Lanqing Li[2,3,4*], Peilin Zhao[2], Pheng-Ann Heng[4], Min-Ling Zhang[1*]
[1]School of Computer Science and Engineering, Southeast University, Nanjing, China
[2]Tencent AI Lab [3]Zhejiang Lab [4]The Chinese University of Hong Kong
{wangdb,zhangml}@seu.edu.cn, lanqingli1993@gmail.com
masonzhao@tencent.com, pheng@cse.cuhk.edu.hk

Abstract

By simply taking convex combinations between pairs of samples and their labels, mixup training has been shown to easily improve predictive accuracy. It has been recently found that models trained with mixup also perform well on uncertainty calibration. However, in this study, we found that mixup training usually makes models less calibratable than vanilla empirical risk minimization, which means that it would harm uncertainty estimation when post-hoc calibration is considered. By decomposing the mixup process into data transformation and random perturbation, we suggest that the confidence penalty nature of the data transformation is the reason of calibration degradation. To mitigate this problem, we first investigate the ***mixup inference*** *strategy and found that despite it improves calibration on mixup, this ensemble-like strategy does not necessarily outperform simple ensemble. Then, we propose a general strategy named* ***mixup inference in training****, which adopts a simple decoupling principle for recovering the outputs of raw samples at the end of forward network pass. By embedding the mixup inference, models can be learned from the original one-hot labels and hence avoid the negative impact of confidence penalty. Our experiments show this strategy properly solves mixup's calibration issue without sacrificing the predictive performance, while even improves accuracy than vanilla mixup.*

1. Introduction

Although modern neural networks have made notable performance on predictive accuracy in various computer vision tasks [7], they have been found to perform poorly in terms of uncertainty calibration, which is an important consideration in many real-world applications [5]. Intuitively, we expect a predictive model to be accurate when it is confident about its outputs while reveal high uncertainty when it is likely to be inaccurate. Otherwise, the miscalibrated prediction of models could cause undesired consequences in many safety-critical applications such as medical diagnosis and autonomous driving. Early researches on uncertainty estimation mainly focus on probabilistic models. However, in deep learning paradigm, training of deep bayesian models are expensive and their performance usually depends on approximate inference methods due to the computational constraint in real-world deployment. Therefore, uncertainty calibration of deterministic neural networks becomes an important topic in recent years.

Guo *et al.* [5] systematically studied the uncertainty calibration problem of modern neural networks with comprehensive experiments. They pointed out that popular modern neural networks usually suffer from severe miscalibration issue than shallow models. They empirically showed that large model capacity without proper regularization is closely related to the miscalibration issue. They also evaluated the performance of various calibration strategies and found that simple post-hoc approaches like temperature scaling (TS) [22] and histogram binning (HB) [34] can reduce the calibration error to a quite low level. Following their work, a number of calibration friendly regularization methods and post-calibration approaches were proposed to address the miscalibration issue of deep neural networks [12, 17, 18, 21].

Recently, researchers investigated the impact of mixup training for calibration. Thulasidasan *et al.* [27] empirically found that mixup improves calibration across various model architectures and datasets. Zhang *et al.* [38] provided a theoretical explanation for the effect of mixup training on calibration in high-dimensional regime. Carratino *et al.* [3] pointed out that mixup implicitly performs label smoothing and hence can avoid the overconfidence issue. However, there are also empirical observations showing that mixup does not necessarily improve calibration. The experiments in [16] provides evidence showing mixup degrades calibration in some cases. The empirical studies in [31] and [23]

[*] Corresponding author
This work is done when Deng-Bao works as an intern in Tencent AI Lab.

found that combining mixup with ensemble degrades calibration performance than individually using one of them. In particular, they suggest that both mixup and ensembling encourage models to be less confident, and hence the under-confidence issue occurs when they are used together.

We notice that most of existing work investigates mixup for calibration without the consideration of post-calibration. As the research in [1] suggested, the comparison of calibration performance between different methods without post-calibration might not provide a fair ranking. Another recent work [30] also pointed out that models with better calibration performance during main training do not necessarily yield better calibration results after post-calibration. Therefore, in this work, we revisit mixup's calibration problem by considering the training stage and post-hoc processing as a unified system. Under this setting, three questions are naturally raised: *(i) Does mixup really help calibration? (ii) If it does not, what leads to the failure? (iii) How can we mitigate the pitfall of mixup on calibration?* To answer these questions, we make the following **contributions**:

- We conduct comprehensive experiments showing that mixup often leads to less calibratable models than vanilla empirical risk minimization (ERM), and hence degrades uncertainty estimation in general when post-calibration is considered after training.
- To explain this phenomenon, we decompose mixup into two components: data transformation and random perturbation. We show that the former part shrinks the training labels to their means and implicitly performs *confidence penalty*, which serves as the reason of calibration degradation.
- We investigate the *mixup inference* strategy for calibration and found that despite it improves calibration on mixup, this ensemble-like approach is no better than vanilla deep ensemble in terms of both calibration and accuracy with same inference budget.
- We show that mixup's calibration issue can be easily solved by translating the mixup inference into training. By this process, the output of each raw sample can be approximately recovered to be learned from the original one-hot labels, and hence avoiding the negative effect induced by confidence penalty. Our experiments show that this strategy outperforms mixup in terms of both accuracy and calibration.

2. Background

2.1. Mixup

By taking convex combinations between pairs of examples and their labels, mixup training has been shown to easily improve predictive accuracy [36]. Given a sample (x_i, y_i), mixup mixes it with sample (x_j, y_j) as

$$\begin{aligned} \widetilde{x}_i &= \lambda x_i + (1-\lambda)x_j, \\ \widetilde{y}_i &= \lambda y_i + (1-\lambda)y_j, \end{aligned} \tag{1}$$

where λ is sampled from Beta(α, α) with $\alpha > 0$, and j is sampled from Uniform$([n])$ and n denotes the dataset size. Due to its simplicity and effectiveness on generalization and robustness, mixup has become a fundamental technique in machine learning community [8, 20]. Moreover, mixup can be easily utilized to improve weakly supervised learning, such as semi-supervised learning [2, 25], noisy-label learning [15] and positive-unlabeled learning [14]. Given its impressive performance, there are also several works that investigate mixup from the theoretical perspective [3, 4, 37], which show mixup has the implicit regularization effect that enables models to better generalize. Most previous studies focus on accuracy despite some of them evaluate mixup with the calibration metrics, while the focus of our work is mainly on calibration. There are some studies propose to extend the linear interpolation to more complicated mixing process [9,33,39], which will not be discussed in this paper.

