LOGIT AS AUXILIARY WEAK-SUPERVISION FOR RELIABLE AND ACCURATE PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

When a person identifies objects, he or she can think by associating objects to many classes and conclude by taking inter-class relations into account. This cognitive system can make a more reliable prediction. Inspired by these observations, we propose a new network training strategy to consider inter-class relations, namely LogitMix. Specifically, we use recent data augmentation techniques (*e.g.*, Mixup, Manifold Mixup, or CutMix) as baselines for generating mixed samples. Then, LogitMix suggests using the mixed logit (*i.e.* the mixture of two logits) as an auxiliary training objective. Because using logit before softmax activation preserves rich class relationships, it can serve as a weak-supervision signal concerning inter-class relations. Our experimental results demonstrate that LogitMix achieves state-of-the-art performance among recent data augmentation techniques in terms of both calibration error and prediction accuracy. The source code is attached as the supplementary material.

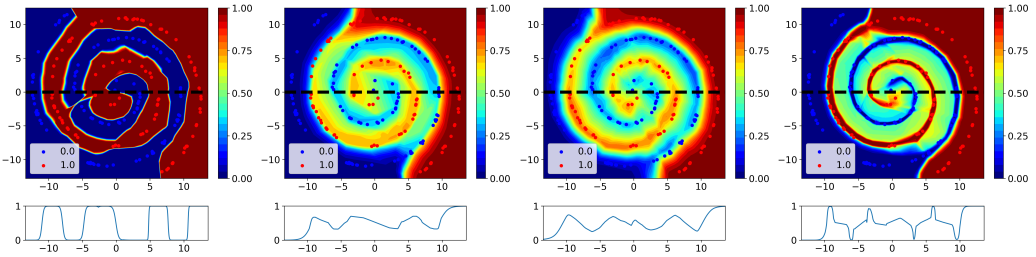
1 INTRODUCTION

Humans improve the object recognition by associating with its surrounding information– instead of focusing only on the target object (Farhadi et al., 2009; Gkioxari et al., 2018; Gould et al., 2009; Salakhutdinov et al., 2011; Torralba et al., 2004). The association with surroundings can be useful when handling more challenging and unusual decision tasks. For example, suppose that we observe an obscure, black object next to the *keyboard* on a *desk*. Considering the co-occurrence with the *desk* and the *keyboard*, we can correctly estimate that the object is probable to be a *mouse*.

Unlike humans, conventional deep neural networks (DNN) are poor at understanding the association with surrounding information. Specifically, DNN learn to de-correlate the relationship between the target class and all other classes. Because the one-hot labels induce zero probability for non-target class labels, it implies no inter-class relationship explicitly (Hou et al., 2017; Li & Maki, 2018). When the model overfits to estimate the target class label, its prediction tends to be overconfident (Guo et al., 2017). The over-confidence causes huge costs for a high-risk system such as medical diagnosis systems and autonomous driving systems. Therefore, prediction confidence can be a critical factor in determining whether or not the model is used in the target application.

Inspired by human cognitive mechanism, we propose a new training strategy considering the associations (*i.e.* class relationships) for holistic recognition. Our goal is to develop a well-calibrated and highly accurate model by considering the class relationships. To this end, we devise 1) the representation to preserve the associations, and 2) the training strategy of accounting the associations in the prediction model. Specifically, we exploit *logits* for designing the new learning signal, where it is combined with the mixing-based data preprocessing for developing our training strategy. The logits were utilized in existing knowledge distillation techniques as the estimates for inter-class correlations; a noisy estimate of associations (Hinton et al., 2015). Notably, we emphasize a clear distinction between the logit and the label distribution (*i.e.* a signal that passed softmax). That is, the label distribution only represents a positive relationship and loses much precision for hidden links. Instead, the logit contains richer information because of revealing both positive and negative (hidden) relationships among classes and encoding such relationships with effectively higher dynamic range.

As the new training tactic for considering the associations, the proposed method utilizes the data augmentation trick in (Tokozume et al., 2018; Verma et al., 2019; Yun et al., 2019; Zhang et al., 2018) for explicitly considering the class relationships in training. Specifically, we randomly select two



(a) Vanilla (b) Mixup (c) Manifold Mixup (d) Ours

Figure 1: Visualizing the probability distribution on networks trained on the 2D spiral dataset. Compared with other representative methods mixing samples, our method shows well-calibrated prediction and smooth transition between two classes. Please refer section 4.1 for details.

training examples, generate the mixed data by the linear combination of the two examples and the mixed logit using two logits (*i.e.* the logit of each example) separately by the same linear combination. Then, we train the model to map from the mixed data to the mixed logit. At the same time, the model should predict the label from original training data. We name it as LogitMix, which utilizes the mixed logit as *weak supervision for associations*. LogitMix can effectively mitigate over-fitting to the target class label (one-hot label) since it should match the mixed logit as an explicit constraint in training. The key idea of LogitMix is to utilize *the mixed logits* as auxiliary supervision, where this idea alone is effective in improving the model. LogitMix can create more synergy when combined with other mixing-based methods using *the mixed labels*.

Summarizing, LogitMix is a simple yet effective training strategy for reflecting the association (*i.e.* inter-class relationships) into decision process of DNN, mimicking human recognition system. By virtue of our simple and powerful training strategy, the proposed method improves generalization performance by preventing both over-fitting and under-fitting. We emphasize that existing techniques focus on resolving over-fitting for training a complex model, thus introducing the penalty for predicting the label. However, such techniques are not always useful for a compact model (*e.g.* MobileNetV2) because it can lead to under-fitting. Meanwhile, LogitMix utilizes mixed logit as side information in training instead. This explains why our method is effective both over-fitting and under-fitting.

