On the Calibration of Image Semi-Supervised Learning Models

Mehrab Mustafy Rahman, Jayanth Mohan, Tiberiu Sosea, Cornelia Caragea

Computer Science
University of Illinois Chicago
mrahm@uic.edu, jmoha11@uic.edu, tsosea2@uic.edu, cornelia@uic.edu

Abstract

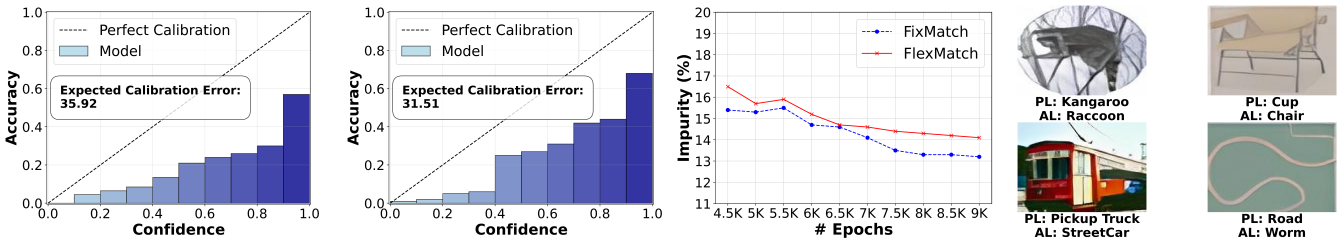
Semi-supervised learning (SSL) has demonstrated high performance in image classification tasks by effectively utilizing both labeled and unlabeled data. However, existing SSL methods often suffer from poor calibration, with models yielding overconfident predictions that misrepresent actual prediction likelihoods. Recently, neural networks trained with `mixup` that linearly interpolates random examples from the training set have shown better calibration in supervised settings. However, calibration of neural models remains under-explored in semi-supervised settings. Although effective in supervised model calibration, random mixup of pseudolabels in SSL presents challenges due to the overconfidence and unreliability of pseudolabels. In this work, we introduce CalibrateMix, a targeted mixup-based approach that aims to improve the calibration of SSL models while maintaining or even improving their classification accuracy. Our method leverages training dynamics of labeled and unlabeled samples to identify "easy-to-learn" and "hard-to-learn" samples, which in turn are utilized in a targeted mixup of easy and hard samples. Experimental results across several benchmark image datasets show that our method achieves lower expected calibration error (ECE) and superior accuracy compared to existing SSL approaches.

Code — https://github.com/mehrab-mustafy/CalibrateMix

Introduction

Deep neural networks (DNNs) (LeCun, Bengio, and Hinton 2015; Mathew, Amudha, and Sivakumari 2021) have achieved remarkable success across a wide range of computer vision tasks, including image classification (Lu and Weng 2007), object detection (Papageorgiou, Oren, and Poggio 1998), and semantic segmentation (Minaee et al. 2021). However, alongside accuracy, the predictive confidence of the models plays a vital role in real-world decision-making applications. For example, in critical applications such as autonomous driving, medical diagnosis, and disaster response, models must be accurate as well as reliably indicate when they are uncertain, so that additional safety measures can be triggered. For this reason, quantifying predictive uncertainty and calibration of the DNNs is a pivotal component toward building more reliable models. Despite strong performance, DNNs often suffer from poor calibration (Guo et al. 2017), which means that the predictive confidence likely overestimates the model's true accuracy. A key reason behind this is that modern DNNs are trained using one-hot encoded labels and the cross-entropy loss, which assumes that every training sample belongs with full certainty to a single class. This forces the model to assign the entire probability mass to a single class label, which in turn suppresses any expression of uncertainty even for "ambiguous" samples. To overcome the issues of overconfidence, label smoothing (Müller, Kornblith, and Hinton 2019) has been introduced. By softening the target distribution during training, label smoothing regularizes the model's output probabilities, encouraging it to remain uncertain where appropriate. More recently, Thulasidasan et al. (2019) explored the use of mixup training (Zhang et al. 2018) for improving model calibration which creates augmented samples through convex combinations of input samples and their labels. Mixup distributes its probability into two classes, which introduces entropy, prevents overconfidence, and has proven to be an effective tool in model calibration.

However, this approach primarily targets the fully supervised setting, which requires a large amount of labeled data. With the emergence of AI in all domains, it is impractical to obtain large amounts of annotated data for every domain. To solve this issue, Semi-supervised learning (SSL) (Chapelle, Scholkopf, and Zien 2009) can be an effective strategy to leverage large amounts of unlabeled examples during training to boost model performance. Despite mixup being successful in supervised learning, mixing up unlabeled examples in an SSL setting poses certain challenges due to the uncertainty of pseudo-label correctness, especially at the early iterations of training. This makes it critical to ensure the quality of pseudo-labels before incorporating them into the learning process. To ensure the quality of pseudo-labels, a popular SSL method to learn from unlabeled examples is pseudo-labeling (Lee et al. 2013), which leverages a model to make predictions on unlabeled examples and assign them pseudo-labels, which are in turn used as (pseudo) ground truth during training. To ensure that correct pseudo-labels are used for model training, modern SSL frameworks such as FixMatch (Sohn et al. 2020) and FlexMatch (Zhang et al. 2021) utilize high confidence thresholds to maintain data quality and filter out potentially incorrect examples. However, the calibration of these SSL models is not well studied, and we found empirical evidence



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

Figure 1: (a) FixMatch reliability diagram on CIFAR-100; (b) FlexMatch reliability diagram on CIFAR-100; (c) Impurity comparison on CIFAR-100; (d) Examples of incorrect pseudo-labels at the end of training. PL denotes the pseudo label predicted by the model, while AL denotes the ground-truth (actual) label. Mislabeling patterns for FlexMatch (left) and FixMatch (right) are shown.

that SSL models also suffer from poor calibration as shown in the reliability diagrams of FixMatch and FlexMatch on CIFAR-100 in Figures 1a and 1b, respectively. The diagrams, which plot accuracy as a function of confidence show that the confidence estimates of the models are not indicative of their correctness. Notably, FixMatch and FlexMatch predictions with confidences higher than 90% have less than 65% accuracy, contradicting the assumption that employing high confidence thresholds leads to high pseudo-label quality.

As shown in Figure 1c, the impurity of unlabeled data for both FixMatch and FlexMatch on CIFAR-100 (Krizhevsky, Hinton et al. 2009) is higher than 13%, indicating that more than 13% of the unlabeled data is utilized with incorrect pseudo-labels during training even at the later stages of training at 9000 epoch. Figure 1d shows examples of incorrect pseudo-labels introduced by both FixMatch and FlexMatch at the end of training. These incorrect pseudo-labels arise due to miscalibrated model predictions, manifested here as overconfidence in potentially incorrect predictions. This calibration gap in SSL can not be fully solved using random mixup. When incorrect pseudo-labels are used in mixup, the interpolation process can propagate label noise across training samples. This not only reinforces errors but also makes them harder to correct. For deep learning models that are highly over-parameterized and capable of achieving near-zero training error, such reinforcement of incorrect labels can lead to severe overfitting to mislabeled data. Thus, without addressing the underlying calibration issue and pseudo-label reliability, mixup can become a mechanism for error amplification rather than a regularization strategy. Hence, we investigate the calibration of SSL models under mixup and propose an enhanced mixup strategy.

