# On Mixup Training: Improved Calibration and Predictive Uncertainty for Deep Neural Networks

**Sunil Thulasidasan** [1 2] **Gopinath Chennupati** [1] **Jeffrey Bilmes** [2] **Sarah Michalak** [1] **Tanmoy Bhattacharya** [1]

## Abstract

Mixup (Zhang et al., 2017) is a recently proposed method for training deep neural networks where additional samples are generated during training by convexly combining random pairs of images and their associated labels. While simple to implement, it has shown to be a surprisingly effective method of data augmentation for image classification; DNNs trained with mixup show noticeable gains in classification performance on a number of image classification benchmarks. In this work, we discuss a hitherto untouched aspect of mixup training – the calibration and predictive uncertainty of the resulting models. We find that DNNs trained with mixup are significantly better calibrated – i.e the predicted softmax scores are much better indicators of the actual likelihood of a correct prediction – than DNNs trained in the regular fashion. We conduct experiments on a number of datasets and architectures – including large-scale datasets like ImageNet – and find this to be the case. We also find improved calibration on NLP datasets for text classification. Additionally, we find that merely mixing features does not result in the same calibration benefit and that the label smoothing in mixup training plays a significant role in improving calibration. Finally, we also observe that mixup-trained DNNs are less prone to over-confident predictions on out-of-distribution and random-noise data. We conclude that the typical overconfidence seen in neural networks, even on in-distribution data is likely a consequence of training with hard labels, suggesting that mixup training be employed for classification tasks where predictive uncertainty is a significant concern.

[1]Los Alamos National Laboratory [2]Department of Electrical and Computer Engineering, University of Washington. Correspondence to: S. Thulasidasan <sunil@lanl.gov>.

Presented at the ICML 2019 Workshop on Uncertainty and Robustness in Deep Learning. 

## 1. Introduction: Overconfidence and Uncertainty in Deep Learning

Machine learning algorithms are replacing or expected to increasingly replace humans in decision-making pipelines. With the deployment of AI-based systems in high risk fields such as medical diagnosis (Miotto et al., 2016), autonomous vehicle control (Levinson et al., 2011) and the legal sector (Berk, 2017), the major challenges of the upcoming era are thus going to be in issues of uncertainty and trust-worthiness of a classifier. With deep neural networks (DNNs) having established supremacy in many pattern recognition tasks, it is the predictive uncertainty of these types of classifiers that will be of increasing importance. A DNN must not only be accurate, but also indicate when it is likely to get the wrong answer. In other words, the confidence of the DNN must be *well calibrated*, where the predicted softmax scores should be indicative of the actual likelihood of correctness. The issue of calibration in modern DNNs was examined in (Guo et al., 2017) where the authors show significant empirical evidence that modern deep neural networks are poorly calibrated, with depth, weight decay and batch normalization all influencing calibration. Modern architectures, it turns out, are prone to overconfidence, meaning accuracy is likely to be lower than what is indicated by the predictive score.

This phenomenon of overconfidence has been observed in a wide variety of deep architectures. Figure 1 shows a series of joint density plots of the average winning score and accuracy of a VGG-16 (Simonyan & Zisserman, 2014) network over the CIFAR-100 (Krizhevsky & Hinton, 2009) validation set, plotted at different epochs. Both the confidence (captured by the winning score) as well as accuracy start out low and gradually increase as the network learns. However, what is interesting – and concerning – is that the confidence always leads accuracy in the later stages of training. Towards the end of training, accuracy saturates while confidence continues to improve resulting in a very sharply peaked distribution of winning scores. One of the contributing factors to the above phenomenon is the fact that the training signal – the one-hot encoded labels in supervised learning, that have all the probability mass in one class – are zero-entropy distributions that admit no uncertainty about