As a result, LogitMix is effective for training a well-calibrated and highly accurate model. Based on extensive experiments on CIFAR100 (100 classes), Tiny-imagenet (200 classes), ILSVRC2015 (1000 classes) datasets, LogitMix outperforms two different kinds of the state-of-the-art regularization techniques using data mixing augmentation (*i.e.* Mixup (Zhang et al., 2018) for whole image-based augmentation, and CutMix (Yun et al., 2019) for patch-based augmentation), both in terms of prediction accuracy and calibration error.

2 Related work

Mixing-based augmentation. Mixup (Tokozume et al., 2018; Zhang et al., 2018) trains networks using the convex combination of two pairs of examples and labels to make the function of networks to be linear among training examples. Notably, this simple learning procedure results in robustness toward adversarial examples (Zhang et al., 2018) and improving calibration (Thulasidasan et al., 2019). AdaMixUp (Guo et al., 2019) diagnoses *manifold intrusion* in Mixup, where the mixed example collides with another example in data manifold which may induce under-fitting. This risk is regularized with a loss term penalizing the intrusion by an intrusion discriminator. Manifold Mixup (Verma et al., 2019) uses two intermediate representations at layer k as examples. When $k = 0$ implying the input layer, it reduces to vanilla Mixup (Zhang et al., 2018). While, for better performance in spatial examples, CutMix (Summers & Dinneen, 2019; Takahashi et al., 2019; Yun et al., 2019) exploits region-based augmentation strategy using a binary mask for selecting a mixing region. Notice that our method is correlated with these methods but has significant difference as we do not use mixture of supervision of true labels.

Logit regularization. Szegedy et al. (2016) propose label smoothing that modifies an one-hot label to a mixture of the one-hot label and uniform distribution with dividing the certain weight factor. Pereyra et al. (2017a) point out the limitation of label smoothing that allocates the same probability across all classes regardless of their relation. Then, they propose a confidence penalty regularizer by penalizing low entropy predictions. Xie et al. (2016) include the softening label by randomly replacing a part of labels like adding noise to the label. A pioneering work (Hinton et al., 2014; Ba & Caruana, 2014) on knowledge distillation (KD) gives us an insight on that inference outputs have useful information on the model and data distribution. From the observation that logits preserve more relations between classes, the proposed method also utilizes the logit, not the prediction distribution.

Confidence calibration. Naeini et al. (2015) diagnose the problem of the confidence calibration of modern networks and proposed the expected calibration error (ECE) as a measurement of calibration. Guo et al. (2017) address that the modern network are poorly calibrated and suggest a simple post processing calibration method which softening the prediction with temperature scaling. Various Bayesian approaches (MacKay, 1992; Neal, 1995) are commonly used for estimating the uncertainty of prediction. Nevertheless, these methods are computationally expensive as they require some modifications of training procedure. Some approaches approximate Bayesian method by using the ensemble of networks (Lakshminarayanan et al., 2017) or stochastic methods using dropout (Gal & Ghahramani, 2016). Seo et al. (2019) proposed a method for well-calibrated prediction without multiple stochastic inferences. This method is related to a label smoothing (Szegedy et al., 2016; Müller et al., 2019) and a confidence penalty (Pereyra et al., 2017b) as it makes the networks output smooth prediction. Recently, Thulasidasan et al. (2019) have empirically shown the network trained with Mixup gives better-calibrated results.

3 Proposed model

3.1 Motivation

DNN are overly confident in predicting the target class (Hou et al., 2017; Li & Maki, 2018) because they over-fit the target class to correctly predict one-hot encoded labels. In practice, over-fitting to the one-hot label is largely accepted as it provides highly accurate predictions. This accuracy gain is the result of sacrificing the accuracy in the uncertainty estimation.

Unlike computational models, humans can achieve high accuracy in both prediction and uncertainty estimation. We conjecture that the human can recognize the objects better because of considering contextual information, such as the relationship between object and object (or object and background). Decisions considering relationships can improve not only the prediction accuracy but also the uncertainty estimation as more evidence (*i.e.* relationships) is involved in the process. Learning from the human, we focus on utilizing inter-class correlations for achieving two goals: 1) to achieve high prediction accuracy and 2) to reduce calibration errors (*i.e.* gap between the accuracy and confidence) at the same time. It is because establishing both the task objective and its reliability are equally important.

The idea of utilizing inter-class correlation has proven to be successful in two independent research groups. Ba & Caruana (2014) and Hinton et al. (2015) also showed that the inter-class correlations are useful information for improving the prediction accuracy. Interestingly, various studies (Guo et al., 2017; Li & Maki, 2018) commonly observe that increasing the inter-class correlations helps to reduce the calibration errors as well. Hinton et al. (2015) stated that, although logits are noisy, they are generally useful supervisions for model training because they provide rich information about inter-class relationships. Based on these observations, we argue that inter-class correlations are useful for improving both prediction and calibration accuracy. That is, instead of simply discriminating the target class samples from all others, the model can perform more accurately and reliably if they learn and understand the positive or negative relationships among classes.