In this paper, we propose CalibrateMix, a novel, targeted mixup-based (Zhang et al. 2018) framework that enhances the confidence calibration of SSL models. CalibrateMix monitors the training dynamics of unlabeled samples during training by keeping track of margins (Bartlett, Foster, and Telgarsky 2017; Elsayed et al. 2018; Sosea and Caragea 2023) of the outputs of the model. These margins are utilized to characterize each sample based on learning difficulty into two categories:"easy-to-learn" samples, which the model perceives to be correctly pseudo-labeled and "hard-to-learn" samples, which the model perceives to be potentially incorrect or ambiguous. Then, our method performs targeted mixup, combining easy-to-learn labeled samples with hard-to-learn unlabeled samples, as well as hard-to-learn labeled samples with easy-to-learn unlabeled samples. This mixup strategy has two main advantages: First, the presence of easy-to-learn samples ensures that the resulting pseudo-label of the mixed sample is qualitative while the presence of the hard-to-learn samples ensures a more difficult environment for the model to learn from, which was shown to benefit SSL training (Xie et al. 2020). Second, CalibrateMix is easy-to-use in practice and can be applied on top of the most popular SSL frameworks such as FixMatch or FlexMatch.

To showcase the benefits of our method, we carry out extensive experiments on well-established small-scale SSL setups - CIFAR10, CIFAR100 (Krizhevsky, Hinton et al. 2009), SVHN (Netzer et al. 2011), STL10 (Coates, Ng, and Lee 2011); and on two large-scale benchmarks - ImageNet (Deng et al. 2009) and WebVision (Li et al. 2017), where we show that CalibrateMix outperforms strong baselines and previous works both in terms of accuracy and expected calibration error (ECE). Notably, CalibrateMix boosts the performance of FixMatch by 8.51% in ECE and by 1.54% in error rate on CIFAR-100 using only 25 labels per class. On the larger datasets, e.g., WebVision, we also see an improvement of 2% ECE on FixMatch and FlexMatch without loss in error rates.

Our contributions are as follows:

1. We introduce CalibrateMix, a novel targeted mixup framework that is primarily designed to improve the confidence calibration of semi-supervised learning (SSL) models.
2. Through extensive experiments on standard benchmark image datasets, we demonstrate that CalibrateMix outperforms existing methods in terms of Expected Calibration Error (ECE), while also often achieving lower error rates compared to multiple state-of-the-art SSL setups. Our method is compatible with any SSL frameworks, and we demonstrate its effectiveness by improving both calibration and, in multiple cases, accuracy when integrated with FixMatch, FlexMatch, and SoftMatch.
3. We carry out extensive ablation studies to understand the contribution of various components to the success of CalibrateMix, as well as an error analysis to analyze the

calibration and correctness of predictions in practice.

Related Work

Despite achieving high accuracy, modern deep networks frequently remain miscalibrated, producing overly confident predictions (Guo et al. 2017; Minderer et al. 2021). Several effective methods have been proposed in supervised learning settings to mitigate calibration issues proactively. Dropout, initially introduced as a regularization technique, doubles as a Bayesian uncertainty estimator at inference via Monte Carlo sampling, providing improved calibration by averaging predictions across multiple stochastic forward passes (Srivastava et al. 2014). Another well established approach is label smoothing, which softens the one-hot encoded targets, thereby discouraging extreme predictions and significantly enhancing calibration and generalization performance, especially under limited or imbalanced data scenarios (Müller, Kornblith, and Hinton 2019). Mixup training, another promising approach, interpolates pairs of training samples and their labels, generating augmented data points that promote smoother predictive boundaries. Thulasidasan et al. (Thulasidasan et al. 2019) explicitly verified mixup's effectiveness in reducing expected calibration error (ECE), highlighting its capability in generating well-calibrated confidence scores and robust predictive uncertainty. Further developments, such as RegMixup, have validated that Mixup based regularization not only enhances calibration but also robustness under distribution shifts, particularly beneficial in out-of-distribution (OOD) detection tasks (Pinto et al. 2022).

Other studies demonstrated that employing focal loss, which emphasizes learning from challenging predictions, also effectively mitigates model overconfidence, enhancing calibration and model reliability (Mukhoti et al. 2020). While supervised calibration methods have matured significantly, translating these advancements directly to semi-supervised learning (SSL) remains challenging. SSL leverages limited labeled data alongside abundant unlabeled data to enhance model learning. Prominent SSL frameworks, such as FixMatch (Sohn et al. 2020), FlexMatch (Zhang et al. 2021), and SoftMatch (Chen et al. 2023), primarily rely on pseudo-labeling and consistency regularization. FixMatch employs fixed high confidence thresholds to filter pseudo-labels, improving overall accuracy but inadvertently exacerbating calibration issues by neglecting calibration aware decision-making. Similarly, FlexMatch introduces a dynamic curriculum based threshold adjustment, enhancing class specific learning but still neglecting explicit calibration mechanisms (Zhang et al. 2021). SoftMatch further improves pseudo-label quality through adaptive weighting strategies; however, calibration remains an implicit, secondary consideration rather than a deliberate design goal (Chen et al. 2023). Consequently, SSL models often remain miscalibrated, assigning overly confident pseudo-labels to unlabeled samples, which compromises their accuracy and robustness. Recent SSL approaches, such as MarginMatch (Sosea and Caragea 2023), SequenceMatch (Nguyen 2024), and FineSSL (Gan and Wei 2024), indirectly target pseudo-label reliability and robustness through pseudo-margin analysis, consistency across augmentations, and robust finetuning strategies, respectively. While these techniques implicitly reduce calibration errors, their main emphasis remains on enhancing pseudo-label quality or robustness rather than directly addressing calibration gaps. Thus, explicit calibration remains largely unaddressed. Motivated by this necessity, we propose CalibrateMix, a targeted mixup-based SSL approach explicitly designed to improve both model calibration and predictive accuracy.

CalibrateMix

Notation Let $D_L = \{(x_1, y_1), \cdots, (x_{B_L}, y_{B_L})\}$ be a batch of labeled samples of size B_L and $D_U = \{\hat{x}_1, \cdots, \hat{x}_{B_U}\}$ be a batch of unlabeled samples of size B_U.

Background

Most deep neural networks (DNNs) trained for classification are trained using one-hot encoded labels, where the entire probability mass is assigned to a single class. This leaves no room for uncertainty during training. As a result, models tend to become overconfident in their predictions. Hence, it is not surprising that modern DNNs are poorly calibrated (Guo et al. 2017). To prevent this overconfidence, Thulasidasan et al. (2019) explored the impacts of mixup training in supervised settings and found that mixup has proven to be effective in reducing miscalibration in supervised settings. Mixup training creates vicinity samples for training, effectively creating more samples for the model to learn from. The mixup augmented samples are generated by the following rule as mentioned in Zhang et al. (2018):

$$\tilde{x} = \gamma x_i + (1 - \gamma) x_j \tag{1}$$

$$\tilde{y} = \gamma y_i + (1 - \gamma) y_j \tag{2}$$

where, x_i and x_j are any two randomly selected samples from the train set and y_i and y_j are their corresponding one-hot labels. Mixup-augmented sample $\tilde{x}$ is generated by interpolating between x_i and x_j. Similarly, the corresponding label $\tilde{y}$ is obtained by mixing y_i and y_j. The mixup coefficient γ, comes from the Beta distribution, with the hyperparameter α controlling the level of interpolation between the two samples. Unlike standard one-hot labels, when $y_i \neq y_j$, the mixup label $\tilde{y}$ distributes the probability mass across two classes rather than a single class. This distribution introduces entropy and uncertainty into the training process, preventing the model from becoming overly confident in its predictions.