2.2. Calibration of Deep Neural Networks

In classification and regression, uncertainty calibration has been studied in a long history [19, 22, 35]. Intuitively, a well calibrated model should be confident on the prediction that is likely to be the ground-truth, while indicate high uncertainty when it is likely to be inaccurate. Formally, a perfectly calibrated model is expected to satisfy $\mathbb{P}(\hat{y} = y \,|\, \hat{p} = p) = p$ for $p \in [0, 1]$, where $\hat{y}$ and y denote the predicted and the ground-truth class respectively, while $\hat{p}$ denotes the model's confidence. In recent years, widely used deep models have been empirically found to produce poorly calibrated outputs [5]. Guo *et al.* systematically studied the calibration problem in deep learning paradigm and pointed out the importance of post-calibration for overconfident deep neural networks [22]. Following their work, there is a surge of studies that try to design new post-calibration methods [10, 11, 21, 24]. In this work, we consider the simplest post-calibration method TS due to its impressive generalization performance [5]. Besides the studies on improving calibration performance, there are also several works that focus on the evaluation metric of calibration performance [6,28,32]. Due to the space limitation, the metrics used in our experiments are described in Appendix.

Mixup for Calibration. As we discussed in Introduction, mixup has been recently studied in terms of uncertainty calibration. Most of these existing studies try to improve calibration performance with the regularization effect induced by mixup [8,16,27,38]. We notice there are some contradictory results on mixup's calibration performance in previous studies [16, 27]. Our experiments show that the contradic-

Table 1. Comparison between mixup and ERM in terms of **uncalibrated ECE**, **calibrated ECE** and the **optimal ECE**. ▲/▼ indicates that mixup outperforms/underperforms the vanilla ERM. The values reported in each entry are the results of different backbones: ResNet-18, ResNet-50, ResNet-110 and ResNet-152.

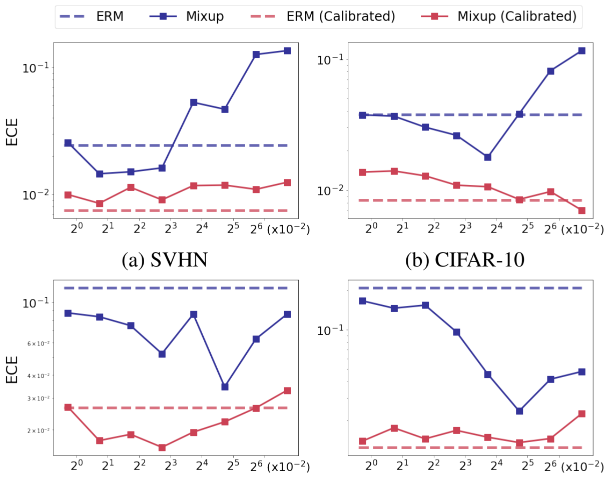
Datasets	Metrics	ERM				mixup ($\alpha=0.1$)				mixup ($\alpha=0.5$)				mixup ($\alpha=1.0$)			
SVHN	ECE	2.15	2.67	2.43	2.56	3.96	1.89▲	2.46	1.38▲	11.2	9.48	8.04	9.37	14.8	13.7	13.8	12.9
	Calibrated ECE	0.50	0.87	0.75	0.90	0.99▼	1.03▼	1.08▼	1.05▼	1.23▼	1.21▼	1.28▼	1.21▼	1.12▼	1.18▼	1.14▼	1.04▼
	Optimal ECE	0.24	0.56	0.45	0.58	0.75▼	0.74▼	0.85▼	0.68▼	1.12▼	0.95▼	0.95▼	0.88▼	1.04▼	0.98▼	0.88▼	0.78▼
CIFAR-10	ECE	3.33	3.99	3.78	3.47	2.57▲	2.22▲	2.55▲	2.53▲	6.87	6.25	6.55	6.20	12.1	11.5	10.5	11.2
	Calibrated ECE	0.65	0.79	0.83	0.65	1.04▼	1.07▼	1.08▼	1.12▼	1.15▼	1.15▼	0.95▼	1.05▼	0.94▼	0.91▼	0.83	0.76▼
	Optimal ECE	0.59	0.63	0.61	0.52	0.97▼	0.98▼	1.01▼	1.01▼	0.97▼	1.03▼	0.88▼	0.88▼	0.85▼	0.80▼	0.71▼	0.65▼
CIFAR-100	ECE	10.9	12.5	11.9	11.7	2.43▲	6.63▲	5.95▲	5.59▲	10.8▲	3.89▲	3.91▲	3.85▲	13.0	7.44▲	7.50▲	7.55▲
	Calibrated ECE	2.56	2.41	2.64	2.42	1.76	1.87	1.37	1.67	1.22	2.63▼	3.21▼	2.57▼	1.25	2.66▼	3.02▼	3.52▼
	Optimal ECE	2.45	2.29	2.44	2.31	1.60	1.59	1.23	1.45	0.98	2.46▼	3.04▼	2.39▼	1.09	2.54▼	2.85▼	3.38▼
Tiny-ImageNet	ECE	23.2	20.5	20.7	21.6	8.57▲	7.51▲	9.76▲	10.4▲	3.98▲	3.92▲	2.16▲	3.29▲	6.85▲	7.44▲	4.98▲	5.93▲
	Calibrated ECE	1.33	1.23	1.36	1.33	1.32	1.28▼	1.55▼	2.08▼	1.33	1.46▼	1.52▼	1.82▼	1.49▼	1.65▼	2.26▼	2.00▼
	Optimal ECE	1.14	1.00	1.16	1.16	1.02	1.05▼	1.40▼	1.93▼	1.08	1.21▼	1.23▼	1.60▼	1.20▼	1.30▼	1.91▼	1.69▼

tory results indeed occur based on different choice of hyperparameter α, when post-calibration is absent. Therefore, we suggest that post-calibration is essential to fairly evaluate mixup's calibration performance. To the best of our knowledge, there is no work that focuses on the pitfall of the interaction between mixup training and post-calibration.

3. Does Mixup Really Help Calibration?

As we mentioned in above sections, there are some contradictory results on mixup's calibration performance in previous studies [16,27]. We suggest that the comparison of calibration performance between different methods without post-calibration might not provide a fair ranking [1]. Therefore, in this section, we compare mixup and ERM in the presence of post-calibration, where we use TS as the post-processing tool due to its simplicity and generalization performance. TS works by replacing the temperature of softmax layer with the value yielding best calibration result on a hold-out validation set. We use expected calibration error (ECE) [5] to evaluate calibration and denote the ECE with the replaced temperature as calibrated ECE [1,2]. We can also find the temperature directly on test set and denote the result as optimal ECE, which can be considered as the lower bound of calibrated ECE and helps us identify which model is more calibratable in pos-hoc calibration stage.