To utilize or learn the relation between classes, recent studies suggest two approaches. The former concatenates multiple samples such as mini-batch and extracts or associates the relational information among samples (Lin et al., 2018). The later combines two or more samples in a pre-determined manner (Zhang et al., 2018; Tokozume et al., 2018; Verma et al., 2019; Yun et al., 2019) to simultaneously learn each sample as well as its relationships, which we refer as the mixing-based augmentation methods. Among them, we select the framework of the mixing-based augmentation approach, because

the concatenation based approach requires the modification of model architecture, including the size of input layer.

In the following section, we first introduce mixing based techniques and explains why it leads to the improvements in both accuracy and calibration. Then, we introduce LogitMix as an effective training strategy to utilize the relation information. Our LogitMix can be combined with the existing mixed based data techniques and further maximize the performance gain, both accuracy and calibration.

3.2 MIXING-BASED AUGMENTATION TECHNIQUES

Recently, the mixing based augmentation techniques (Tokozume et al., 2018; Verma et al., 2019; Yun et al., 2019; Zhang et al., 2018) achieve highly accurate prediction performance. They generate the input data by a linear combination of two, randomly selected, training data. Likewise, their corresponding labels are also generated by the same linear combination of the two labels. By doing so, they effectively improve prediction accuracy and prevent undesirable behaviors such as memorization and sensitivity to adversarial examples at the same time. Furthermore, Thulasidasan et al. (2019) reports that Mixup training encourages that the output of DNN, the estimated label distributions, serves as a better indicator of the actual likelihood of a correction prediction. Specifically, for generating an augmented sample, the algorithm of Mixup training is as follows:

$$x_{mix} = \begin{cases} \lambda x_1 + (1-\lambda)x_2, \\ \mathbf{M} \odot x_1 + (1-\mathbf{M}) \odot x_2, \end{cases} \quad , \; y_{mix} = \lambda y_1 + (1-\lambda)y_2, \quad \lambda \sim \mathrm{Beta}(\alpha, \alpha), \tag{1}$$

where x and y denote a training sample and its label. $\mathbf{M} \in \{0,1\}^{W \times H}$ denotes a binary mask indicating where to drop out and fill in from two images, and $\odot$ is an element-wise multiplication. $\mathrm{Beta}(\cdot,\cdot)$ implies a Beta distribution, and $\alpha \in (0, \infty)$ is the parameter to control the shape of the Beta distribution. Using the mixed input and the mixed label, the model minimizes the following equation.

$$\mathcal{L}_{mix} = \lambda \mathcal{H}(\tilde{y}_{mix}, y_1) \; + (1-\lambda)\mathcal{H}(\tilde{y}_{mix}, y_2), \tag{2}$$

where $\tilde{y}$ $(= \sigma(f(x)))$ indicates the predicted label distribution from the model, f is the model, σ is an activation function which is usually the softmax function, $\sigma_{sm}(z) = \exp(z)/\sum_{i=1}^{N} \exp(z_i)$, and $\mathcal{H}$ is the cross entropy function formulated by $\mathcal{H}(p,q) = -\int_x p(x) \log q(x)$.

Whereas the original label y is encoded as an one-hot vector, the mixed label y_{mix} allows multiple factional values in the label distribution, thereby empirically yields the label smoothing effect. Lately, Thulasidasan et al. (2019) empirically show that the label smoothing effect is a key factor for achieving the accurate predictive uncertainty. Regarding mixing-based techniques as a strong data augmentation scheme, Thulasidasan et al. (2019) show that the data augmentation alone without mixed labels can substantially improve the prediction accuracy, but not the predictive uncertainty.

3.3 LOGITMIX

Utilizing in-between class relationships, LogitMix aims to achieve high prediction accuracy with the reliable prediction confidence. Specifically, we devise the mixed data augmentation trick like (Tokozume et al., 2018; Verma et al., 2019; Yun et al., 2019; Zhang et al., 2018) for explicitly considering the inter-class relationships. However, unlike existing techniques, LogitMix enforces not only label matching, but also mixed logit matching. Specifically, we suggest two loss terms: 1) the estimated label distribution should match the one-hot label, and 2) the estimated mixed logit should match the target mixed logit. For that, we assign the weak supervision for the logit of mixed data as the mixture of two logits. We call it as the weak supervision because the mixed logit is not an oracle supervision. Then, the objective of LogitMix is formulated as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \mathcal{L}_{sim}, \qquad \mathcal{L}_{sim} = \|(\lambda f(x_1) + (1-\lambda) f(x_2)) - f(x_{mix})\|_2. \tag{3}$$

As shown in Eq. 3, we linearly combine two logits for generating the target mixed logit. It is motivated by previous studies (Mikolov et al., 2013; Radford et al., 2015) that the linear interpolations are an effective way of combining factors. Also, we often observe that the linear interpolations between the hidden representations result in the meaningful regions of embedding (Radford et al., 2015; Verma et al., 2019). Thus, we utilize a simple linear mixture of two logits as weak supervision for logit training.