In SSL, selecting samples at random for mixup can be harmful because of the unreliability of pseudo-labels especially in the early stages of training when models are more prone to errors and because of the confirmation bias. This is particularly severe for difficult samples. From Figure 1, we observe that the pseudo-labels may remain incorrect at the end of training, and SSL models can produce high-confidence incorrect predictions. Such miscalibrated predictions not only hamper model learning progress but also risk propagating errors across training iterations. Given that pseudo-labeled data typically outnumbers labeled data in SSL settings, random pairing during mixup increases the chance of combining two erroneous pseudo-labels. This can yield misleading targets that further reinforce incorrect representations. Hence, to reduce the impact of noisy labels in training and ensure quality,

it is important to avoid mixup between two difficult samples or two pseudo-labeled samples.

Proposed Approach: CalibrateMix

To address the limitations of standard random mixup in SSL and to deal with the propagation of overconfident or noisy pseudo-labels we propose **CalibrateMix**, a targeted mixup-based framework. CalibrateMix performs a controlled mixup between labeled and pseudo-labeled samples guided by their learning difficulty to generate higher-quality augmented training samples. The inclusion of labeled samples in the mixup ensures that no two pseudo-labeled samples take part in the mixup at the same time, while the inclusion of pseudo-labeled samples introduces entropy and uncertainty. This helps prevent the reinforcement of noise and encourages the model to express necessary uncertainty, leading to better calibrated confidence estimates. Hence, the core advantage of CalibrateMix lies in its structured pairing strategy that mixes labeled and unlabeled samples based on their learning difficulty. To quantify the learning difficulty of the samples of the labeled data we monitor the Area Under the Margin (AUM) (Pleiss et al. 2020) of the outputs of the model at each iteration. For the unlabeled data, we monitor the Average Pseudo Margin (APM) (Sosea and Caragea 2023) of the outputs of the model at each iteration. We do this because relying solely on the model's current prediction confidence is insufficient. Confidence at a single iteration does not reliably reflect the sample's margin or the correctness of its pseudo-label, as shown in Sosea and Caragea (2023). To calculate APM for an unlabeled sample $\hat{x}_i$, we use pseudo-margins defined as:

$$\text{PM}_c^t(\hat{x}_i) = z_c(\hat{x}_i) - \max_{j \neq c} z_j(\hat{x}_i) \tag{3}$$

where at iteration t, $z_c(\hat{x}_i)$ is the logit corresponding to sample $\hat{x}_i$ for assigned pseudo-label c and $\max_{j \neq c} z_j(\hat{x}_i)$ is the largest other logit corresponding to a label j other than c for the same sample $\hat{x}_i$. We use the pseudo-labels at the current iteration t as the "ground-truth". Then, the average pseudo-margin (APM) for the unlabeled sample $\hat{x}_i$ with pseudo-label c at iteration t is defined as follows:

$$\text{APM}_c^t(\hat{x}_i) = \frac{1}{t} \sum_{e=1}^{t} (\text{PM}_c^e(\hat{x}_i)) \tag{4}$$

An important consideration here is that for any previous iteration t', if the pseudo-label was c' (where $c \neq c'$) the pseudo-margin $\text{PM}^{t'}$ is calculated with respect to c' and also the APM is averaged from 1 to t' with respect to c'. In practice, we maintain a vector of pseudo-margins for all classes accumulated over the training iterations and dynamically retrieve the accumulated pseudo-margin value of the argmax class c to obtain the APM_c^t at iteration t.

To handle old pseudo-margin deprecation across large number of iterations, we use an exponential moving average of pseudo-margins to place higher importance on recent iterations. Hence, APM follows:

$$\text{APM}_c^t(\hat{x}_i) = \text{PM}_c^t(\hat{x}_i) \cdot \frac{\delta}{1+t} + \text{APM}_c^{t-1}(\hat{x}_i) \cdot \left(1 - \frac{\delta}{1+t}\right) \tag{5}$$

Similar to Sosea and Caragea (2023), we set the smoothing parameter $\delta = 0.997$.

For both labeled and unlabeled data, samples with larger AUM and APM values, respectively, are considered easier for the model to learn, and those with smaller values are generally ambiguous, harder to learn, or mislabeled. These hard to learn or ambiguous samples are often the cause of overconfidence and errors. Hence, in the mixup, we include one easy sample and one hard sample. The easy-to-learn sample ensures that the resulting pseudo-label of the mixed sample is qualitative, while the presence of the hard-to-learn sample ensures more uncertainty for the model. To perform these difficulty-aware splits, we partition both labeled and pseudo-labeled data at each iteration into "easy-to-learn" and "hard-to-learn" subsets. This categorization is based on the batch medians of AUM for labeled samples (τ_L) and APM for pseudo-labeled samples (τ_U), following (Park and Caragea 2022) but in a batch-wise manner, effectively distinguishing samples by their learning difficulty.

After obtaining the difficulty-aware asymmetric splits, mixup is then performed as follows: for each easy labeled sample, we randomly select one from its *top-k* most dissimilar hard pseudo-labeled samples, and for each hard labeled sample, we do the same with the *top-k* most dissimilar easy pseudo-labeled samples. This dissimilarity based sample selection for mixup better simulates the out-of-domain (OOD) distribution by producing mixup augmented examples that deviate from the in-domain distribution. Furthermore, randomly selecting a sample from the *top-k* most dissimilar samples introduces diversity across iterations and avoids overfitting. As a result, the model becomes less likely to make overconfident predictions on ambiguous inputs, thereby improving calibration. For the mixup, we compute a convex combination of the two inputs using weights γ and $1 - \gamma$ as follows:

$$\tilde{x_1} = \gamma x_{le} + (1-\gamma)\hat{x}_{uh} \tag{6}$$

$$\tilde{y_1} = \gamma y_{le} + (1-\gamma)\hat{y}_{uh} \tag{7}$$

$$\tilde{x_2} = \gamma x_{lh} + (1-\gamma)\hat{x}_{ue} \tag{8}$$

$$\tilde{y_2} = \gamma y_{lh} + (1-\gamma)\hat{y}_{ue} \tag{9}$$

where x_{le}, y_{le} are easy labeled samples and their ground-truth labels, x_{lh}, y_{lh} are hard labeled samples and their ground-truth labels, $\hat{x}_{ue}, \hat{y}_{ue}$ are easy pseudo-labeled samples and their pseudo-labels, and $\hat{x}_{uh}, \hat{y}_{uh}$ are hard pseudo-labeled samples and their pseudo-labels.

The final model input includes labeled, unlabeled, and mixup-augmented data. The total loss $\mathcal{L}$ is computed as:

$$\mathcal{L} = \mathcal{L}_L + \lambda_U \cdot \mathcal{L}_U + \mathcal{L}_{\text{mixup}} \tag{10}$$

where $\mathcal{L}_L$ is the supervised loss on labeled data, $\mathcal{L}_U$ is the unsupervised loss on the unlabeled data and λ_U is a hyper-parameter that controls the weight of $\mathcal{L}_U$ in the total loss $\mathcal{L}$, and $\mathcal{L}_{\text{mixup}}$ is the loss on mixup-augmented samples. In experiments, we set λ_U equals 1 consistent with previous approaches (Sohn et al. 2020; Chen et al. 2023).

The CalibrateMix algorithm is shown in Algorithm 1.