Table 1 reports the comparative results between mixup and ERM with four ResNet backbones of different sizes [3]. As is shown, there are nearly 60% cases showing models trained with mixup are better than those trained with ERM in terms of raw ECE (without post-calibration). However, when post-calibration is involved, there are nearly 82% cases that mixup are worse than ERM (in terms of calibrated ECE and optimal ECE). These results demonstrate that despite that the mixup-trained models may give better calibration performance after main training, it is harder to further improve them in post-calibration stage, namely not being as calibratable as ERM-trained models. This phenomenon is even more transparent for larger models: In terms of calibrated ECE and optimal ECE, mixup outperforms ERM in 12 cases out of total 48 cases on ResNet-18 and ResNet-50, while only in 4 cases on ResNet-110 and ResNet-152.



(a) SVHN

(b) CIFAR-10

(c) CIFAR-100

(d) Tiny-ImageNet

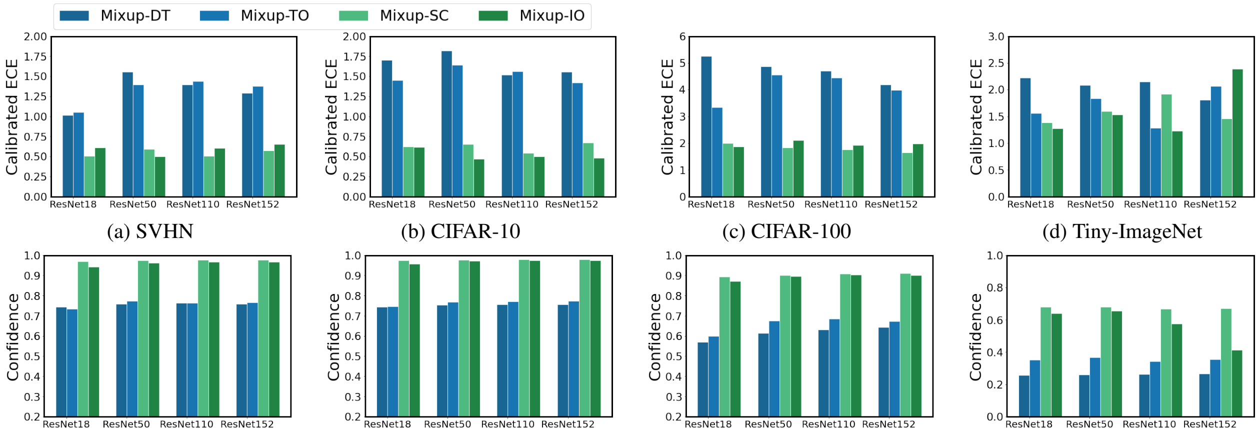
Figure 1. The comparison of mixup with different choices of α. The experiments are conducted with ResNet110.

We also notice that different coefficient α leads to different calibration performance: With $\alpha = 0.1$, mixup outperforms ERM in 14 and 5 cases out of 16 cases respec-

[1] The experiments in Section 3, 4 and 5 are also evaluated with other two metrics ACE and NLL, and the results can be found in Appendix.

[2] The reported results of all tables are the average of 3 random runs. In each run, the results of last 10 epochs are averaged as the final result.

[3] The implementation details can be found in Appendix.



(a) SVHN (b) CIFAR-10 (c) CIFAR-100 (d) Tiny-ImageNet

(e) SVHN (f) CIFAR-10 (g) CIFAR-100 (h) Tiny-ImageNet

Figure 2. The top row shows the comparison of **Calibrated ECE** between four ablated variants of mixup, where the variants with blue color use the transformed labels while the variants with green color use the original one-hot labels. The bottom row shows the comparison of **average confidence** between four ablated variants.

tively in terms of uncalibrated and calibrated ECE, while with $\alpha = 1$, mixup outperforms ERM in only 7 cases and 1 case on these two terms. Figure 1 shows the result of mixup with $\alpha \in \{0.01, 0.02, 0.04, 0.08, ..., 1.28\}$. As is shown, in the absence of post-calibration, mixup training with a proper α indeed helps calibration especially on CIFAR-100 and Tiny-ImageNet. Intuitively speaking, this improvement might be induced by the regularization effect of mixup, which implicitly penalizes the sharp outputs to avoid the overconfidence issue. However, when the regularization becomes stronger, namely larger α used in mixup, it may cause underconfidence issue, which is also a miscalibration case. The drawbacks of this confidence penalty mechanism is demonstrated in Figure 1: (1) The best α varies across datasets, which means we need to carefully choose α on a new task; (2) Even with the best α, there is still a large margin between uncalibrated and calibrated ECE. Once post-calibration is involved, mixup tends to degrade the calibration performance especially on large α (which is important to obtain desired accuracy).

In summary, there exists a dilemma between accuracy and calibration in using mixup when considering training and post-calibration as a unified system. In this study, we aim to mitigate the pitfall of mixup for calibration without sacrificing its predictive performance. Before that, let us first explain why mixup causes the failure of calibration by empirical study.

4. Why Mixup Fails on Calibration

As shown by Equation (1), mixup takes the linear interpolation between pairs of inputs and labels to create new samples. Following the result of [3], this formulation can be decomposed into two operations.

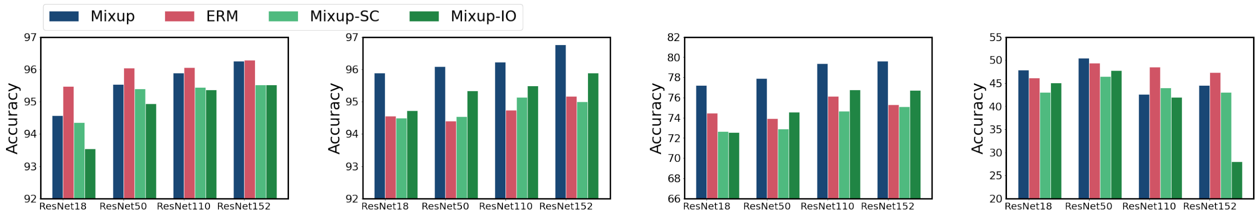
Remark 1. [3] *Let* $\lambda \sim \mathrm{Beta}_{[\frac{1}{2},1]}(\alpha, \alpha)$ *and* $j \sim$ *Uniform* $([n])$ *be two random variables with* $\alpha > 0$, $n > 0$ *and let* $\bar{\lambda} = \mathbb{E}_{\lambda}\lambda$. *The mixed sample* $(\widetilde{x}_i, \widetilde{y}_i)$ *as in Equaton (1) for any* $i \in [n]$ *can be reformulated as:*

$$\begin{aligned} \widetilde{x}_i &= \bar{x} + \bar{\lambda}(x_i - \bar{x}) + (\lambda - \bar{\lambda})x_i + (1-\lambda)x_j - (1-\bar{\lambda})\bar{x}, \\ \widetilde{y}_i &= \underbrace{\bar{y} + \bar{\lambda}(y_i - \bar{y})}_{\text{Data Transformation } x'_i, y'_i} + \underbrace{(\lambda - \bar{\lambda})y_i + (1-\lambda)y_j - (1-\bar{\lambda})\bar{y}}_{\text{Random Perturbation } \epsilon^x_i, \epsilon^y_i}, \end{aligned} \tag{2}$$

where $\bar{x}, \bar{y}$ *are the mean of inputs and labels of all training samples, and the perturbation terms satisfy* $\mathbb{E}_{\lambda,j}\epsilon^x_i = \mathbb{E}_{\lambda,j}\epsilon^y_i = 0$.