Why LogitMix? The first term $\mathcal{L}_{cls}$ is the classification loss with true labels and the second term $\mathcal{L}_{sim}$ is the Euclidean distance between the estimated mixed logit and the target mixed logit, namely the similarity loss. Here, $\mathcal{L}_{sim}$ assigns the constraint for the logit, a signal before softmax activation and is rewritten as follows:

$$\begin{aligned}\mathcal{L}_{sim} &= \|(\lambda \sum_{i,j}\sum_{k} \mathbf{w}_k \, l_k^{x_1}(i,j) + (1-\lambda)\sum_{i,j}\sum_{k} \mathbf{w}_k \, l_k^{x_2}(i,j)) - \sum_{i,j}\sum_{k} \mathbf{w}_k \, l_k^{x_{mix}}(i,j)\|_2 \\ &= \|(\lambda S(x_1) + (1-\lambda)S(x_2)) - S(x_{mix})\|_2\end{aligned} \tag{4}$$

where $l_k^x(i,j)$ represent the activation of unit k in the last convolutional layer at spatial location (i,j) and $\mathbf{w}_k$ is a vector of $\{w_k^1, ..., w_k^c ..., w_k^C\}$ when w_k^c is the weight of the fully connected (fc) layer that maps the k-th features to c-th class. From Eq. 4, the summation over the spatial location (i,j) is equivalent to global average pooling (GAP). If the predicted label distribution before softmax activation (*i.e.* logit) is the one-hot vector, Eq 4 becomes a matching constraint on class activation mapping (CAM) (Zhou et al., 2016); the mixture of two CAMs should equal to CAM of mixed data. In practice, logit is far from the one-hot representation and exhibits inter-class relations including negative relationships. Therefore, $\mathcal{L}_{sim}$ can be interpreted as a matching constraint on CAM aggregated over all classes; the mixture of two aggregated CAMs should equal to the aggregated CAM of mixed data.

4 Proof-of-concept study

4.1 Toy example

Here, we conduct a two-dimensional classification task with on the two-class spiral dataset (Verma et al., 2019) to reveal an impact of LogitMix comparing with the other mixing augmentation methods. To show the decision boundary and hidden representation of each method, we visualize the different predictions in different colors; two-class samples are noted either red or blue color-coded points. Since this visualization cannot conduct for region-based methods (a point does not have spatial information), we compare our approach with vanilla, Mixup (Zhang et al., 2018), and Manifold Mixup (Verma et al., 2019), as shown in Fig. 1.

First, vanilla training results in irregular boundaries, and the predictions for the inputs off-the-data manifold are over-confident; the predictions for the out-of-distribution samples exhibit as high confidence as in-distribution samples. Also, the sharp transition around decision boundaries indicates the vulnerability of adversarial attacks (Goodfellow et al., 2015). In the case of Mixup, we observe a gradual transition in the inter-class area having an intensity of 0.3 to 0.7 around the decision boundaries; however, a part of those observations originates from lower-confidence on samples.

In our experiment, Manifold Mixup has more precise decision boundaries comparing with the two previous cases, especially on the transition. However, overall confidence is biased to be lower than the others. One possible explanation for this is under-fitting since its complicate training prevents sufficient convergence for the conventional learning procedure (Thulasidasan et al., 2019). Last, our method, LogitMix, leads to improve hidden representation by having high-confidence on samples as well as gradual transition over inter-class regions as a supportive clue of a well-calibrated model of confidence estimation.

4.2 Combination of losses

To provide weak supervision for the logit of mixed data with a mixture of two logits from separate inferences, the supervision for samples from data distribution is needed. Although we exploit the widely-used classification loss (*i.e.*cross-entropy loss) for the basic LogitMix method, the term can be replaced with other supervision loss like $\mathcal{L}_{mix}$. To confirm the feasibility of loss combination, we measure the classification accuracy with ResNet50 on CIFAR100 dataset, which consists of 50,000 images with 100 classes. The more details for training is in Section 5. To guarantee the reliability of the experiment, we performed training five times for the same condition, and report the average of the results.

As our method is independent with the classification loss in Eq.3, we can freely choose $\mathcal{L}_{cls}$. To analyse the effect of combinations of classification loss, we examined the experiment with CIFAR-100. We simply changed the combinations of losses retaining other configurations the same. Table 1

Name	Loss	Accuracy(%)	Name	Loss	Accuracy(%)
Vanila	$\mathcal{L}_{cls}$	78.32 ± 0.07	LogitMix$_c$	$\mathcal{L}_{cls} + \mathcal{L}_{sim}$	80.11 ± 0.09
Mixup	$\mathcal{L}_{mix}$	79.82 ± 0.08	LogitMix$_m$	$\mathcal{L}_{mix} + \mathcal{L}_{sim}$	80.51 ± 0.08
			LogitMix$_{cm}$	$\mathcal{L}_{cls} + \mathcal{L}_{mix} + \mathcal{L}_{sim}$	80.96 ± 0.08

Table 1: Accuracy of CIFAR-100 under various combinations of losses. We achieved consistently better accuracy by adding the proposed loss $\mathcal{L}_{sim}$ on existing classification losses. The combination of all three losses (the last row) record the best accuracy.

shows results. Firstly, we combined our method with conventional cross entropy loss ($\mathcal{L}_{cls}$, Eq. 3) and denoted as LogitMix$_c$. By adding our similarity loss, the accuracy is increased largely. The accuracy of this combination is better than that of using Mixup loss Zhang et al. (2018) ($\mathcal{L}_{mix}$, Eq. 5). It is also possible to use Mixup loss with our method as our method is not depend on a classification loss.

$$\mathcal{L}_{cls} \;\; = \mathcal{H}(\tilde{x}, y), \qquad \mathcal{L}_{mix} = \lambda \mathcal{H}(\tilde{y}_{mix}, y_1) \; + (1-\lambda)\mathcal{H}(\tilde{y}_{mix}, y_2), \tag{5}$$