Experimental Analysis

We test the performance of our approach across a variety of image benchmarks by following standard SSL settings (Chen

Algorithm 1: CalibrateMix

Require: Labeled batch D_L; unlabeled batch D_U; model θ; number of classes C; weak augmentation π; strong augmentation Π; mixup coefficient γ; number of dissimilar samples k; confidence threshold ω; labeled-to-unlabeled loss ratio λ; medians of labeled and unlabeled batches τ_L, τ_U; current iteration t.

Compute pseudo-labels for the unlabeled batch: $\hat{y}_i^t = \arg\max_{c\in\{1,\dots,C\}} p_\theta(\pi(\hat{x}_i))$ for $\hat{x}_i \in D_U$ and build $\hat{D}_U = \{(\hat{x}_i, \hat{y}_i^t) \mid \hat{x}_i \in D_U\}$

Calculate AUM^t for samples in D_L and APM^t for samples in D_U

Split the labeled and unlabeled batch based on samples' learning difficulty:

$D_{L_{easy}} \leftarrow \{(x_i, y_i) \in D_L \mid \text{AUM}^t(x_i, y_i) \geq \tau_L\}$

$D_{L_{hard}} \leftarrow \{(x_i, y_i) \in D_L \mid \text{AUM}^t(x_i, y_i) < \tau_L\}$

$\hat{D}_{U_{easy}} \leftarrow \{(\hat{x}_i, \hat{y}_i^t) \in \hat{D}_U \mid \text{APM}^t(\hat{x}_i, \hat{y}_i^t) \geq \tau_U\}$

$\hat{D}_{U_{hard}} \leftarrow \{(\hat{x}_i, \hat{y}_i^t) \in \hat{D}_U \mid \text{APM}^t(\hat{x}_i, \hat{y}_i^t) < \tau_U\}$

Mix easy labeled with hard unlabeled and hard labeled with easy unlabeled following Eqs (6), (7), (8), (9):

$\tilde{D}_{Mix1} \leftarrow \text{Mixup}(D_{L_{easy}}, \textit{top-k dissim. from } \hat{D}_{U_{hard}})$

$\tilde{D}_{Mix2} \leftarrow \text{Mixup}(D_{L_{hard}}, \textit{top-k dissim. from } \hat{D}_{U_{easy}})$

$D_M \leftarrow \tilde{D}_{Mix1} + \tilde{D}_{Mix2}$

Optimize total loss: $\mathcal{L} \leftarrow \mathcal{L}_L + \lambda \cdot \mathcal{L}_U + \mathcal{L}_{Mixup}$, where

$\mathcal{L}_L = \frac{1}{|D_L|} \sum_{k=1}^{|D_L|} H(y_k, p_\theta(\pi(x_k)))$

$\mathcal{L}_U = \frac{1}{|D_U|} \sum_{k=1}^{|D_U|} \mathbb{1}(\max(p_\theta(\pi(\hat{x}_k))) \geq \omega) \cdot H(\hat{y}_k^t,\ p_\theta(\Pi(\hat{x}_k)))$

$\mathcal{L}_{Mixup} = \frac{1}{|D_M|} \sum_{k=1}^{|D_M|} H(\tilde{y}_k, p_\theta(\tilde{x}_k))$

et al. 2023; Wang et al. 2022). Specifically, we conduct experiments varying the amounts of labeled samples on CIFAR-10, CIFAR-100 (Krizhevsky, Hinton et al. 2009), SVHN (Netzer et al. 2011), STL-10 (Coates, Ng, and Lee 2011), ImageNet (Deng et al. 2009), and WebVision (Li et al. 2017). For the CIFAR-10, CIFAR-100, SVHN, and STL-10 datasets, we follow Sohn et al. (2020) and randomly select a small number of labeled samples ranging from 4 labels per class up to 400 labels per class and treat the remaining samples as unlabeled data, except for STL-10, which has its own set of unlabeled samples. For the ImageNet and WebVision datasets, we use 10% of the available labeled samples as labeled data and the remaining 90% as unlabeled data. In addition to the SSL setups, we also show results on the Fully Supervised setting.

We run all our experiments three times and report the average ECE and error rates, as well as their standard deviations. Similar to Sohn et al. (2020), we utilize Wide Residual Networks (Zagoruyko 2016) for small-scale datasets: WRN-28-2 for CIFAR-10 and SVHN, WRN-37-2 for STL-10 and WRN-28-8 for CIFAR-100; and use ResNet-50 (He et al. 2016) for the large-scale ImageNet and WebVision. Additionally, we utilize the SGD optimizer with a momentum of 0.9 and an initial learning rate of 0.03. We use a cosine annealing schedule to dynamically adjust the learning rate over a total of 2^{20} training steps. For each dataset, the batch size for labeled data and mixup data is set to 64, while the batch size for unlabeled data is configured to be seven times larger than that of the labeled data. We also use weak (flip and shift) and strong (RandAugment (Cubuk et al. 2020)) augmentations for the unlabeled data. We set mixup coefficient γ to be 0.4 similar to prior work (Thulasidasan et al. 2019). For CalibrateMix, we include warmup during training with the warmup phase consisting of approximately 100 epochs.

Performance on CIFAR-10, CIFAR-100, STL-10, SVHN

We compare CalibrateMix to relevant SSL approaches: FixMatch (Sohn et al. 2020), FlexMatch (Zhang et al. 2021), SoftMatch (Chen et al. 2023) and report ECE and top-1 error rates in Table 1. We further compare CalibrateMix against two calibration techniques: Random Mixup and Label Smoothing (LS) in Table 2. For Random Mixup, we randomly pair two samples from the combined pool of labeled and unlabeled data. Additionally, we apply LS following Müller, Kornblith, and Hinton (2019), with a factor of $\gamma = 0.1$ to the labeled, unlabeled, and mixup samples for all methods: FixMatch, FlexMatch, SoftMatch, and CalibrateMix (including combining CalibrateMix with LS). The results for Label Smoothing are denoted as 'LS' and the results with Random Mixup are denoted as 'RandomMixup' in Table 2.

Across all datasets, CalibrateMix consistently improves model calibration and, in many cases, enhances classification accuracy as can be seen from Table 1. For example, when integrated with FixMatch, CalibrateMix yields better-calibrated models. On CIFAR-10 with only 4 labels per class, CalibrateMix achieves a reduction in ECE by 2.18%. On the more challenging CIFAR-100 dataset with 25 labels per class, the improvements are more significant, reducing ECE by 8.51% and error rate by 1.54%. On the more realistic STL-10 dataset, CalibrateMix reduces ECE by 2.73%, further demonstrating its robustness. CalibrateMix also significantly enhances FlexMatch, particularly under low-label settings. Furthermore, integrating CalibrateMix with SoftMatch provides significant gains. On CIFAR-100 with 25 labels per class, ECE is improved by 5.68%. On STL-10 with 4 labels per class, CalibrateMix achieves a 0.84% lower ECE and a 5.72% reduction in error rate compared to SoftMatch. We can also observe that CalibrateMix yields well calibrated models in the Fully Supervised setting.