This reformulation allow us to consider mixup as the combination of two complementary components, i.e., data transformation and random perturbation. As $\bar{\lambda} < 1$, the data transformation part shrinks the inputs and labels towards their means. Assuming balanced label distribution, the label transformation is equivalent to the label smoothing technique introduced in [26]. Moreover, since the expectations of ϵ^x_i and ϵ^y_i are zero, the random perturbation terms will add zero-mean noises to each transformed input and label.

Inspired by the recent work [30] that shows the negative impact of label smoothing on calibration. We conjecture that the label transformation part of the second Equation in (2) leads to the failure on calibration. To verify this, we come up with the following four ablated variants of mixup: Mixup-DT, which only uses the **D**ata **T**ransformation part of Equation (2); Mixup-TO, which mixes between **T**arget labels **O**nly; Mixup-SC, which mixes within **S**ame **C**lass; and Mixup-IO, which mixes between **I**nputs **O**nly. The formulations of these variants are shown in Table 2. It is shown that different from the vanilla mixup and the former two variants, the latter two variants learn models from one-hot



(a) SVHN (b) CIFAR-10 (c) CIFAR-100 (d) Tiny-ImageNet

Figure 3. The predictive **accuracy** of Mixup-IO, Mixup-SC by comparing with that of vanilla mixup and ERM. The detailed results could be found in Table 4 and Table 5.

Table 2. The inputs and targets used by 4 ablated variants.

Variants	Inputs	Targets	One-hot?
Mixup-DT	$\{\bar{x}+\bar{\lambda}(x_i-\bar{x})\}$	$\{\bar{y}+\bar{\lambda}(y_i-\bar{y})\}$	✘
Mixup-TO	$\{x_i\}$	$\{\lambda y_i+(1-\lambda)y_j\}$	✘
Mixup-SC	$\{\lambda x_i+(1-\lambda)x_j \vert y_i=y_j\}$	$\{y_i\}$	✔
Mixup-IO	$\{\lambda x_i+(1-\lambda)x_j\}$	$\{y_i\}$	✔

labels. We conduct experiments on these derivations and make the following two observations on calibration and accuracy.

Confidence penalty hurts calibration. The top row of Figure 2 shows the comparative results on calibrated ECE between these four derivations. It is clearly shown that Mixup-DT and Mixup-TO achieves much larger calibration error than the others. The bottom row shows that Mixup-DT and Mixup-TO make models less confident on their prediction during training. Therefore, we can tell that although the transformation on labels helps penalize the overconfident outputs during training, it compresses the room of potential improvement in post-calibration, and hence hurt calibration performance in general.

Trivial confidence promotion hurts accuracy. Unlike the former two derivations, Mixup-SC and Mixup-IO directly use the original one-hot labels to learn models, which does not induce the confidence penalty effect, and leads to more calibratable models. Unfortunately, Figure 3 shows that the superiority of mixup on predictive performance would be eliminated when using these two derivations. As is shown, the accuracy of Mixup-SC and Mixup-IO is not consistently higher than ERM, while always lower than original mixup. Therefore, the dilemma between the accuracy and calibration can not be solved by these trivial confidence promotion stragety.

5. Mitigating the Pitfall of Mixup

5.1. Mixup Inference

Most of the existing studies only focus on exploiting mixup mechanism in the training phase, while the study in [20] found that the mixing of features at inference phase can further improve mixup-trained models' robustness against adversarial perturbations. The mixup inference approach used in their paper is quite simple: At every inference time t, they first mix the test sample x with a uniformly drawn sample x' as $\widetilde{x}_t = \lambda x + (1-\lambda)x'$, then calculate the model prediction $\widehat{y}_t = f(\widetilde{x}_t)$; After T iterations, they simply average the predictions of the mixed samples as $\bar{f}(x) = \frac{1}{T}\sum_{t=1}^{T}\widehat{y}_t$. The experiments reported in [20] demonstrate the superiority of this simple approach for adversarial robustness. However, without a *decoupling* process in output space, it is hard to obtain precise label confidence information by this simple mixup inference approach. The following remark demonstrates the decoupling principle of mixup-trained models.

Remark 2. *Recall the basic idea of mixup: linear interpolations of feature vectors should lead to linear interpolations of the output space. Based on this assumption, by mixing two samples twice with* $\lambda_1 \neq \lambda_2 \in (0,1)$*, as is*

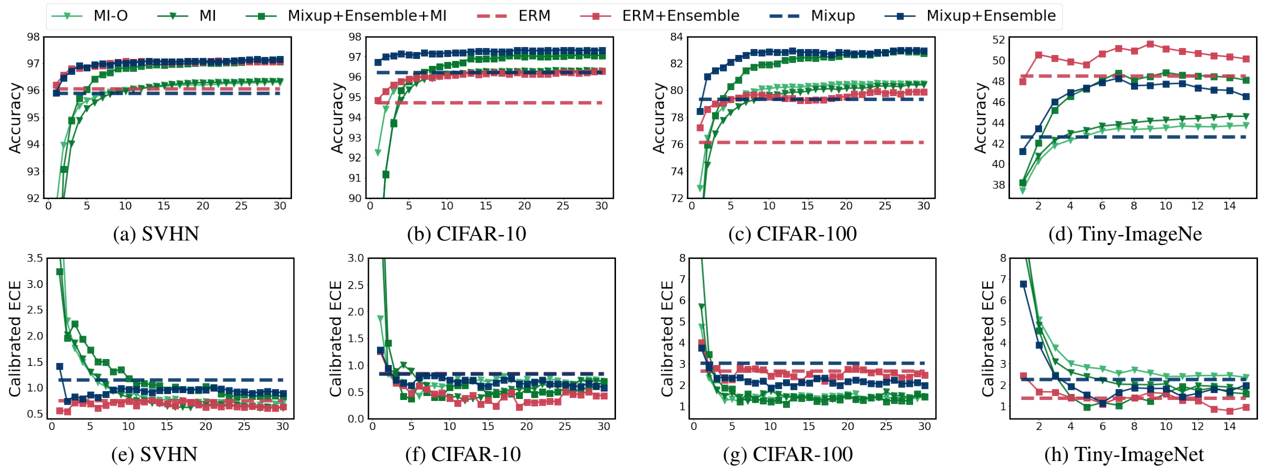
$$\begin{aligned}\widetilde{x}_1 &= \lambda_1 x_a + (1-\lambda_1)x_b,\\ \widetilde{x}_2 &= \lambda_2 x_a + (1-\lambda_2)x_b,\end{aligned} \tag{3}$$