If our method is combined with Mixup (LogitMix$_m$), its accuracy is also increased and this result shows that our method is a general method for boosting the accuracy of the network regardless of the classification loss. When alpha is bigger than 1 (*i.e.* uniform distribution), the probability of seeing the original sample is very low. Since $\mathcal{L}_{cls}$ could give feedbacks the original sample, $\mathcal{L}_{mix}$ with $\mathcal{L}_{cls}$ scores 80.08 ± 0.49 that is higher than $\mathcal{L}_{mix}$ only. However, we confirm that the training with a combination of two supervision losses is less stable (*i.e.* high variance) than the combination of the supervision (*i.e.* $\mathcal{L}_{cls}$ or $\mathcal{L}_{mix}$) and weak supervision losses (*i.e.* $\mathcal{L}_{sim}$). The final combination is to use all of these losses (*i.e.* $\mathcal{L}_{cls} + \mathcal{L}_{mix} + \mathcal{L}_{sim}$, LogitMix$_{cm}$). As these three losses operate differently, the highest accuracy could be achieved than any other combination. Compared to the Mixup, it was possible to achieve high performance with a healthy margin. As using these three losses the most effective combination of using our method, we used this combination (LogitMix$_{cm}$) as a default configuration of our method for the rest of the paper.

5 Experiments

In this section, we evaluate LogitMix in various tasks and comparing with the competitors. We stress that LogitMix can be combined with any mixing-based data augmentation techniques (Verma et al., 2019; Yun et al., 2019; Zhang et al., 2018) and enjoy their performance gains. It is simply done by incorporating our similarity loss term in their objective. First experiments assess the ability to improve prediction accuracy and confidence calibration. The performance gain in both aspects implies that the trained model better predicts the true posterior distribution; thus it better reveals the inter-class correlations of the data distribution. Due to the page limit, several experimental results are moved to Appendix B and C. For example, we examine the robustness of the model trained with LogitMix (Appendix B). This result can quantify how LogitMix is successful in learning the vicinity of boundaries. We also conduct an ablation study by changing α for an empirical understanding of LogitMix (Appendix C).

Model architecture. We select five CNN architectures as backbone networks: three of them are conventional CNNs (*i.e.*, VGGNet (Simonyan & Zisserman, 2015), ResNet (He et al., 2016), and ResNeXt (Xie et al., 2017)), and the others are light-weight CNNs (*i.e.*, MobileNetV2 (Sandler et al., 2018) and ShuffleNet (Ma et al., 2018)).

Datasets. we validate the effectiveness of LogitMix on three benchmark datasets, ranging from small to large-scale: CIFAR100 (Krizhevsky & Hinton (2009), 32×32 RGB images in 100 classes), TinyImageNet (CS231N (2017), 64×64 RGB images in 100 classes) and ILSVRC2015 (Russakovsky et al. (2015), 256×256 RGB images in 1000 classes).

Evaluation protocol. All networks are trained by a stochastic gradient decent optimization with the momentum of 0.9. All methods for our comparison follow the same training schedule along with the dataset. For CIFAR100, we set an initial learning rate to 0.1 and decay the learning rate by 0.2 at every 60, 120, 160, and 200 epoch. In TinyImageNet and ILSVRC2015, we set the initial learning rate to 0.1 and decay by 0.1 at 75, 150, and 225 epoch. Because light-weight models have a different ideal training scheme, we follow the procedure described in their papers. In order to regularize the

Dataset	Network	Metric	Vanila	Mixup	LogitMix$_{cm}$	Cutmix	LogitMix$_{cc}$
CIFAR100	VGG16	Acc	74.30	75.02	76.22 (+1.20)	75.34	76.10 (+0.76)
		ECE	0.176	0.060	0.035 (-0.025)	0.051	0.062 (+0.011)
		OE	0.154	0.035	0.025 (-0.010)	0.022	0.008 (-0.014)
	ResNet50	Acc	78.32	79.82	80.96 (+1.14)	80.57	81.02 (+0.45)
		ECE	0.087	0.040	0.014 (-0.026)	0.078	0.073 (-0.005)
		OE	0.073	0.028	0.003 (-0.025)	0.064	0.060 (-0.004)
	ResNeXt50	Acc	79.18	81.10	81.63 (+0.53)	81.16	81.46 (+0.30)
		ECE	0.069	0.042	0.021 (-0.021)	0.059	0.032 (-0.027)
		OE	0.057	0.001	0.000 (-0.001)	0.047	0.023 (-0.024)
	MobileNetV2	Acc	69.69	69.98	73.90 (+3.92)	68.82	69.91 (+1.09)
		ECE	0.061	0.091	0.048 (-0.043)	0.050	0.049 (-0.001)
		OE	0.042	0.000	0.000 (0.000)	0.000	0.000 (0.000)
	ShuffleNetV2	Acc	72.17	74.17	75.53 (+1.36)	73.60	73.73 (+0.13)
		ECE	0.079	0.060	0.042 (-0.018)	0.016	0.023 (+0.007)
		OE	0.060	0.000	0.000 (-0.000)	0.002	0.000 (-0.002)
TinyImagenet	ResNet50	Acc	66.6	68.34	70.71 (+2.37)	69.08	69.87 (+0.79)
		ECE	0.098	0.032	0.030 (-0.002)	0.029	0.034 (+0.005)
		OE	0.076	0.022	0.010 (-0.012)	0.015	0.005 (-0.010)
	MobileNetV2	Acc	57.62	59.55	62.12 (+2.57)	53.54	57.66 (+4.12)
		ECE	0.073	0.091	0.032 (-0.059)	0.094	0.082 (-0.012)
		OE	0.045	0.019	0.000 (-0.019)	0.000	0.000 (0.000)
ILSVRC2015	ResNet50	Acc	76.13	77.37	78.38 (+1.01)	78.43	78.51 (+0.08)
		ECE	0.370	0.041	0.028 (-0.013)	0.028	0.020 (-0.008)
		OE	0.030	0.003	0.001 (-0.002)	0.029	0.029 (0.000)