CalibrateMix consistently demonstrates improvements in model calibration and often in error rates when compared to prior calibration strategies such as random mixup and label smoothing, as shown in Table 2. When applied to FixMatch, CalibrateMix outperforms random mixup across several settings. Notably, on CIFAR-100 and STL-10 with 4 labels per class, it achieves ECE reductions of 2.62% and 5.65%, along with error rate reductions of 6.48% and 3.39%, respectively. Furthermore, on CIFAR-100 with 25 labels per class, combining CalibrateMix with label smoothing results in a 4.48% reduction in ECE over using label smoothing alone with FixMatch. On FlexMatch, similar benefits are observed. When integrated with SoftMatch, CalibrateMix continues to enhance calibration by outperforming random mixup. For example, with 4 labels per class on CIFAR-100 and STL-10, CalibrateMix reduces ECE by 3.55% and 9.31%, and decreases error rates by 3.52% and 13.56%, respectively, compared to random mixup. Compared to label smoothing, ECE is reduced by 4.62% on CIFAR-100 and 3.21% on STL-10

Metric	Method	CIFAR-10			CIFAR-100			SVHN		STL-10	
		4	25	400	4	25	100	4	100	4	100
ECE	FixMatch	$5.42_{0.36}$	$2.85_{0.23}$	$2.68_{0.09}$	$35.88_{1.1}$	$21.19_{0.62}$	$15.73_{0.05}$	$\mathbf{1.02}_{0.02}$	$1.1_{0.04}$	$32.66_{1.29}$	$4.22_{0.37}$
	FixMatch + CalibrateMix (Ours)	$\mathbf{3.24}_{0.12}$	$\mathbf{1.97}_{0.13}$	$\mathbf{1.25}_{0.27}$	$\mathbf{35.43}_{0.05}$	$\mathbf{12.68}_{0.42}$	$\mathbf{8.74}_{0.6}$	$1.06_{0.07}$	$\mathbf{0.82}_{0.13}$	$\mathbf{29.93}_{0.88}$	$\mathbf{3.9}_{0.25}$
ECE	FlexMatch	$4.76_{0.95}$	$\mathbf{2.92}_{0.53}$	$2.98_{0.55}$	$31.51_{9.55}$	$29.54_{7.24}$	$18.55_{4.15}$	$10.15_{2.88}$	$7.81_{1.55}$	$35.62_{8.57}$	$9.45_{2.12}$
	FlexMatch + CalibrateMix (Ours)	$\mathbf{4.41}_{0.22}$	$2.94_{0.15}$	$\mathbf{2.95}_{0.24}$	$\mathbf{27.65}_{0.33}$	$\mathbf{28.23}_{0.27}$	$\mathbf{18.02}_{0.54}$	$\mathbf{10.01}_{0.08}$	$\mathbf{7.54}_{0.11}$	$\mathbf{32.45}_{0.23}$	$\mathbf{9.26}_{0.41}$
ECE	SoftMatch	$3.02_{0.21}$	$2.95_{0.53}$	$2.21_{0.03}$	$26.4_{1.2}$	$19.58_{0.1}$	$15.34_{0.32}$	$1.44_{0.03}$	$1.17_{0.07}$	$13.24_{0.06}$	$4.31_{0.4}$
	SoftMatch + CalibrateMix (Ours)	$\mathbf{1.69}_{0.15}$	$\mathbf{2.09}_{0.18}$	$\mathbf{1.40}_{0.26}$	$\mathbf{23.1}_{0.3}$	$\mathbf{13.9}_{0.1}$	$\mathbf{11.9}_{0.33}$	$\mathbf{0.70}_{0.05}$	$\mathbf{0.80}_{0.03}$	$\mathbf{12.4}_{0.08}$	$\mathbf{3.72}_{0.6}$
ECE	FullySupervised		$2.22_{0.13}$			$6.77_{0.67}$			$0.4_{0.04}$		$24.14_{0.38}$
	FullySupervised + CalibrateMix (Ours)		$\mathbf{1.84}_{0.22}$			$\mathbf{4.74}_{0.13}$			$\mathbf{0.29}_{0.11}$		$\mathbf{13.77}_{0.23}$
Error Rate	FixMatch	$7.29_{0.05}$	$\mathbf{4.91}_{0.02}$	$4.3_{0.02}$	$44.45_{0.15}$	$29.88_{0.17}$	$22.88_{0.03}$	$3.65_{0.23}$	$\mathbf{2.04}_{0.1}$	$\mathbf{36.34}_{0.4}$	$6.2_{0.07}$
	FixMatch + CalibrateMix (Ours)	$\mathbf{7.13}_{0.02}$	$4.97_{0.01}$	$\mathbf{4.26}_{0.08}$	$\mathbf{44.05}_{0.11}$	$\mathbf{28.34}_{0.41}$	$\mathbf{22.3}_{0.3}$	$\mathbf{3.63}_{0.11}$	$2.17_{0.03}$	$37.3_{0.12}$	$\mathbf{5.78}_{0.06}$
Error Rate	FlexMatch	$\mathbf{5.03}_{0.08}$	$\mathbf{4.98}_{0.07}$	$4.28_{0.02}$	$40.15_{1.87}$	$27.73_{0.35}$	$21.93_{0.33}$	$7.88_{1.23}$	$6.78_{1.15}$	$29.78_{4.01}$	$6.29_{0.54}$
	FlexMatch + CalibrateMix (Ours)	$5.04_{0.05}$	$4.99_{0.02}$	$\mathbf{4.23}_{0.05}$	$\mathbf{40.02}_{0.16}$	$\mathbf{26.33}_{0.31}$	$\mathbf{21.55}_{0.16}$	$\mathbf{7.56}_{0.14}$	$\mathbf{6.71}_{0.08}$	$\mathbf{29.03}_{0.43}$	$\mathbf{6.25}_{0.09}$
Error Rate	SoftMatch	$5.09_{0.07}$	$\mathbf{4.90}_{0.02}$	$4.16_{0.07}$	$37.11_{0.3}$	$\mathbf{26.76}_{0.1}$	$\mathbf{22.11}_{0.3}$	$2.59_{0.8}$	$2.09_{0.02}$	$20.90_{2.47}$	$6.10_{0.06}$
	SoftMatch + CalibrateMix (Ours)	$\mathbf{5.03}_{0.06}$	$4.93_{0.03}$	$\mathbf{4.05}_{0.04}$	$\mathbf{36.68}_{0.18}$	$26.86_{0.13}$	$22.21_{0.03}$	$\mathbf{2.58}_{0.01}$	$\mathbf{2.08}_{0.2}$	$\mathbf{15.18}_{0.63}$	$\mathbf{6.07}_{0.04}$
Error Rate	FullySupervised		$4.63_{0.03}$			$19.42_{0.17}$			$2.19_{0.01}$		$34.41_{0.17}$
	FullySupervised + CalibrateMix (Ours)		$\mathbf{4.51}_{0.08}$			$\mathbf{18.97}_{0.26}$			$\mathbf{2.17}_{0.04}$		$\mathbf{27.81}_{0.21}$

Table 1: Expected Calibration Errors (ECE, %) and Top-1 error rates (%) on CIFAR-10, CIFAR-100, SVHN, and STL-10 datasets by FixMatch, FlexMatch, SoftMatch and CalibrateMix (the lower the better). Values are reported in the format X_Y where X is the mean and Y is the Standard Deviation across 3 runs. Better scores in comparison are shown in **bold**.

on the 4 labels per class setting. These consistent reductions across different models and datasets highlight CalibrateMix as a strong solution to existing calibration techniques, making it an effective and versatile regularization method for SSL.