we can decouple these two samples in outputs space:

$$\begin{aligned}\widehat{y}_a &= \frac{f(\widetilde{x}_1) - f(\widetilde{x}_2)(1-\lambda_1)/(1-\lambda_2)}{\lambda_1 - \lambda_2(1-\lambda_1)/(1-\lambda_2)},\\ \widehat{y}_b &= \frac{f(\widetilde{x}_1) - f(\widetilde{x}_2)\lambda_2/\lambda_1}{1-\lambda_2-(1-\lambda_1)\lambda_2/\lambda_1}.\end{aligned} \tag{4}$$

Considering the convex combinations of hidden representations [29], this decoupling principle could be exploited in the hidden layers of neural networks. In particular, to avoid the negative value of model confidence, we can simply adopt the decoupling process before softmax layer. We conjecture that by exploiting this decoupling principle in the mixup inference, we can improve the calibration performance of mixup-trained models. As is shown in Algorithm I in Appendix, at every inference time, our MI approach recovers the prediction of x by Equation (4), which adopts the same coefficient α as in training phase.

Table 3 shows the results of our new MI strategy on both predictive accuracy and calibration. The implementation of



(a) SVHN (b) CIFAR-10 (c) CIFAR-100 (d) Tiny-ImageNe

(e) SVHN (f) CIFAR-10 (g) CIFAR-100 (h) Tiny-ImageNet

Figure 4. The comparison between our mixup inference approaches and different ensemble strategies. For comparison, we also present the results of individual ERM-trained and mixup-trained models with dashed lines. The experiments are conducted with ResNet110.

Table 3. The comparison between mixup inference and mixup. ▲/▼ indicates that MI-O/MI outperform/underperform mixup.

Datasets	Backbones	Mixup		MI-O		MI	
		Acc	ECE	Acc	ECE	Acc	ECE
SVHN	ResNet18	94.5	1.12	95.0	0.45▲	94.9	0.70▲
	ResNet50	95.5	1.18	95.7	0.56▲	95.7	0.60▲
	ResNet110	95.8	1.14	96.1	0.77▲	96.1	0.64▲
	ResNet152	96.2	1.04	96.5	0.61▲	96.5	0.71▲
CIFAR-10	ResNet18	95.8	0.94	95.7	0.69▲	95.9	0.63▲
	ResNet50	96.0	0.91	95.9	0.65▲	95.9	0.62▲
	ResNet110	96.2	0.83	96.2	0.68▲	96.2	0.46▲
	ResNet152	96.7	0.76	96.6	0.57▲	96.6	0.45▲
CIFAR-100	ResNet18	77.2	1.25	78.1	1.21▲	78.2	1.25
	ResNet50	77.8	2.66	78.6	1.38▲	78.6	1.61▲
	ResNet110	79.3	3.02	80.3	1.45▲	79.9	1.39▲
	ResNet152	79.6	3.52	80.5	1.23▲	80.2	1.67▲
Tiny-ImageNet	ResNet18	47.8	1.49	50.6	1.28▲	50.2	1.44▲
	ResNet50	50.4	1.65	52.5	1.92▼	52.1	1.89▼
	ResNet110	42.6	2.26	43.7	2.35▼	44.6	1.74▲
	ResNet152	44.6	2.00	44.8	2.93▼	45.5	2.25▼

MI is same with the pseudo-code shown in Appendix, while MI-O fixes $\lambda_2 = 0$. In MI-O, $\widetilde{y}_2$ can be collected before the testing phase, and hence only one single forward pass is needed in each iteration. As is shown, both of these two mixup inference approaches improve the calibration performance of mixup, while slightly improve the predictive accuracy in partial cases. Despite the improvement, we have to face the computational cost and time consuming problem during inference phase (the results of Table 3 are achieved by 15 iterations per sample). As is studied in previous work, if do not consider the inference time, simple ensemble of independently-trained networks is a good choice in terms of both accuracy and calibration [1, 13, 31]. Therefore, a realistic question is raised: *Is this ensemble-like strategy better than vanilla deep ensemble?* Similar problem has been investigated in [1], which found that popular ensemble-like approaches require dozens of ensemble members to achieve equivalent performance of the ensemble of only few independently trained models.

The comparison between MI and ensemble is presented in Figure 4. We compare our MI approaches with ensembles of models independently trained with ERM or mixup ($\alpha = 1$). We also make MI interact with ensemble, where we use different independently trained models among inference iterations. The top row shows the comparison on accuracy. It is clearly shown that ensemble of mixup-trained models outperforms MI and the combination of MI and ensemble. Even simple ensemble of ERM-trained models achieves comparable results compare with MI. The comparison on calibrated ECE in the bottom row, as well as on calibrated NLL in appendix, also shows similar phenomenon that MI do not necessarily outperform ensemble.

5.2. Mixup Inference in Training

As we suggested in Section 4, the confidence penalty property of mixup seems to be the reason of the degradation on calibration. On the contrary, Mixup-SC and Mixup-IO, which promote model confidence by directly learning on the original one-hot labels, instead achieve very good calibration performance. This phenomenon inspires us to design a new strategy that only preserves the data augmentation nature of mixup, but do not penalize the model confidence during training.

We propose to achieve this goal by simply translating the mixup inference process into the training. With the inference process, the output of each raw sample could be approximately recovered. Then, one can learn models by

Table 4. The overall comparative results in terms of **calibrated ECE**. The number in each bracket indicates the ranking across all methods. The orange/blue color indicates that a method outperforms/underperforms ERM in average. The **boldface** and underline denote the best and the second best results of each row. The marker † means the backbone is pretrained.

