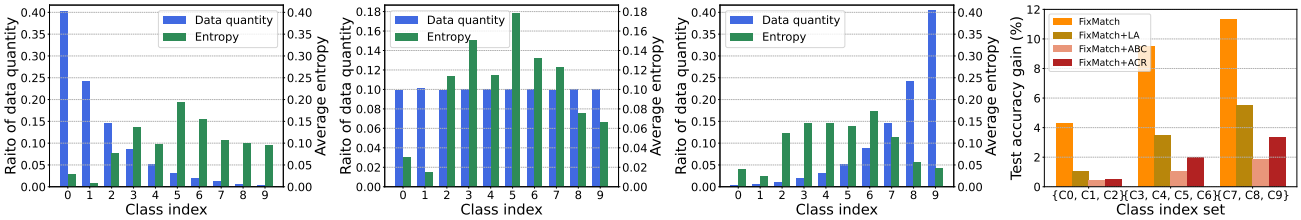
BEM: Balanced and Entropy-based Mix for Long-Tailed Semi-Supervised Learning

Hongwei Zheng[1,2], Linyuan Zhou[1], Han Li[2], Jinming Su[1], Xiaoming Wei[1], Xiaoming Xu[1] †
[1] Meituan [2] Shanghai Jiao Tong University
{zhenghongwei04, zhoulinyuan, sujinming, weixiaoming, xuxiaoming04}@meituan.com
{qingshi9974}@sjtu.edu.cn



(a) Consistent class distribution (b) Uniform class distribution (c) Reversed class distribution (d) Test accuracy gain

Figure 1. Experimental results on CIFAR10-LT [24]. **(a)-(c):** Class distribution of unlabeled data quantity and entropy for three typical settings, which have the same labeled data quantity distribution but differ in unlabeled ones. Both the data quantity and entropy are the statistical averages within one epoch after model convergence. Unexpected discrepancies are observed across all settings between the distribution of data quantity and entropy, particularly for head and tail classes. Notably, classes 3-6 exhibit the highest entropy, indicating greater uncertainty. **(d):** Test accuracy gain brought by BEM for various LTSSL frameworks in consistent setting.

Abstract

Data mixing methods play a crucial role in semi-supervised learning (SSL), but their application is unexplored in long-tailed semi-supervised learning (LTSSL). The primary reason is that the in-batch mixing manner fails to address class imbalance. Furthermore, existing LTSSL methods mainly focus on re-balancing data quantity but ignore class-wise uncertainty, which is also vital for class balance. For instance, some classes with sufficient samples might still exhibit high uncertainty due to indistinguishable features. To this end, this paper introduces the Balanced and Entropy-based Mix (BEM), a pioneering mixing approach to re-balance the class distribution of both data quantity and uncertainty. Specifically, we first propose a class balanced mix bank to store data of each class for mixing. This bank samples data based on the estimated quantity distribution, thus re-balancing data quantity. Then, we present an entropy-based learning approach to re-balance class-wise uncertainty, including entropy-based sampling strategy, entropy-based selection module, and entropy-based class balanced loss. Our BEM first leverages data mixing for improving LTSSL, and it can also serve as a complement to the existing re-balancing methods. Experimental results show that BEM significantly enhances various LTSSL frameworks and achieves state-of-the-art performances across multiple benchmarks.

1. Introduction

Semi-supervised learning (SSL) capitalizes on unlabeled data to reduce the cost of data labeling and boost the performance of models [15, 25, 32, 37, 42]. The general paradigm of most approaches is to randomly generate two views of an image with various augmentation methods and then use the output of one as the pseudo label to supervise the other [7, 38, 51]. As a simple and effective augmentation technique introduced in supervised learning [3, 17, 20, 43], data mixing is widely used in SSL algorithms [4, 5, 44], further enhancing model generalization and performance.

However, most existing SSL algorithms assume a balanced dataset, ignoring the real-world prevalence of long-tailed class distributions [2, 10, 18, 19, 53, 54]. To deal with this class imbalance scenario, various long-tailed semi-supervised learning (LTSSL) methods have been proposed, such as re-sampling [46], logit alignment [30, 47], and pseudo label alignment [26, 31]. Nevertheless, data mixing is rarely explored in LTSSL. The primary reason is that

† Corresponding Author

existing data mixing methods (*e.g.* MixUp [52], CutMix [49], and SaliencyMix [40]) often perform random mixing within a batch. Consequently, the infrequent tail classes may not be adequately sampled when the batch size is small. This hinders a balanced class distribution, which is crucial for LTSSL.