Performance on ImageNet and WebVision

We report the performance of CalibrateMix on two large-scale datasets: ImageNet (Deng et al. 2009) and WebVision (Li et al. 2017). We randomly sample 10% examples from the training set to be used as labeled samples and use the rest of the examples as unlabeled data. We show the results of FixMatch, FlexMatch, SoftMatch and FullySupervised in terms of ECE and Error rate in Table 3. We note that CalibrateMix considerably boosts the calibration of our models in all setups and yields small error rate improvements in most settings. Specifically, CalibrateMix outperforms SoftMatch by 1.79% ECE on WebVision and pushes the performance over FixMatch by 0.99% ECE on ImageNet. Additionally, CalibrateMix reduces ECE by 1.98% over FlexMatch on the WebVision dataset. When applied to the FullySupervised setting, it also reduces ECE by 1.50% compared to the base supervised model on WebVision. These results highlight the effectiveness of our approach in large-scale, real-world scenarios, leading to better-calibrated models.

Ablation Study

We conduct experiments to capture: (1) the effect of the targeted mixup on different setups such as how to mix labeled and unlabeled samples as well as with and without warmup, and (2) mixup with and without considering the dissimilar samples (without considering cosine similarity).

Effect of Different Mixup Strategies With and Without Warmup

We run this ablation on CIFAR-100 and STL-10 (4 labels per class) using SoftMatch, with results reported in Table 4. We denote "easy-to-learn" labeled samples as *LE*, "hard-to-learn" labeled samples as *LH*, and the corresponding unlabeled categories as *UE* and *UH*. Our first ablation compares CalibrateMix, which mixes only across labeled and unlabeled easy/hard sets (*LE+UH, LH+UE*), to a mixup strategy that mixes both across and within labeled/unlabeled sets (*LE+LH, LE+UH, LH+UE, UE+UH*). We evaluate this "mixup all" variant with and without warmup.

As shown in the table, CalibrateMix yields lower ECE and error rates than mixup all +warmup. Moreover, removing warmup significantly increases error rates on both CIFAR-100 and STL-10, despite producing slightly lower ECE on CIFAR-100.

Figure 2: Confidence comparison on images from the STL-10 dataset for **SoftMatch (SM)** and **CalibrateMix (CM)**. For each image, **P** denotes the predicted class, **A** the ground-truth (actual) class, and **C** the associated prediction confidence.

Effect of Selecting Dissimilar Samples

We evaluate our choice of using the *top-k* most dissimilar samples for mixup by experimenting SoftMatch + CalibrateMix with different values of k (5%, 10%, 15%). Results on CIFAR-100 and STL-10 (Table 4) show that k = 5% with cosine similarity yields the best performance. Removing cosine sim-