	Backbones	ERM	Mixup (0.1)	Mixup (0.5)	Mixup (1.0)	Mixup (DT)	Mixup (TO)	Mixup (SC)	Mixup (IO)	**MIT-A**	**MIT-L** ($\Delta\lambda>\frac{1}{2}$)	**MIT-A** ($\Delta\lambda>\frac{1}{2}$)
SVHN	ResNet18	0.50 (2)	0.99 (7)	1.23 (11)	1.12 (10)	1.01 (8)	1.05 (9)	0.50 (3)	0.61 (5)	**0.47 (1)**	0.65 (6)	0.53 (4)
	ResNet50	0.87 (6)	1.03 (7)	1.21 (9)	1.18 (8)	1.55 (11)	1.39 (10)	0.59 (4)	0.50 (2)	**0.49 (1)**	0.52 (3)	0.66 (5)
	ResNet110	0.75 (6)	1.08 (7)	1.28 (9)	1.14 (8)	1.39 (10)	1.43 (11)	0.50 (2)	0.60 (4)	**0.48 (1)**	0.53 (3)	0.70 (5)
	ResNet152	0.90 (6)	1.05 (8)	1.21 (9)	1.04 (7)	1.28 (10)	1.37 (11)	0.57 (2)	0.65 (4)	0.61 (3)	**0.53 (1)**	0.67 (5)
	Avg. gain	—	+ 0.28	+ 0.47	+ 0.36	+ 0.55	+ 0.55	– 0.21	– 0.16	– 0.24	– 0.19	– 0.11
CIFAR-10	ResNet18	0.65 (6)	1.04 (8)	1.15 (9)	0.94 (7)	1.70 (11)	1.45 (10)	0.62 (4)	0.61 (3)	**0.56 (1)**	0.59 (2)	0.62 (5)
	ResNet50	0.79 (6)	1.07 (8)	1.15 (9)	0.91 (7)	1.81 (11)	1.64 (10)	0.65 (4)	**0.46 (1)**	0.63 (3)	0.59 (2)	0.68 (5)
	ResNet110	0.83 (7)	1.08 (9)	0.95 (8)	0.83 (6)	1.52 (10)	1.56 (11)	0.54 (3)	**0.50 (1)**	0.52 (2)	0.54 (4)	0.78 (5)
	ResNet152	0.65 (4)	1.12 (9)	1.05 (8)	0.76 (7)	1.55 (11)	1.42 (10)	0.67 (5)	**0.48 (1)**	0.57 (3)	0.50 (2)	0.67 (6)
	Avg. gain	—	+ 0.34	+ 0.34	+ 0.13	+ 0.91	+ 0.78	– 0.11	– 0.21	– 0.15	– 0.17	– 0.04
CIFAR-100	ResNet18	2.56 (9)	1.76 (5)	**1.22 (1)**	1.25 (2)	5.24 (11)	3.33 (10)	2.00 (7)	1.87 (6)	1.44 (3)	2.18 (8)	1.75 (4)
	ResNet50	2.41 (7)	1.87 (2)	2.63 (8)	2.66 (9)	4.86 (11)	4.55 (10)	**1.82 (1)**	2.10 (5)	1.90 (3)	2.15 (6)	1.97 (4)
	ResNet110	2.64 (7)	**1.37 (1)**	3.21 (9)	3.02 (8)	4.70 (11)	4.45 (10)	1.76 (2)	1.93 (3)	1.98 (4)	2.25 (6)	2.00 (5)
	ResNet152	2.42 (6)	1.67 (2)	2.57 (8)	3.52 (9)	4.19 (11)	3.97 (10)	**1.65 (1)**	1.98 (4)	1.71 (3)	2.47 (7)	2.17 (5)
	Avg. gain	—	– 0.84	– 0.10	+ 0.10	+ 2.24	+ 1.56	– 0.69	– 0.53	– 0.74	– 0.24	– 0.53
Tiny-ImageNet	ResNet18	1.33 (4)	1.32 (2)	1.33 (3)	1.49 (8)	2.22 (11)	1.55 (10)	1.38 (5)	**1.30 (1)**	1.47 (7)	1.54 (9)	1.41 (6)
	ResNet50	1.23 (3)	1.28 (4)	1.46 (6)	1.65 (9)	2.08 (11)	1.83 (10)	1.59 (8)	1.58 (7)	**1.18 (1)**	1.23 (2)	1.36 (5)
	ResNet110	1.36 (4)	1.55 (8)	1.52 (7)	2.26 (11)	2.14 (10)	**1.28 (1)**	1.92 (9)	1.49 (6)	1.35 (3)	1.29 (2)	1.39 (5)
	ResNet152	1.33 (3)	2.08 (10)	1.82 (7)	2.00 (8)	1.81 (6)	2.06 (9)	1.46 (5)	2.19 (11)	1.43 (4)	**1.17 (1)**	1.30 (2)
	ResNet18†	1.12 (2)	1.43 (5)	1.22 (3)	1.31 (4)	2.83 (11)	1.90 (10)	1.58 (7)	1.72 (8)	1.56 (6)	1.79 (9)	**1.11 (1)**
	ResNet152†	1.96 (6)	1.57 (4)	2.75 (9)	2.74 (8)	4.83 (10)	6.60 (11)	**1.19 (1)**	1.68 (5)	1.37 (3)	2.58 (7)	1.26 (2)
	Avg. gain	—	+ 0.14	+ 0.29	+ 0.51	+ 1.26	+ 1.14	+ 0.13	+ 0.27	0.00	+ 0.21	– 0.08

fitting the decoupled outputs to the original one-hot labels and hence avoid the confidence penalty effect caused by label smoothing. This procedure is illustrated in Figure A in the Appendix. In practice, the decoupling process may induce noise if one of the mixed example is poorly learned. We suggest that a large margin between λ_1 and λ_2 can reduce this noise, and we can achieve this by sampling λ_1 and λ_2 from $\text{Beta}_{[0.5,1]}(\alpha, \alpha)$ and $\text{Beta}_{[0,0.5]}(\alpha, \alpha)$ respectively, or further force them to be greater than a specific constant. The implementation details of the sample strategy is presented in Appendix.

The decoupling process can be easily extended to the hidden layers of neural networks. For example, in the l-th hidden layer, we can decouple the features to recover z_a^l, z_b^l with $\lambda_1^{l-1}, \lambda_2^{l-1}$ that used in the mixing process of the last previous layer. Then, we remix z_a^l, z_b^l twice with newly sampled λ_1^l, λ_2^l and feed the remixed features into the next layer. In practice, this mix-then-decouple process can be embedded in any hidden layer with only several lines of codes and negligible computational cost.

Table 4 and 5 show the overall comparative results on calibrated ECE and accuracy. MIT-L means that we simply employ the decoupling process in the last layer (before softmax), and MIT-A means that we apply it to all the blocks of the ResNets and also the last layer. MIT-L/A with $\Delta\lambda > \frac{1}{2}$ means that we force the difference between λ_1 and λ_2 to be greater than $\frac{1}{2}$ (see details in Appendix). As we can see, the methods that learn from one-hot labels (the right 5 columns) clearly outperform the others on calibrated ECE. As is mentioned in Section 4, two ablated variants Mixup-SC and Mixup-IO perform well on calibration, however, they are sub-optimal in terms of the predictive accuracy. As is shown in Table 5, they underperform ERM on SVHN and Tiny-ImageNet with all backbones, while yield very small improvements on CIFAR-10/100.