Table 2: Comparison ACC, ECE, OE on Cifar, TinyImagenet. LogitMix$_{cm}$ and LogitMix$_{cc}$ represents the combination of $\mathcal{L}_{cls}$, $\mathcal{L}_{sim}$ losses with Mixup and Cutmix respectively. The blue (or red color) text represents the increase (or decrease) in value.

model, we exploit weight decaying with $4e{-}5$ for CIFAR100 and $1e{-}4$ for the others. Each model is trained with the batch size of 128, 200, and 256 for CIFAR100, TinyImageNet, and ILSVRC datasets.

5.1 Classification accuracy with expected calibration error

We showed that our method improves classification accuracy (*i.e.* high confidence on training samples) and helps to learn the gradual transition of probability between two classes (*i.e.* appropriate confidence on samples from off-the-data distribution) with a toy example (see Section 4.1), which is a supportive indication of well-calibration. In this section, we conduct experiments with a more realistic scenario. For the quantitative analysis of the confidence calibration, we used two popular metrics, the expected calibration error (ECE, Naeini et al. (2015)) and the overconfidence error (OE, Thulasidasan et al. (2019)). Please refer Appendix A for the computational model for each metric.

The ECE represents an average difference between true confidence and predicted confidence. If ECE is zero, it means the network is correctly calibrated. The OE is similar to ECE, but it only measures the confidence difference when it indicates over-confident. The over-confidence is particularly a critical factor in high-risk systems; thereby this metric is a good indicator to assess system reliability for high-risk applications. These two measures are calculated on validation sets.

Table 2 shows the experimental results using various networks and datasets. We consistently achieved better accuracy and confidence calibration after combining our method. Especially, our gain in prediction accuracy is substantial where the gap between the baseline and the baseline combining with LogitMix is as much as the gap between the vanilla and the other competitors. As a result, our method surpasses the performance of existing methods in most experimental conditions.

One crucial remark can be made via the experiment with a compact model such as MobileNetV2. Generally speaking, Mixup-like approaches act as an augmentation method, which populates training examples to prevent over-fitting. However, if it injects examples far from the training distribution, such an augmentation can induce under-fitting. Under-fitting normally does not degrade the performance of high-capacity networks, but it can hurt the performance of the low-capacity network. As the

Method	VWCI Seo et al. (2019)	Mixup	LogitMix$_{cm}$ (Ours)
Acc	73.87 (+0.09)	75.02 (+0.72)	76.22 (+1.92)
ECE	0.098 (-0.089)	0.057 (-0.116)	0.035 (-0.141)

Table 3: Comparison with other confidence calibration methods. The blue (or red color) text represents the increase (or decrease) in value.

MobileNetV2 is the low-capacity model, it requires less regularization compared to large models, and it might be sufficient to apply weak regularization (*i.e.* a small weight decay). When CutMix is used for training MobileNetV2 on TinyImageNet, we observe severe performance degradation, and we speculate that the accuracy is decreased because strong regularization induces under-fitting.

In contrast, when LogitMix is combined with CutMix, this effect of under-fitting is substantially reduced, filling the degradation gap introduced by CutMix. From this result, we argue that our method does not penalize the vanilla training to prevent over-fitting. Instead, LogitMix helps the model to understand the hidden relationships by weak supervision. Because the hidden relationships can provide a reasonable interpretation for understanding the examples far from training distribution, LogitMix can prevent under-fitting as well. This observation is coherent with our motivation that the inter-class correlation helps to improve both the prediction accuracy and the estimate of predictive confidence.

5.2 Comparison with confidence calibration methods

In previous experiments, we showed our method can improve both prediction accuracy and confidence calibration simultaneously. As our method calibrates the prediction with a single inference, we did not compare with Bayesian approaches (MacKay, 1992; Neal, 1995) or stochastic approaches with multiple inferences (Gal & Ghahramani, 2016; Lakshminarayanan et al., 2017). The works that are most relevant to our method are Mixup (Thulasidasan et al., 2019; Zhang et al., 2018) and variance-weighted confidence-integrated Loss (VWCI, Seo et al. (2019)). In order to match the same recipe in the results reported by VWCI, we compared the methods for CIFAR-100 using VGG16 (Table 3). The result of VWCI is from the original paper, and the delta values are also shown in parentheses with the absolute values of accuracy and ECE as the reported baseline performance from VWCI is slightly different from our results. Compared with the other methods, our method achieved the highest accuracy and lowest calibration error; comparing with delta values, our results were the best of the three.

6 Conclusion

The idea of LogitMix is motivated by the human cognition that heavily relies on the relational information for making a decision. Based on this analogy, we proposed LogitMix, a novel training strategy that takes into account the inter-class relationships in model training by utilizing the mixed logit as weak supervision.