Metric	Method	CIFAR-10			CIFAR-100			SVHN		STL-10	
		4	25	400	4	25	100	4	100	4	100
ECE	**FixMatch + RandomMixup**	$5.26_{2.61}$	$3.19_{1.42}$	$1.72_{0.11}$	$38.05_{2.51}$	$14.78_{0.21}$	$\mathbf{8.72}_{1.91}$	$2.87_{0.25}$	$0.87_{0.06}$	$35.58_{3.61}$	$4.16_{0.30}$
	FixMatch + CalibrateMix (Ours)	$\mathbf{3.24}_{0.12}$	$\mathbf{1.97}_{0.13}$	$\mathbf{1.25}_{0.27}$	$\mathbf{35.43}_{0.05}$	$\mathbf{12.68}_{0.42}$	$8.74_{0.6}$	$\mathbf{1.06}_{0.07}$	$\mathbf{0.82}_{0.13}$	$\mathbf{29.93}_{0.88}$	$\mathbf{3.9}_{0.25}$
	FixMatch + LS	$4.07_{0.48}$	$2.52_{0.22}$	$2.69_{0.11}$	$34.11_{2.01}$	$17.37_{1.11}$	$7.56_{0.95}$	$1.23_{0.05}$	$1.19_{0.06}$	$32.96_{1.73}$	$4.02_{0.32}$
	FixMatch + CalibrateMix (Ours) + LS	$\mathbf{4.07}_{0.41}$	$\mathbf{2.06}_{0.23}$	$\mathbf{1.22}_{0.45}$	$\mathbf{31.7}_{1.33}$	$\mathbf{12.89}_{1.1}$	$\mathbf{7.11}_{0.8}$	$\mathbf{1.22}_{0.1}$	$\mathbf{0.99}_{0.13}$	$\mathbf{29.56}_{2.34}$	$\mathbf{3.9}_{0.28}$
ECE	**FlexMatch + RandomMixup**	$4.52_{0.18}$	$3.57_{0.31}$	$3.24_{0.02}$	$27.80_{0.08}$	$\mathbf{26.72}_{1.05}$	$21.79_{0.19}$	$10.45_{0.34}$	$7.80_{0.76}$	$\mathbf{31.80}_{1.38}$	$9.35_{0.23}$
	FlexMatch + CalibrateMix (Ours)	$\mathbf{4.41}_{0.22}$	$\mathbf{2.94}_{0.15}$	$\mathbf{2.95}_{0.24}$	$\mathbf{27.65}_{0.33}$	$28.23_{0.27}$	$\mathbf{18.02}_{0.54}$	$\mathbf{10.01}_{0.08}$	$\mathbf{7.54}_{0.11}$	$32.45_{0.23}$	$\mathbf{9.26}_{0.41}$
	FlexMatch + LS	$4.01_{0.32}$	$\mathbf{2.23}_{0.21}$	$2.96_{0.31}$	$29.41_{5.71}$	$29.03_{3.17}$	$18.17_{3.61}$	$\mathbf{9.65}_{1.16}$	$7.23_{1.34}$	$31.87_{4.5}$	$7.86_{1.54}$
	FlexMatch + CalibrateMix (Ours) + LS	$\mathbf{3.99}_{0.32}$	$2.24_{0.19}$	$\mathbf{2.91}_{0.14}$	$\mathbf{26.84}_{1.34}$	$\mathbf{27.53}_{01.05}$	$\mathbf{17.03}_{0.61}$	$9.67_{0.15}$	$\mathbf{7.01}_{0.16}$	$\mathbf{30.44}_{1.73}$	$\mathbf{7.1}_{0.34}$
ECE	**SoftMatch + RandomMixup**	$2.75_{0.64}$	$2.17_{0.10}$	$1.84_{0.04}$	$26.65_{1.12}$	$14.12_{0.32}$	$\mathbf{11.62}_{0.58}$	$2.44_{0.28}$	$0.84_{0.05}$	$21.71_{3.47}$	$4.07_{0.05}$
	SoftMatch + CalibrateMix (Ours)	$\mathbf{1.69}_{0.15}$	$\mathbf{2.09}_{0.18}$	$\mathbf{1.4}_{0.26}$	$\mathbf{23.1}_{0.3}$	$\mathbf{13.9}_{0.1}$	$11.9_{0.33}$	$\mathbf{0.7}_{0.05}$	$\mathbf{0.8}_{0.03}$	$\mathbf{12.4}_{0.08}$	$\mathbf{3.72}_{0.6}$
	SoftMatch + LS	$6.62_{0.32}$	$6.83_{0.3}$	$7.02_{0.08}$	$20.89_{0.40}$	$7.88_{0.37}$	$6.11_{0.35}$	$8.31_{0.34}$	$7.80_{0.23}$	$11.26_{3.44}$	$7.46_{0.5}$
	SoftMatch + CalibrateMix (Ours) + LS	$\mathbf{4.76}_{0.39}$	$\mathbf{5.70}_{0.37}$	$\mathbf{6.49}_{0.11}$	$\mathbf{16.27}_{0.44}$	$\mathbf{6.25}_{0.59}$	$\mathbf{4.49}_{0.19}$	$\mathbf{4.64}_{1.49}$	$\mathbf{7.32}_{0.06}$	$\mathbf{8.05}_{2.18}$	$\mathbf{7.01}_{0.25}$
Error Rate	**FixMatch + RandomMixup**	$8.98_{1.85}$	$6.89_{1.33}$	$4.35_{0.04}$	$50.53_{2.19}$	$\mathbf{27.17}_{0.33}$	$\mathbf{20.76}_{0.23}$	$\mathbf{2.48}_{0.04}$	$2.40_{0.13}$	$40.69_{5.30}$	$6.44_{0.11}$
	FixMatch + CalibrateMix (Ours)	$\mathbf{7.13}_{0.02}$	$\mathbf{4.97}_{0.01}$	$\mathbf{4.26}_{0.08}$	$\mathbf{44.05}_{0.11}$	$28.34_{0.41}$	$22.3_{0.3}$	$3.63_{0.11}$	$\mathbf{2.17}_{0.03}$	$\mathbf{37.3}_{0.12}$	$\mathbf{5.78}_{0.06}$
	FixMatch + LS	$7.41_{0.61}$	$5.06_{0.98}$	$4.32_{0.17}$	$44.57_{4.16}$	$29.82_{2.64}$	$23.62_{1.44}$	$3.71_{0.46}$	$2.11_{0.24}$	$\mathbf{36.43}_{2.32}$	$6.5_{0.28}$
	FixMatch + CalibrateMix (Ours) + LS	$\mathbf{7.13}_{0.61}$	$\mathbf{4.99}_{0.23}$	$\mathbf{4.27}_{0.25}$	$\mathbf{44.11}_{1.42}$	$\mathbf{28.64}_{1.24}$	$\mathbf{22.52}_{1.05}$	$\mathbf{3.64}_{0.31}$	$\mathbf{2.11}_{0.25}$	$37.54_{1.76}$	$\mathbf{5.72}_{0.42}$
Error Rate	**FlexMatch + RandomMixup**	$5.84_{0.34}$	$5.45_{0.08}$	$4.69_{0.05}$	$40.62_{0.88}$	$26.53_{0.43}$	$22.12_{0.13}$	$7.78_{0.45}$	$6.96_{0.36}$	$31.60_{6.21}$	$6.66_{0.13}$
	FlexMatch + CalibrateMix (Ours)	$\mathbf{5.04}_{0.05}$	$\mathbf{4.99}_{0.02}$	$\mathbf{4.23}_{0.05}$	$\mathbf{40.02}_{0.16}$	$\mathbf{26.33}_{0.31}$	$\mathbf{21.55}_{0.16}$	$\mathbf{7.56}_{0.14}$	$\mathbf{6.71}_{0.08}$	$\mathbf{29.03}_{0.43}$	$\mathbf{6.25}_{0.09}$
	FlexMatch + LS	$\mathbf{4.81}_{0.51}$	$4.72_{0.47}$	$\mathbf{4.01}_{0.34}$	$40.11_{1.64}$	$27.56_{1.15}$	$21.53_{1.04}$	$7.51_{0.31}$	$6.91_{0.37}$	$29.44_{1.27}$	$\mathbf{5.88}_{0.49}$
	FlexMatch + CalibrateMix (Ours) + LS	$4.83_{0.41}$	$\mathbf{4.71}_{0.37}$	$4.05_{0.22}$	$\mathbf{40.01}_{1.87}$	$\mathbf{26.02}_{1.46}$	$\mathbf{21.35}_{1.21}$	$\mathbf{7.45}_{0.48}$	$\mathbf{6.63}_{0.32}$	$\mathbf{28.66}_{0.28}$	$6.04_{0.16}$
Error Rate	**SoftMatch + RandomMixup**	$7.44_{0.96}$	$6.02_{0.36}$	$4.42_{0.08}$	$40.20_{0.6}$	$\mathbf{25.42}_{0.15}$	$\mathbf{21.58}_{0.86}$	$\mathbf{2.51}_{0.07}$	$2.64_{0.18}$	$28.74_{6.26}$	$\mathbf{5.78}_{0.14}$
	SoftMatch + CalibrateMix (Ours)	$\mathbf{5.03}_{0.06}$	$\mathbf{4.93}_{0.03}$	$\mathbf{4.05}_{0.04}$	$\mathbf{36.68}_{0.18}$	$26.86_{0.13}$	$22.21_{0.03}$	$2.58_{0.01}$	$\mathbf{2.08}_{0.2}$	$\mathbf{15.18}_{0.63}$	$6.07_{0.04}$
	SoftMatch + LS	$5.56_{0.46}$	$5.19_{0.12}$	$4.46_{0.09}$	$39.10_{0.46}$	$\mathbf{26.50}_{0.25}$	$22.13_{0.11}$	$3.14_{0.27}$	$3.07_{0.20}$	$24.57_{3.93}$	$\mathbf{5.63}_{0.08}$
	SoftMatch + CalibrateMix (Ours) + LS	$\mathbf{5.49}_{0.34}$	$\mathbf{5.09}_{0.14}$	$\mathbf{4.43}_{0.11}$	$\mathbf{37.61}_{1}$	$27.41_{1.18}$	$\mathbf{21.81}_{0.28}$	$\mathbf{2.9}_{0.18}$	$\mathbf{2.84}_{0.07}$	$\mathbf{19.24}_{5.76}$	$5.97_{0.12}$

Table 2: Expected Calibration Errors (ECE, %) and Top-1 error rates (%) on CIFAR-10, CIFAR-100, SVHN, and STL-10 datasets by Random Mixup, Label Smoothing (LS) and CalibrateMix (the lower the better). Values are reported in the format X_Y where X is the mean and Y is the Standard Deviation across 3 runs. Better scores in comparison are shown in **bold**.

Dataset	ImageNet		WebVision	
	ECE	Error	ECE	Error
Fixmatch	9.44	**43.54**	12.5	44.51
Fixmatch + Ours	**8.45**	43.61	**10.4**	**44.47**
Flexmatch	10.45	**42.9**	13.32	43.7
Flexmatch + Ours	**9.76**	43.01	**11.34**	**43.66**
SoftMatch	9.88	40.05	10.8	42.01
SoftMatch + Ours	**7.56**	**39.88**	**9.01**	**41.77**
FullySupervised	6.55	**21.06**	7.57	27.54
FullySupervised + Ours	**5.76**	21.87	**6.07**	**27.22**

Table 3: ECE and Top-1 error rates on ImageNet and WebVision (the lower the better). Better results in comparison are shown in **bold**.

Method	CIFAR-100		STL-10	
	ECE	Err	ECE	Err
SoftMatch	$26.4_{1.2}$	$37.11_{0.3}$	$13.24_{0.06}$	$20.90_{2.47}$
+ mixup all -warmup	$16.5_{3.35}$	$86.95_{1.05}$	$18.77_{2.21}$	$23.41_{2.42}$
+ mixup all +warmup	$30.63_{0.64}$	$41.5_{0.76}$	$16.66_{3.35}$	$19.70_{5.07}$
+ CM k=0 (no cosine)	$23.25_{0.32}$	$37.13_{0.35}$	$16.81_{4.01}$	$15.69_{2.78}$
+ CM k=10	$24.42_{1.04}$	$37.8_{1.3}$	$12.71_{1.02}$	$14.92_{1.03}$
+ CM k=15	$23.14_{0.04}$	$36.92_{0.02}$	$14.66_{0.71}$	$15.38_{0.85}$
+ CM k=5 (Ours)	$23.1_{0.3}$	$36.68_{0.18}$	$12.4_{0.08}$	$15.18_{0.63}$

Table 4: Ablation on CIFAR-100 and STL-10 (4 labels/class). CalibrateMix has warmup and cosine similarity. CM stands for CalibrateMix.

ilarity consistently degrades calibration and accuracy, with large increases in ECE and error rate on STL-10.