Our methods improve calibration without sacrificing the predictive performance. As is shown Table 4, all of our methods attain lower calibrated ECE than ERM, which is opposite to the results of vanilla mixup. And more surprisingly, Table 5 shows that our methods can also achieve better performance on accuracy than vanilla mixup. It is worth noting that, in terms of accuracy, vanilla mixup fails in several cases on SVHN and Tiny-ImageNet (see the results of ResNet-110/152 on Tiny-ImageNet), but our methods show stable improvements. Furthermore, we can observe that with the constraint on λ, MIT-A ($\Delta\lambda > \frac{1}{2}$) achieves higher accuracy compared with the ablated version MIT-A. The comparative results with other calibration methods are

Table 5. The overall comparative results in terms of the predictive **accuracy**. The number in each bracket indicates the ranking across all methods. The orange/blue color indicates that a method outperforms/underperforms ERM in average. The **boldface** and underline denote the best and the second best results of each row. The marker † means the backbone is pretrained.

	Backbones	ERM	Mixup (0.1)	Mixup (0.5)	Mixup (1.0)	Mixup (DT)	Mixup (TO)	Mixup (SC)	Mixup (IO)	**MIT-A**	**MIT-L** $(\Delta\lambda{>}\frac{1}{2})$	**MIT-A** $(\Delta\lambda{>}\frac{1}{2})$
SVHN	ResNet18	95.4 (5)	95.5 (4)	94.8 (7)	94.5 (8)	95.6 (3)	**96.0 (1)**	94.3 (9)	93.5 (10)	95.0 (6)	93.2 (11)	95.7 (2)
	ResNet50	96.0 (4)	96.0 (3)	95.8 (5)	95.5 (8)	95.5 (7)	95.7 (6)	95.3 (9)	94.9 (10)	96.2 (2)	94.3 (11)	**96.2 (1)**
	ResNet110	96.0 (5)	96.1 (4)	96.3 (3)	95.8 (7)	95.6 (8)	95.9 (6)	95.4 (9)	95.3 (10)	96.5 (2)	95.0 (11)	**96.7 (1)**
	ResNet152	96.2 (5)	96.6 (2)	96.4 (4)	96.2 (6)	95.6 (8)	95.9 (7)	95.5 (9)	95.5 (10)	96.5 (3)	94.9 (11)	**96.7 (1)**
	Avg. gain	—	+ 0.11	– 0.10	– 0.40	– 0.32	– 0.06	– 0.78	– 1.12	+ 0.14	– 1.55	+ 0.41
CIFAR-10	ResNet18	94.5 (9)	95.1 (6)	95.7 (3)	95.8 (2)	93.9 (11)	94.5 (8)	94.4 (10)	94.7 (7)	95.5 (4)	95.2 (5)	**95.9 (1)**
	ResNet50	94.4 (9)	95.3 (7)	95.8 (3)	**96.0 (1)**	93.1 (11)	94.2 (10)	94.5 (8)	95.3 (6)	95.8 (4)	95.7 (5)	96.0 (2)
	ResNet110	94.7 (9)	95.7 (6)	**96.3 (1)**	96.2 (2)	93.7 (11)	94.3 (10)	95.1 (8)	95.4 (7)	96.1 (4)	96.0 (5)	96.1 (3)
	ResNet152	95.1 (8)	95.8 (7)	96.4 (2)	**96.7 (1)**	93.9 (11)	94.8 (10)	95.0 (9)	95.8 (6)	96.3 (4)	96.2 (5)	96.4 (3)
	Avg. gain	—	+ 0.78	+ 1.36	+ 1.53	– 1.01	– 0.21	+ 0.08	+ 0.64	+ 1.27	+ 1.12	+ 1.41
CIFAR-100	ResNet18	74.4 (8)	75.3 (7)	76.8 (2)	**77.2 (1)**	72.4 (11)	76.4 (4)	72.6 (9)	72.5 (10)	76.2 (5)	75.9 (6)	76.6 (3)
	ResNet50	73.9 (9)	76.4 (6)	78.3 (2)	77.8 (3)	68.2 (11)	75.1 (7)	72.9 (10)	74.5 (8)	**78.3 (1)**	76.6 (5)	77.7 (4)
	ResNet110	76.1 (9)	77.9 (6)	**80.1 (1)**	79.3 (2)	70.9 (11)	77.3 (7)	74.6 (10)	76.7 (8)	78.7 (4)	77.9 (5)	79.1 (3)
	ResNet152	75.3 (9)	78.2 (6)	79.7 (2)	79.6 (3)	72.5 (11)	76.9 (7)	75.1 (10)	76.7 (8)	79.1 (4)	78.2 (5)	**79.8 (1)**
	Avg. gain	—	+ 2.01	+ 3.79	+ 3.55	– 3.92	+ 1.50	– 1.12	+ 0.18	+ 3.14	+ 2.24	+ 3.38
Tiny-ImageNet	ResNet18	46.1 (9)	46.6 (7)	47.4 (5)	47.8 (4)	36.5 (11)	47.1 (6)	43.0 (10)	46.6 (8)	**49.5 (1)**	48.5 (3)	49.3 (2)
	ResNet50	49.3 (7)	49.5 (6)	50.0 (5)	50.4 (4)	37.5 (11)	49.0 (8)	46.4 (10)	48.8 (9)	51.4 (2)	51.0 (3)	**51.8 (1)**
	ResNet110	48.5 (3)	43.6 (7)	42.7 (9)	42.6 (10)	35.6 (11)	44.6 (4)	43.9 (6)	43.5 (8)	48.6 (2)	44.4 (5)	**50.8 (1)**
	ResNet152	47.3 (2)	44.7 (5)	42.3 (9)	44.6 (6)	34.5 (11)	45.5 (4)	43.0 (8)	39.7 (10)	46.1 (3)	43.8 (7)	**50.0 (1)**
	ResNet18†	53.6 (6)	53.5 (8)	54.0 (5)	53.5 (7)	44.1 (11)	**54.7 (1)**	49.7 (10)	50.5 (9)	54.5 (3)	54.4 (4)	54.7 (2)
	ResNet152†	62.4 (6)	63.2 (2)	**63.7 (1)**	63.0 (3)	49.6 (11)	62.6 (4)	58.8 (10)	59.9 (9)	61.9 (7)	62.5 (5)	61.6 (8)
	Avg. gain	—	– 1.01	– 1.18	– 0.87	– 11.5	– 0.60	– 3.70	– 3.04	+ 0.81	– 0.41	+ 1.83

presented in Appendix, from which we can see similar phenomenon that these methods improve calibration in training but often degrade the result after post-calibration.