Based on extensive evaluations using various network architectures on four datasets (three popular benchmark datasets and one synthetic dataset), empirical analysis and ablation study, we have demonstrated various useful properties of LogitMix. 1) The hidden representations and decision boundary are improved by adopting our method while keeping the confidence of training examples. 2) LogitMix can improve the prediction accuracy and the estimate of predictive uncertainty simultaneously. We highlight that the accuracy gain by LogitMix is significant; our improvements over the competitors are as large as the gap between the vanila and competitor. Moreover, our method is effective in both high-capacity and low-capacity models although the competitor (CutMix) is not. This is an evidence that LogitMix improves generalization by preventing both the over-fitting and under-fitting. 3) LogitMix can be combined with any supervision losses (i.e., cross entropy, Mixup or CutMix) and enjoy their performance gains.

REFERENCES

Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pp. 2654–2662, 2014.

Stanford CS231N. Tiny imagenet visual recognition challenge, 2017. URL `https://tiny-imagenet.herokuapp.com/`.

Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1778–1785. IEEE, 2009.

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

Georgia Gkioxari, Ross Girshick, Piotr Dollár, and Kaiming He. Detecting and recognizing human-object interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8359–8367, 2018.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

Stephen Gould, Tianshi Gao, and Daphne Koller. Region-based segmentation and object detection. In *Advances in neural information processing systems*, pp. 655–663, 2009.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Hongyu Guo, Yongyi Mao, and Richong Zhang. MixUp as Locally Linear Out-of-Manifold Regularization. In *AAAI Conference on Artificial Intelligence*, volume 33, pp. 3714–3722, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on CVPR*, pp. 770–778, 2016.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. In *NIPS 2014 Deep Learning Workshop*, 2014.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

Le Hou, Chen-Ping Yu, and Dimitris Samaras. Squared earth mover's distance-based loss for training deep neural networks. In *NIPS Learning on Distributions, Functions, Graphs and Groups Workshop*, 2017.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pp. 6402–6413, 2017.

Vladimir Li and Atsuto Maki. Feature contraction: New convnet regularization in image classification. In *BMVC*, pp. 213, 2018.

Zinan Lin, Ashish Khetan, Giulia Fanti, and Sewoong Oh. Pacgan: The power of two samples in generative adversarial networks. In *Advances in Neural Information Processing Systems*, pp. 1498–1507, 2018.

Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 116–131, 2018.

David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013.

Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? *arXiv preprint arXiv:1906.02629*, 2019.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Radford M Neal. *BAYESIAN LEARNING FOR NEURAL NETWORKS*. PhD thesis, Citeseer, 1995.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017a.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. Regularizing neural networks by penalizing confident output distributions. In *International Conference on Learning Representations, ICLR*, 2017b.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015.

Ruslan Salakhutdinov, Antonio Torralba, and Josh Tenenbaum. Learning to share visual appearance for multiclass object detection. In *CVPR 2011*, pp. 1481–1488. IEEE, 2011.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

Seonguk Seo, Paul Hongsuck Seo, and Bohyung Han. Learning for single-shot confidence calibration in deep neural networks through stochastic inferences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9030–9038, 2019.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Cecilia Summers and Michael J. Dinneen. Improved mixed-example data augmentation. In *IEEE Winter Conference on Applications of Computer Vision*, pp. 1262–1270, 2019.

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*, pp. 2818–2826, 2016.

Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara. Data Augmentation using Random Image Cropping and Patching for Deep CNNs. In *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

Sunil Thulasidasan, Gopinath Chennupati, Jeff 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*, 2019.

Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. Between-class learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5486–5494, 2018.

Antonio Torralba, Kevin P Murphy, William T Freeman, et al. Sharing features: efficient boosting procedures for multiclass object detection. *CVPR (2)*, 3, 2004.

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 *Proceedings of the 36th International Conference on Machine Learning*, pp. 6438–6447. PMLR, 2019.

Lingxi Xie, Jingdong Wang, Zhen Wei, Meng Wang, and Qi Tian. Disturblabel: Regularizing cnn on the loss layer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4753–4762, 2016.

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*, pp. 1492–1500, 2017.

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 *International Conference on Computer Vision (ICCV)*, 2019.

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

Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2018.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

APPENDIX

A METRIC FOR CALIBRATION ERROR

As denoted in Section 5.1, for the quantitative analysis of the confidence calibration, we used two popular metrics, the expected calibration error (ECE) and the overconfidence error (OE). When B_m indicates the set of samples whose prediction scores (the winning softmax score) fall into bin B_m, the accuracy and confidence of B_m are:

$$\mathrm{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbf{1}\left(\max(\tilde{y}_i) = y_i\right), \qquad \mathrm{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \tilde{y}_i.$$

The expected calibration error computes as the weighted average of the absolute difference between $\mathrm{acc}(B_m)$ and $\mathrm{conf}(B_m)$:

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} |\mathrm{acc}(B_m) - \mathrm{conf}(B_m)|, \tag{6}$$

where n is the total number of samples across all bins. The ideally well-calibrated model is achieved when $\mathrm{ECE} = 0$ (*i.e.*$\mathrm{acc}(B_m) = \mathrm{conf}(B_m)$ for all bins).

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

where *max* denotes the class index which has the maximum value among the all classes in the predicted distribution.