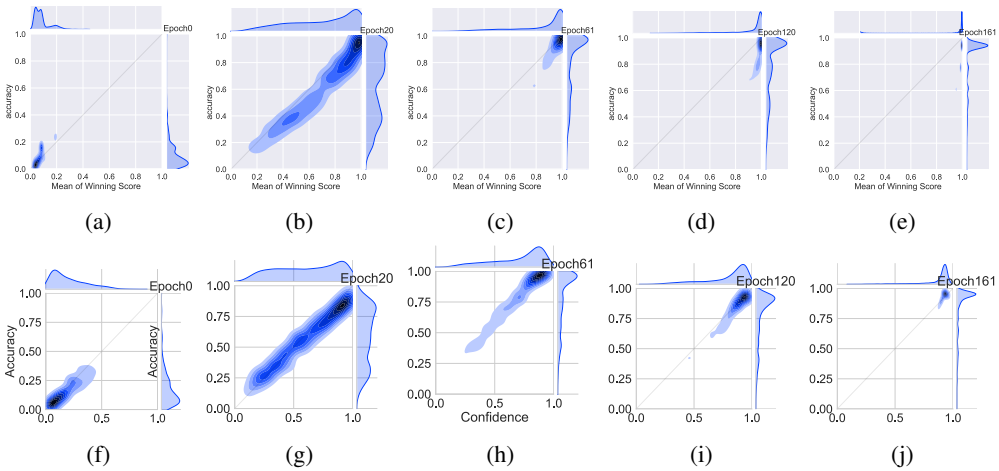



*Figure 1.* Joint density plot of accuracy vs confidence (captured by the winning softmax score) on the CIFAR-100 validation set at different training epochs for the VGG-16 deep neural network. **Top Row:** In regular training, the DNN moves from underconfidence, at the beginning of training, to overconfidence at the end. A well-calibrated classifier would have most of the density lying on the $x = y$ gray line. **Bottom Row:** Training with mixup on the same architecture and dataset. At corresponding epochs, the network is much better calibrated.

the input. The DNN is thus, in some sense, trained to become overconfident. Hence a worthwhile line of exploration is whether principled approaches to label smoothing can somehow temper overconfidence. In this work, we carry out such an exploration by investigating the effect of the recently proposed *mixup* (Zhang et al., 2017) method of training deep neural networks. In mixup, additional synthetic samples are generated during training by convexly combining random pairs of images and, imporantly, their labels as well. While simple to implement, it has shown to be a surprisingly effective method of data augmentation: DNNs trained with mixup show noticeable gains in classification performance on a number of image classification benchmarks. However neither the original work or subsequent extensions to mixup (Verma et al., 2018; Guo et al., 2018; Liang et al., 2018) have explored the effect of mixup on predictive uncertainty and DNN calibration; this is precisely what we aim to do in this paper.

Our findings are as follows: mixup trained DNNs are significantly more well calibrated – i.e., the predicted softmax scores are much better indicators of the actual likelihood of a correct prediction – than DNNs trained without mixup (see Figure 1 bottom row for an example). We also observe that merely mixing features does not result in the same calibration benefit and that the label smoothing in mixup training plays a significant role in improving calibration. Further, we also observe that mixup-trained DNNs are less prone to over-confident predictions on out-of-distribution and random-noise data. We discuss details in the following sections.

## 2. An Overview of Mixup Training

Mixup training (Zhang et al., 2017) is based on the principle of Vicinal Risk Minimization (Chapelle et al., 2001)(VRM): the classifier is trained not only on the training data, but also in the *vicinity* of each training sample. The vicinal points are generated according to the following simple rule introduced in (Zhang et al., 2017):

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j$$
$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j$$

where $x_i$ and $x_j$ are two randomly sampled input points, and $y_i$ and $y_j$ are their associated one-hot encoded labels (more details on mixup training are given in the appendix in Section A). Training this way not only augments the feature set $\tilde{X}$, but the induced set of soft-labels also encourages the strength of the classification regions to vary linearly between samples. The experiments in (Zhang et al., 2017) and related work in (Inoue, 2018; Verma et al., 2018; Guo et al., 2018) show noticeable performance gains in various image classification tasks. The linear interpolator $\lambda \in [0, 1]$ that determines the mixing ratio is drawn from a symmetric Beta distribution, $Beta(\alpha, \alpha)$;see section A of the appendix for more details. In this work, we also look at the effect of $\alpha$ on calibration performance.