Error Analysis

Figure 2 presents the confidence, predicted labels, and ground-truth labels for six STL-10 test images under the 4-labels-per-class setting. In the top row, both CalibrateMix and SoftMatch misclassify the images. However, SoftMatch does so with extremely high confidence (over 98%), whereas CalibrateMix assigns noticeably lower confidence, reflecting appropriate uncertainty and thus better calibration. In the bottom row, SoftMatch again produces overconfident errors, while CalibrateMix makes correct predictions with high confidence, indicating better reliability. Overall, these cases show that CalibrateMix reduces overconfident mistakes while still maintaining strong confidence on correct predictions.

Conclusion

We proposed CalibrateMix, a targeted mixup strategy that improves the calibration of SSL models by utilizing training dynamics and dissimilarity-aware pairing of easy and hard samples. Specifically, CalibrateMix leverages Area Under the Margin (AUM) and Average Pseudo Margin (APM) to identify sample difficulty, and then performs mixup between labeled and pseudo-labeled samples based on their difficulty and feature dissimilarity. Our method consistently reduces ECE and generally improves accuracy (yielding lower error rates), especially in low-label settings. This enhances the reliability of AI systems in high-stakes applications while also lowering data annotation costs. CalibrateMix contributes to positive societal impact, enabling safer deployments through better-calibrated predictions. For future work, we aim to extend CalibrateMix to other tasks such as object detection, semantic segmentation, and out-of-domain (OOD) scenarios.

Acknowledgements

This research is supported in part by the NSF IIS award 2107518, a UIC Discovery Partners Institute (DPI) award, and a Google CAHSI award. Any opinions, findings, and conclusions expressed here are those of the authors and do not necessarily reflect the views of NSF, DPI, or Google CAHSI. We thank our anonymous reviewers for their constructive feedback, which helped improve the quality of our paper.

References

Bartlett, P. L.; Foster, D. J.; and Telgarsky, M. J. 2017. Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems*, 30.

Chapelle, O.; Scholkopf, B.; and Zien, A. 2009. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3): 542–542.

Chen, H.; Tao, R.; Fan, Y.; Wang, Y.; Wang, J.; Schiele, B.; Xie, X.; Raj, B.; and Savvides, M. 2023. Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning. *arXiv preprint arXiv:2301.10921*.

Coates, A.; Ng, A.; and Lee, H. 2011. An Analysis of Single-Layer Networks in Unsupervised Feature Learning. In Gordon, G.; Dunson, D.; and Dudík, M., eds., *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, 215–223. Fort Lauderdale, FL, USA: PMLR.

Cubuk, E. D.; Zoph, B.; Shlens, J.; and Le, Q. V. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 702–703.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Elsayed, G.; Krishnan, D.; Mobahi, H.; Regan, K.; and Bengio, S. 2018. Large margin deep networks for classification. *Advances in neural information processing systems*, 31.

Gan, K.; and Wei, T. 2024. Erasing the Bias: Fine-Tuning Foundation Models for Semi-Supervised Learning. *arXiv preprint arXiv:2405.11756*.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On calibration of modern neural networks. In *International conference on machine learning*, 1321–1330. PMLR.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *nature*, 521(7553): 436–444.

Lee, D.-H.; et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, 896. Atlanta.

Li, W.; Wang, L.; Li, W.; Agustsson, E.; and Van Gool, L. 2017. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*.

Lu, D.; and Weng, Q. 2007. A survey of image classification methods and techniques for improving classification performance. *International journal of Remote sensing*, 28(5): 823–870.

Mathew, A.; Amudha, P.; and Sivakumari, S. 2021. Deep learning techniques: an overview. *Advanced Machine Learning Technologies and Applications: Proceedings of AMLTA 2020*, 599–608.

Minaee, S.; Boykov, Y.; Porikli, F.; Plaza, A.; Kehtarnavaz, N.; and Terzopoulos, D. 2021. Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7): 3523–3542.

Minderer, M.; Djolonga, J.; Romijnders, R.; Hubis, F.; Zhai, X.; Houlsby, N.; Tran, D.; and Lucic, M. 2021. Revisiting the calibration of modern neural networks. *Advances in neural information processing systems*, 34: 15682–15694.

Mukhoti, J.; Kulharia, V.; Sanyal, A.; Golodetz, S.; Torr, P.; and Dokania, P. 2020. Calibrating deep neural networks using focal loss. *Advances in neural information processing systems*, 33: 15288–15299.

Müller, R.; Kornblith, S.; and Hinton, G. E. 2019. When does label smoothing help? *Advances in neural information processing systems*, 32.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A. Y.; et al. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, 4. Granada.

Nguyen, K.-B. 2024. Sequencematch: Revisiting the design of weak-strong augmentations for semi-supervised learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 96–106.

Papageorgiou, C. P.; Oren, M.; and Poggio, T. 1998. A general framework for object detection. In *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, 555–562. IEEE.

Park, S. Y.; and Caragea, C. 2022. On the calibration of pre-trained language models using mixup guided by area under the margin and saliency. *arXiv preprint arXiv:2203.07559*.

Pinto, F.; Yang, H.; Lim, S. N.; Torr, P.; and Dokania, P. 2022. Using mixup as a regularizer can surprisingly improve accuracy & out-of-distribution robustness. *Advances in Neural Information Processing Systems*, 35: 14608–14622.

Pleiss, G.; Zhang, T.; Elenberg, E.; and Weinberger, K. Q. 2020. Identifying mislabeled data using the area under the margin ranking. *Advances in Neural Information Processing Systems*, 33: 17044–17056.

Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C. A.; Cubuk, E. D.; Kurakin, A.; and Li, C.-L. 2020.

Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33: 596–608.

Sosea, T.; and Caragea, C. 2023. MarginMatch: Improving semi-supervised learning with pseudo-margins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15773–15782.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1): 1929–1958.

Thulasidasan, S.; Chennupati, G.; Bilmes, J. A.; Bhattacharya, T.; and Michalak, S. 2019. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. *Advances in neural information processing systems*, 32.

Wang, Y.; Chen, H.; Fan, Y.; SUN, W.; Tao, R.; Hou, W.; Wang, R.; Yang, L.; Zhou, Z.; Guo, L.-Z.; Qi, H.; Wu, Z.; Li, Y.-F.; Nakamura, S.; Ye, W.; Savvides, M.; Raj, B.; Shinozaki, T.; Schiele, B.; Wang, J.; Xie, X.; and Zhang, Y. 2022. USB: A Unified Semi-supervised Learning Benchmark for Classification. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 3938–3961. Curran Associates, Inc.

Xie, Q.; Luong, M.-T.; Hovy, E.; and Le, Q. V. 2020. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10687–10698.

Zagoruyko, S. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Zhang, B.; Wang, Y.; Hou, W.; Wu, H.; Wang, J.; Okumura, M.; and Shinozaki, T. 2021. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34: 18408–18419.

Zhang, H.; Cissé, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond Empirical Risk Minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.