The results of our methods in Table 4 and 5 are conducted by simply setting $\alpha = 1$. In appendix, we show the performance of our method can be further slightly improved with other choices of α. Moreover, due to the space limitation, we present and discuss the results on two other calibration metric (average calibration error) ACE and (negative log-likelihood) NLL in Appendix.

6. Conclusion

We systemically studied the calibration peformance of mixup, with a focus on the interaction between mixup and post-calibration. We found a pathological but interesting phenomenon that although mixup-trained models yield better accuracy and raw calibration performance, they are usually not as calibratable as models trained with ERM. We explain this by decomposing mixup into data transformation and random perturbation, and show that the former implicitly performs confidence penalty and hence degrades the model's calibratability. To tackle this, we first studied the mixup inference strategy with the help of a decoupling process. We found that despite it improves on mixup, this ensemble-like strategy does not necessarily outperform simple ensemble of independently trained models. To better deal with the dilemma between accuracy and calibration, we proposed to perform mixup inference in training. It is shown that this simple strategy can properly solve mixup's calibration issue, and also improve accuracy when applying it in the hidden layers. Despite the surprising performance, we prefer to regard this work as an empirical study, which contributes nontrivial knowledge to the understanding of deep model calibration.

7. Acknowledgments

The authors wish to thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by the National Science Foundation of China (62225602). We thank the Big Data Center of Southeast University for providing the facility support on the numerical calculations in this paper. The first author was supported by Tencent Rhino-Bird Elite Training Program. The work described in this paper was also supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (Project Reference Number: 14201620 and 14201321).

References

[1] Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. In *International Conference on Learning Representations*, 2019. 2, 3, 6

[2] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019. 2

[3] Luigi Carratino, Moustapha Cissé, Rodolphe Jenatton, and Jean-Philippe Vert. On mixup regularization. *arXiv preprint arXiv:2006.06049*, 2020. 1, 2, 4

[4] Muthu Chidambaram, Xiang Wang, Yuzheng Hu, Chenwei Wu, and Rong Ge. Towards understanding the data dependency of mixup-style training. In *International Conference on Learning Representations*, 2021. 2

[5] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330, 2017. 1, 2, 3

[6] Kartik Gupta, Amir Rahimi, Thalaiyasingam Ajanthan, Thomas Mensink, Cristian Sminchisescu, and Richard Hartley. Calibration of neural networks using splines. In *International Conference on Learning Representations*, 2020. 2

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[8] 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 *International Conference on Learning Representations*, 2019. 2

[9] Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *International Conference on Machine Learning*, pages 5275–5285. PMLR, 2020. 2

[10] 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. *Advances in neural information processing systems*, 32, 2019. 2

[11] Ananya Kumar, Percy S Liang, and Tengyu Ma. Verified uncertainty calibration. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[12] Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *International Conference on Machine Learning*, pages 2805–2814, 2018. 1

[13] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017. 6

[14] Changchun Li, Ximing Li, Lei Feng, and Jihong Ouyang. Who is your right mixup partner in positive and unlabeled learning. In *International Conference on Learning Representations*, 2021. 2

[15] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*, 2019. 2

[16] Juan Maroñas, Daniel Ramos, and Roberto Paredes. On calibration of mixup training for deep neural networks. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 67–76. Springer, 2021. 1, 2, 3

[17] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating deep neural networks using focal loss. In *Advances in Neural Information Processing Systems*, pages 15288–15299, 2020. 1

[18] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems*, pages 4696–4705, 2019. 1

[19] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 625–632, 2005. 2

[20] Tianyu Pang, Kun Xu, and Jun Zhu. Mixup inference: Better exploiting mixup to defend adversarial attacks. In *International Conference on Learning Representations*, 2019. 2, 5

[21] Kanil Patel, William Beluch, Bin Yang, Michael Pfeiffer, and Dan Zhang. Multi-class uncertainty calibration via mutual information maximization-based binning. In *International Conference on Representation Learning*, 2021. 1, 2

[22] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3):61–74, 1999. 1, 2

[23] Rahul Rahaman et al. Uncertainty quantification and deep ensembles. *Advances in Neural Information Processing Systems*, 34:20063–20075, 2021. 1

[24] Amir Rahimi, Kartik Gupta, Thalaiyasingam Ajanthan, Thomas Mensink, Cristian Sminchisescu, and Richard Hartley. Post-hoc calibration of neural networks. *arXiv preprint arXiv:2006.12807*, 2020. 2

[25] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020. 2

[26] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 4

[27] 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 *Advances in Neural Information Processing Systems*, pages 13888–13899, 2019. 1, 2, 3

[28] Juozas Vaicenavicius, David Widmann, Carl Andersson, Fredrik Lindsten, Jacob Roll, and Thomas Schön. Evalu-

ating model calibration in classification. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3459–3467. PMLR, 2019. 2

[29] 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 *International Conference on Machine Learning*, pages 6438–6447. PMLR, 2019. 5

[30] Deng-Bao Wang, Lei Feng, and Min-Ling Zhang. Rethinking calibration of deep neural networks: Do not be afraid of overconfidence. *Advances in Neural Information Processing Systems*, 34:11809–11820, 2021. 2, 4

[31] Yeming Wen, Ghassen Jerfel, Rafael Muller, Michael W Dusenberry, Jasper Snoek, Balaji Lakshminarayanan, and Dustin Tran. Combining ensembles and data augmentation can harm your calibration. In *International Conference on Learning Representations*, 2020. 1, 6

[32] David Widmann, Fredrik Lindsten, and Dave Zachariah. Calibration tests in multi-class classification: A unifying framework. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[33] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 2

[34] Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *International Conference on Machine Learning*, pages 609–616, 2001. 1

[35] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699, 2002. 2

[36] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 2

[37] Linjun Zhang, Zhun Deng, Kenji Kawaguchi, Amirata Ghorbani, and James Zou. How does mixup help with robustness and generalization? In *International Conference on Learning Representations*, 2020. 2

[38] Linjun Zhang, Zhun Deng, Kenji Kawaguchi, and James Zou. When and how mixup improves calibration. In *International Conference on Machine Learning*, pages 26135–26160. PMLR, 2022. 1, 2

[39] Jianchao Zhu, Liangliang Shi, Junchi Yan, and Hongyuan Zha. Automix: Mixup networks for sample interpolation via cooperative barycenter learning. In *European Conference on Computer Vision*, pages 633–649. Springer, 2020. 2