## 3. Experiments

We perform numerous experiments to analyze the effect of mixup training on the calibration of the resulting trained classifiers. We use the following datasets in our experiments: STL-10 (Coates et al., 2011), CIFAR-10 and

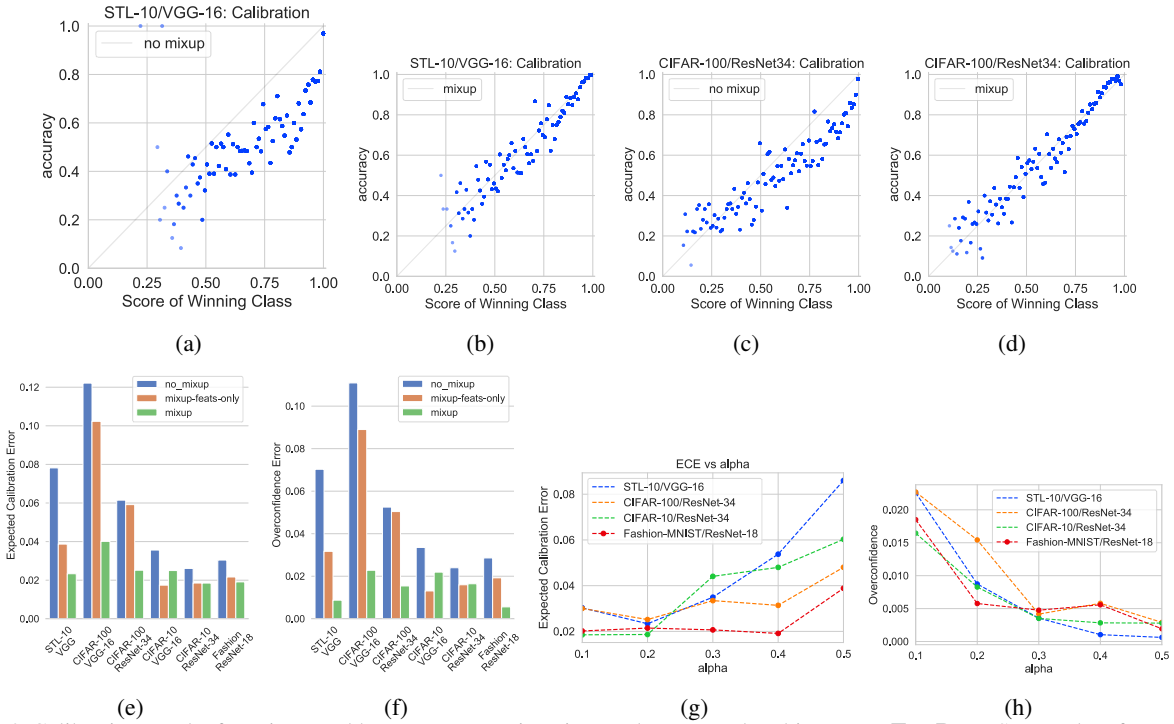



*Figure 2.* Calibration results for mixup and base-case on various image datasets and architectures. **Top Row**: Scatterplots for accuracy and confidence for STL-10(a,b) and CIFAR-100(c,d). The mixup case is much better calibrated with the points lying closer to the $x = y$ line, while in the base case, points tend to lie in the overconfident region. *Bottom Row*: Expected Calibration Error (e) and Overconfidence error (f) on various architectures. Experiments suggest best ECE is achieved in the [0.2,0.4] range for $\alpha$(g), while overconfidence error decreases monotonically with $\alpha$ due to underfitting (h).

CIFAR-100 (Krizhevsky & Hinton, 2009) and Fashion-MNIST (Xiao et al., 2017). Additional details regarding experimental setup are given in the appendix. We measure the calibration of the network using the **Expected Calibration Error** (as described in (Guo et al., 2017)), and also introduced an additional **Overconfidence Penalty** metric; see appendix for details. In this work we only apply mixup to *pairs* of images as done in (Zhang et al., 2017); all experiments were done using the mixup authors' code available at (Zhang).

### 3.1. Results

Results on the various datasets and architectures are shown in Figure 2. While the performance gain in validation accuracy (not shown) was generally consistent with the results reported in (Zhang et al., 2017), it is the effects on network calibration that we focus here. The top row shows a calibration scatter plot for STL-10 and CIFAR-100, highlighting the effect of mixup training. In a well calibrated model, where the confidence matches the accuracy most of the points will be on $x = y$ line. We see that in the base case, both for STL-10 and CIFAR-100, most of the points tend to lie in the overconfident region. The mixup case is much better calibrated, noticeably in the high-confidence regions.