B ROBUSTNESS

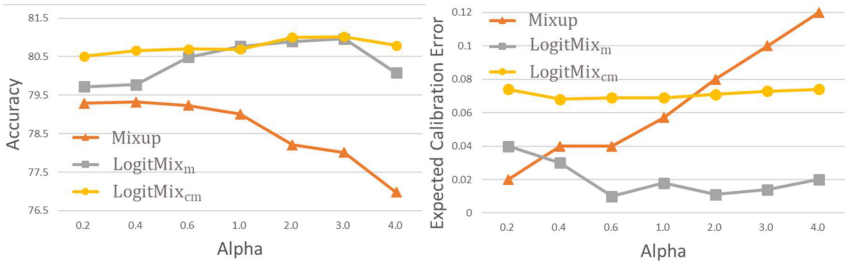
Recently, many studies address that DNN is vulnerable to adversarial examples. Although imperceptible noise is added to the sample, the model fails to predict the perturbed sample even with high confidence. To improve the robustness from adversarial attacks, some of the approaches utilize the augmented samples by adversarial training, adding noise, or erasing the subregion of the sample. Since the mixing-based augmentation techniques also generate the sample which is in between samples, the model could learn the near boundary samples when the ratio between the sample is a half. Hence, the models trained with mixed samples are expected to achieve improvement in robustness.

In order to evaluate the model robustness, we utilize the two kinds of adversarial example. One generated with the Fast Gradient Sign Method (FGSM) Goodfellow et al. (2015), which takes the sign of a gradient obtained from the trained model and then perturbs along the gradient direction. We set the attack step size as 8/255 without target setting. The other is Natural Adversarial Example (NAE) Zhao et al. (2018). NAE which is collected from the ILSVRC2015 is real-word sample, but they significantly degrade the classification performance. Note that, to evaluate the model trained on TinyImagenet dataset, we compose the tiny-NAE dataset by finding the intersection between TinyImagenet and NAE, and then down-scale to match the size of TinyImagenet. Table S1 reports top-1 and top-5 accuracy after attack on three benchmark datasets. In all cases, LogitMix+ improve or at least preserve the robustness to the two different adversarial attacks compared to their baseline.

C THE EFFECT OF ALPHA (α)

In mixed-base approaches Tokozume et al. (2018); Verma et al. (2019); Yun et al. (2019); Zhang et al. (2018), the mixing rate is controlled by α which changes the shape of probability distribution for λ. As a result, This hyper-parameter has a high impact on performance. Especially, using a large α value degrades the accuracy largely in Mixup Thulasidasan et al. (2019); Zhang et al. (2018). Using large α means that the image is likely to be mixed in half and can be a severe augmentation while training. Another disadvantage of using large α is that the network has little chance of seeing the original sample. Then, it raises a natural question: *is a large α also bad on LogitMix?*

To analyze the effect of the α, we ablated the performance of the network in terms of accuracy and ECE by change α using ResNet50 and CIFAR-100 dataset. Fig. S1(a) shows the effect of α on



(a) Accuracy (b) Expected calibration error

Figure S1: Ablation study on the effect of α.

accuracy. As expected, the accuracy starts to drop with high α in Mixup. However, the accuracy is proportional to the α value in our method. This is not the effect of explicit supervision of the original sample by using $\mathcal{L}_{cls}$ (Eq. 5). The tendency is similar even though not using $\mathcal{L}_{cls}$ (LogitMix$_m$, orange line). This means that LogitMix can give implicit supervision of the original samples with almost-half-mixing samples, and our method transforms half-mixing into helpful augmentation, not the severe one. Extreme α values are also harmful to LogitMix as they reduce the training a lot near original samples. Following this experimental result, we choose three as a default value for α.

Regarding the confidence calibration, ECE is worsened as the α is increasing (Fig. S1(b)). We observed OE is decreasing as α is increasing, this means that ECE is increasing due to the under-confident, and it means that the network is overly regularized with severe augmentation. This result is consistent with the tendency with accuracy. Unlike Mixup, our method achieved good calibration result regardless of α.

Dataset	Network	Method	Vanila	Mixup	LogitMix$_{cm}$	Cutmix	LogitMix$_{cc}$
CIFAR100	ResNet50	FGSM	18.37	19.76	22.51	13.89	15.37
			38.34	40.66	42.24	33.76	35.19
	ResNeXt50	FGSM	17.96	18.76	20.77	12.62	14.22
			38.41	38.88	40.93	32.44	33.77
	MobileNetV2	FGSM	15.23	17.43	17.44	12.64	12.25
			35.10	38.18	38.87	30.88	30.65
	ShuffleNetV2	FGSM	17.15	19.28	19.60	14.07	14.18
			39.06	40.07	40.82	33.97	34.53
TinyImagenet	ResNet50	FGSM	11.79	15.39	18.10	12.25	13.96
			28.22	31.08	36.12	28.51	30.47
		NAE	1.67	1.54	2.01	2.12	2.65
			7.48	7.66	9.07	9.00	10.70
	MobileNetV2	FGSM	7.32	8.75	9.26	6.06	6.54
			21.44	23.75	24.10	18.08	19.57
		NAE	1.32	1.62	1.65	0.91	1.18
			6.88	7.00	7.03	6.17	7.64
ILSVRC2015	ResNet50	FGSM	23.54	35.71	37.40	38.094	38.68
			48.13	60.22	61.34	62.85	63.2
		NAE	0.09	0.09	0.10	0.12	0.11
			0.40	0.59	0.61	0.65	0.56

Table S1: Top-1 and Top-5 accuracy for the adversarial examples generated by FGSM and the natural adversarial examples (NAE). The blue (or red color) text represents the increase (or decrease) in value.