In the bottom row, we compare the effect on ECE for both mixup and the case where we only mix features to tease out the effect of label mixing. As we see, merely mixing features does not generally provide the calibration benefits seen in the full-mixup case suggesting that the point-mass distributions in hard-coded labels are contributing factors to overconfidence. We also show the effect on ECE as we vary the hyperparameter $\alpha$ of the mixing parameter distribution. For very low values of $\alpha$, the behavior is similar to the base case (as expected), but ECE also noticeably worsens for higher values of $\alpha$ due to the model being *under-confident*. Overconfidence alone decreases monotonically as we increase $\alpha$ as shown in Figure 2h

#### 3.1.1. Large-scale Experiments on ImageNet

Here we report the results of calibration metrics resulting form mixup training on the 1000-class version of the ImageNet (Deng et al., 2009) data comprising of over 1.2 million images. One of the advantages of mixup and its implementation is that it adds very little overhead to the training time, and thus can be easily applied to large scale datasets like ImageNet. We perform distributed parallel training on the ResNext-101 (32x4d) (Xie et al., 2017) architecture using the synchronous version of stochastic gradient descent and train till 93% accuracy is reached over the top-5

predictions. The results are shown in Figure 3. The calibration scatter-plot suggests that mixup training provides noticeable benefits even in the large-data scenario, where the models should be less prone to over-fitting. We also observed that the mixup model also achieved a consistently higher classification performance of $\approx 0.4$ percent over the other methods.

Additional results on NLP datasets are given in the appendix.

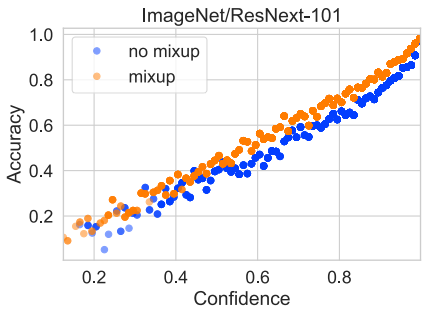



*Figure 3.* Calibration on ImageNet for ResNext 101

#### 3.1.2. Testing on Out-of-Distribution and Random Data

In this section, we explore the effect of mixup training when predicting on samples from unseen classes (out-of-distribution) and random noise images. We first train a VGG-16 network on in-distribution data (STL-10) and then predict on classes not seen in training sampled from the ImageNet dataset. For the random noise images, we test on gaussian random noise with the same mean and variance as the training set. We compare the performance of a mixup-trained model with that of the baseline, as well as a temperature calibrated per-trained baseline as described in (Guo et al., 2017). We also compare the prediction uncertainty using the Montecarlo dropout method described in (Gal & Ghahramani, 2016) where multiple forward passes using dropout are made during test-time. We average predictions over 10 runs.

The distribution over prediction scores for out-of-distribution and random data for mixup and comparison methods are shown in Figure 4. The differences versus the baseline are striking; in both cases, the mixup DNN is noticeably less confident than its non-mixup counterpart, with the score distribution being nearly perfectly separable in the random noise case. Temperature scaling is a post-training calibration method and we expect it to be well calibrated on real images, and indeed we see that temperature scaling is more conservative than mixup on real but out-of-sample data. However, it is noticeably more overconfident than mixup in the random-noise case. Further, mixup

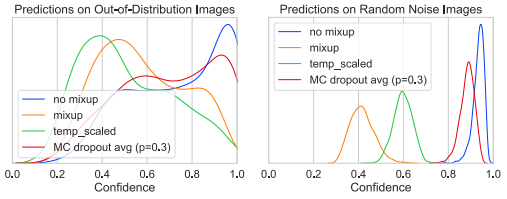



*Figure 4.* Behavior of mixup training vs base-case on out-of-distribution (left) and random noise images(right). Model trained on STL-10 images and tested on out-of-category classes from Imagenet and gaussian random noise.

also performs significantly better than MC-dropout in both cases. The results here suggest that the effect of training with interpolated samples and the resulting label smoothing tempers over-confidence in regions away from the training data. While these experiments were limited to two datasets and one architecture, the results indicate that training by minimizing vicinal risk can be an effective way to enhance reliability of predictions in DNNs.

## 4. Conclusions

We presented results on an unexplored area of mixup based training – its effect on DNN calibration and predictive uncertainty. Existing empirical work has conclusively shown the benefits of mixup for boosting classification performance; in this work, we show an additional important benefit – mixup trained networks turn out to be better calibrated and provide more reliable estimates both for in-sample and out-of-sample data (being under-confident in the latter case). There are possibly multiple reasons for this: the data augmentation provided by mixup is a form of regularization that prevents over-fitting and memorization, tempering over-confidence in the process. The label smoothing resulting from mixup might be viewed as a form of entropic regularization on the training signals, again preventing the DNN from driving the training error to zero. Recent work (Verma et al., 2018) has shown how the classification regions in mixup are smoother, without sudden jumps from one high confidence region to the other suggesting that the lack of sharp boundary transitions in classification regions play an important role in producing well-calibrated classifiers. Indeed, the classification performance boost coupled with the well-calibrated nature of mixup trained DNNs as studied in this paper suggest that mixup based training be employed in situations where predictive uncertainty is a significant concern.

## Acknowledgements

This work has been supported in part by the Joint Design of Advanced Computing Solutions for Cancer (JDACS4C) pro-

gram established by the U.S. Department of Energy (DOE) and the National Cancer Institute (NCI) of the National Institutes of Health. This work was performed under the auspices of the U.S. Department of Energy by Los Alamos National Laboratory under Contract DE-AC5206NA25396.

## References

Berk, R. An impact assessment of machine learning risk forecasts on parole board decisions and recidivism. *Journal of Experimental Criminology*, 13(2):193–216, 2017.

Chapelle, O., Weston, J., Bottou, L., and Vapnik, V. Vicinal risk minimization. In *Advances in neural information processing systems*, pp. 416–422, 2001.

Coates, A., Ng, A., and Lee, H. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223, 2011.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255. Ieee, 2009.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059, 2016.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. *arXiv preprint arXiv:1706.04599*, 2017.

Guo, H., Mao, Y., and Zhang, R. Mixup as locally linear out-of-manifold regularization. *arXiv preprint arXiv:1809.02499*, 2018.

Inoue, H. Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929*, 2018.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Levinson, J., Askeland, J., Becker, J., Dolson, J., Held, D., Kammel, S., Kolter, J. Z., Langer, D., Pink, O., Pratt, V., et al. Towards fully autonomous driving: Systems and algorithms. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pp. 163–168. IEEE, 2011.

Liang, D., Yang, F., Zhang, T., and Yang, P. Understanding mixup training methods. *IEEE Access*, 6:58774–58783, 2018.

Miotto, R., Li, L., Kidd, B. A., and Dudley, J. T. Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. *Scientific reports*, 6:26094, 2016.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Verma, V., Lamb, A., Beckham, C., Courville, A., Mitliagkis, I., and Bengio, Y. Manifold mixup: Encouraging meaningful on-manifold interpolation as a regularizer. *arXiv preprint arXiv:1806.05236*, 2018.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

Zhang, H. https://github.com/hongyi-zhang/mixup.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

# Appendix A Details of Mixup Training

Mixup training [16] is based on the principle of Vicinal Risk Minimization [3](VRM): the classifier is trained not only on the training data, but also in the *vicinity* of each training sample. The vicinal points are generated according to the following simple rule introduced in [16]:

$$\tilde{x} = \lambda x_i + (1-\lambda)x_j$$
$$\tilde{y} = \lambda y_i + (1-\lambda)y_j$$

where $x_i$ and $x_j$ are two randomly sampled input points, and $y_i$ and $y_j$ are their associated one-hot encoded labels. This has the effect of the empirical Dirac delta distribution

$$P_\delta(x,y) = \frac{1}{n}\sum_i^n \delta(x = x_i, y = y_i)$$

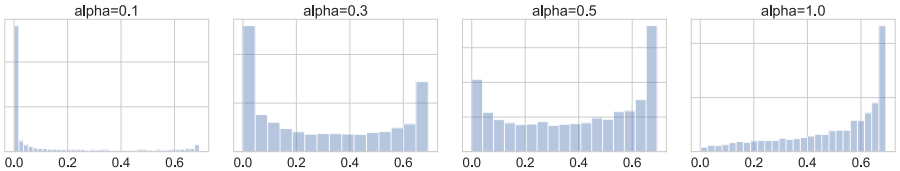
centered at $(x_i, y_i)$ being replaced with the *empirical vicinal distribution*

$$P_\nu(\tilde{x},\tilde{y}) = \frac{1}{n}\sum_i^n \nu(\tilde{x},\tilde{y}|x_i,y_i)$$

The vicinal samples $(\tilde{x},\tilde{y})$ are generated as above, and during training minimization is performed on the *empirical vicinal risk*:

$$R_\nu(f) = \frac{1}{m}\sum_{i=1}^m \mathcal{L}(f(\tilde{x_i}),\tilde{y_i})$$

where $L$ is the standard cross-entropy loss, but calculated on the soft-labels $\tilde{y_i}$ instead of hard labels. Training this way not only augments the feature set $\tilde{X}$, but the induced set of soft-labels also encourages the strength of the classification regions to vary linearly betweens samples. The experiments in [16] and related work in [7, 14, 4] show noticeable performance gains in various image classification tasks.




(a) (b) (c) (d)

Figure 1: Entropy distribution of training labels as a function of the $\alpha$ parameter of the $Beta(\alpha,\alpha)$ distribution from which the mixing parameter is sampled.

The linear interpolator $\lambda \in [0,1]$ that determines the mixing ratio is drawn from a symmetric Beta distribution, $Beta(\alpha,\alpha)$, where $\alpha$ is the hyper-parameter

that controls the strength of the interpolation between pairs of images and the associated smoothing of the training labels. $\alpha = 0$ recovers the base case corresponding to zero-entropy training labels (one-hot encodings, in which case the resulting image is either just $x_i$ or $x_j$), while a high value of $\alpha$ ends up in always averaging the inputs and labels. The authors in [16] remark that values of $\alpha \in [0.1, 0.4]$ gave the best performing results for classification, while high values of $\alpha$ resulted in significant under-fitting. As we saw in our experiments, the choice of $\alpha$ also has an important effect on resulting calibration of the trained DNNs.

# Appendix B Experimental Setup

For STL-10, we use the VGG-16 [13] network. CIFAR-10 and CIFAR-100 experiments were carried out on VGG-16 as well as ResNet-34 models. For Fashion-MNIST, we used a ResNet-18 model. For all experiments, we use batch normalization, weight decay of $5 \times 10^{-4}$, trained the network using SGD with Nesterov momentum, training for 200 epochs with an initial learning rate of 0.1 halved at 2 at 60,120 and 160 epochs. Unless otherwise noted, calibration results are reported for the best performing epoch on the validation set.

For the ImageNet experiments, we perform distributed parallel training using the synchronous version of stochastic gradient descent. We use the ImageNet training code from [1], which uses a cyclical learning rate and progressive resizing of images for faster training times. We train on a 32-GPU cluster using the ResNext-101 (32x4d) [15] architecture and train till 93% accuracy is reached over the top-5 predictions.

# Appendix C Calibration Metrics

Softmax predictions are grouped into $M$ interval bins of equal size. Let $B_m$ be the set of samples whose prediction scores (the winning softmax score) fall into bin $B_m$. The accuracy and confidence of $B_m$ are defined as

$$\text{acc}(B_m) = \frac{1}{|B_m|} \sum_{iinB_m} \mathbf{1}(\hat{y}_i = y_i)$$

$$\text{conf}(B_m) = \frac{1}{|B_m|} \sum_{iinB_m} \hat{p}_i$$

where $\hat{p}_i$ is the confidence (winning score) of sample $i$. The **Expected Calibration Error** (ECE) is then defined as:

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \Big| \text{acc}(B_m) - \text{conf}(B_m) \Big|$$

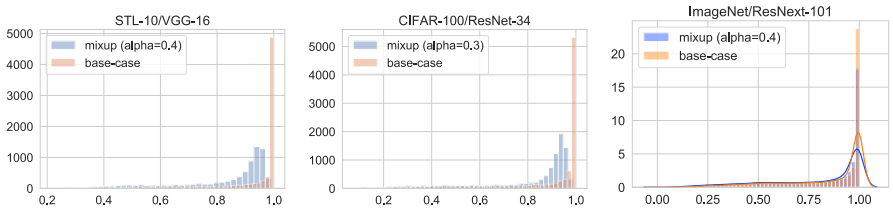
In high-risk applications, confident but wrong predictions can be especially harmful; thus we also define an additional calibration metric – the **Overconfidence**

**Error** (OE)– as follows

$$\mathrm{OE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \Big[ \mathrm{conf}(B_m) \times \max\Big( \mathrm{conf}(B_m) - \mathrm{acc}(B_m), 0 \Big) \Big]$$

This penalizes predictions by the weight of the confidence but only when confidence exceeds accuracy; thus overconfident bins incur a high penalty.

# Appendix D Prediction Confidence of Mixup




(a) (b) (c)

Figure 2: Distribution of winning scores

As we have seen, mixup trained models are less overconfident than their non-mixup counterparts. Here we show the distribution of the winning scores for various image datasets. As shown, mixup models are less peaked in the very-high confidence region.

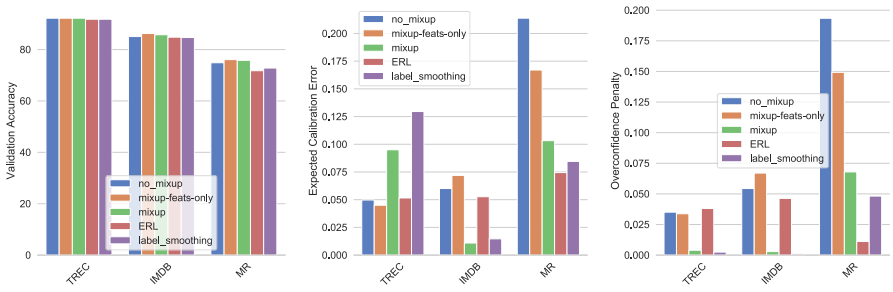
# Appendix E Experiments on NLP Data

While mixup was originally suggested as a method to mostly improve performance on vision classification tasks, here we explore the effect of mixup training in the NLP domain. Very recent work[5] has shown classification performance benefits in the NLP domain though this has been a relatively under-explored area for mixup training. Note that a straight-forward mixing of inputs (as in pixel-mixing in images) will generally produce nonsense input since the semantics are unclear. To avoid this, we modify the mixup strategy to perform mixup on the embeddings layer rather than directly on the input documents. For our experiments, we employ mixup on NLP data for text classification using the following three datasets:

1. MR [11]: Movie reviews with two classes, documents are split into train/test sets of 9596/1066.

2. TREC [9]: Question dataset, where the classification involves identifying six classes. The dataset is divided into 5452/500 documents for train/test splits.

3. IMDB [10]: Binary classification with movie reviews split into train/test sets of- 25000/25000




(a) (b) (c)

Figure 3

We train CNN for sentence classification (Sentence-level CNN) [8], where we initialize all the words with pre-trained GloVe [12] embeddings, which are modified while training on each dataset. For the remaining parameters, we use the values suggested in [8]. We refrain from training the most recent NLP models [6, 2, 17], since our aim here is not to show state-of-art classification performance on these datasets, but to study the effect on calibration. Also, the design of the more recent NLP models makes embedding mixup less straightforward. Nevertheless, the performance benefits on calibration, shown in Figure 3 are evident where mixup provides noticeable gains for all datasets, both in terms of calibration and overconfidence.

# References


[1] Yaroslav Bulatov. https://github.com/diux-dev/imagenet18.

[2] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

[3] Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. Vicinal risk minimization. In *Advances in neural information processing systems*, pages 416–422, 2001.

[4] Hongyu Guo, Yongyi Mao, and Richong Zhang. Mixup as locally linear out-of-manifold regularization. *arXiv preprint arXiv:1809.02499*, 2018.

[5] Hongyu Guo, Yongyi Mao, and Richong Zhang. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint arXiv:1905.08941*, 2019.

[6] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[7] Hiroshi Inoue. Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929*, 2018.

[8] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[9] Xin Li and Dan Roth. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics, 2002.

[10] Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics, 2011.

[11] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 115–124. Association for Computational Linguistics, 2005.

[12] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[14] Vikas Verma, Alex Lamb, Christopher Beckham, Aaron Courville, Ioannis Mitliagkis, and Yoshua Bengio. Manifold mixup: Encouraging meaningful on-manifold interpolation as a regularizer. *arXiv preprint arXiv:1806.05236*, 2018.

[15] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[16] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[17] Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. *arXiv preprint arXiv:1611.06639*, 2016.