To make data mixing suitable for LTSSL, let us consider two key questions: **i) How can we apply data mixing to effectively re-balance the data quantity for each class? ii) Is it sufficient to solely focus on re-balancing data quantity to achieve class balance?** For the second question, we notice that previous LTSSL methods mainly focus on addressing the issue of long-tailed class distribution in terms of data quantity. These methods ignore the fact that class performance also depends on class-wise uncertainty [27–29, 36], which is associated with the training difficulty for each class. For instance, some classes with sufficient samples may still encounter high training difficulty due to the high uncertainty induced by indistinguishable features. As shown in Fig. 1 (a)-(c), we quantify the uncertainty by entropy [29] and compare this class-wise entropy with data quantity under three typical settings [47]. The results reveal a **significant disparity** between the class distribution of entropy and data quantity across all settings. This finding emphasizes the limitation of solely re-balancing data quantity, as it does not consider classes with high uncertainty, ultimately limiting performance improvement. Thus, it is crucial to also address the re-balancing of class-wise uncertainty, *i.e.* entropy.

To tackle the above problems, this paper presents a novel data mixing paradigm, called Balanced and Entropy-based Mix (BEM), for LTSSL. Specifically, we first introduce a simple mixing strategy, named as CamMix, which has a strong localization capability to avoid redundant areas for mixing. Then, we establish a class balanced mix bank (CBMB) to store and sample class-wise data for mixing. The sampling function follows the estimated class distribution of data quantity and we adopt the effective number [10] to represent the realistic data quantity of each class. Our CamMix incorporated CBMB can effectively re-balance the class-wise data quantity in an end-to-end optimized manner, which can not be achieved by the re-sampling methods [6, 22, 46, 56] with the complex training procedures.

Further, we present a novel entropy-based learning approach to re-balance class-wise uncertainty. Entropy-based sampling strategy (ESS) integrates class-wise entropy into the quantity-based sampling function. In addition, entropy-based selection module (ESM) adaptively determines the sampled data ratio between labeled and unlabeled data during mixing to manage the trade-off between guiding high-uncertainty unlabeled data [1, 45] with confident labeled data and maximizing the utilization of unlabeled data. Finally, we incorporate the class balanced loss [10] with class-wise entropy to form entropy-based class balanced (ECB) loss.

We highlight that our BEM is the first method that leverages data mixing to enhance LTSSL. Our results demonstrate that BEM can effectively complement existing re-balancing methods by boosting their performance across several benchmarks. As shown in Fig. 1 (d), our method enhances FixMatch [38], FixMatch+LA [30], FixMatch+ABC [26], FixMatch+ACR [47], achieving to 11.8%, 4.4%, 1.4% and 2.5% average gains on test accuracy, respectively. Additionally, BEM proves to be a versatile framework, performing well across different data distributions, diverse datasets, and various SSL learners.

2. Related Work

Data mixing. MixUp [52] and CutMix [49] are typical data mixing methods used in various computer vision tasks. While performing mixing at element-wise and region-wise levels respectively, they share a common limitation of neglecting class content, thus introducing substantial redundant context irrelevant to class content [13, 33]. To achieve class balance in LTSSL, it is essential to ensure that the selected region for mixing contains related class content and avoids redundancy. SaliencyMix [40] alleviates this issue by using a saliency map to ensure that selected regions contain class content, but the resulting region is still too coarse to avoid numerous redundant areas. In our paper, CamMix achieves tighter localization of class regions to minimize redundant areas, which is particularly well-suited for LTSSL.

Data mixing in semi-supervised learning. Data mixing is crucial in SSL, enhancing model performance by creating diverse training samples. For instance, MixMatch [5] utilizes MixUp [52] as the data mixing technique to learn a robust model. ReMixMatch [4] adds distribution alignment and augmentation anchoring to the MixMatch framework. ICT [44] employs the mean teacher model and implements MixUp on unsupervised samples. Despite widely used in SSL algorithms, almost no methods in LTSSL apply data mixing. This is mainly due to the limitation of employing in-batch mixing, which fails to address the class imbalance problem. Our method stands out as the first to incorporate data mixing in LTSSL.

Long-tailed semi-supervised learning. LTSSL is gaining attention due to its real-world applicability. For example, CReST [46] refines the model by iteratively enriching the labeled set with high-quality pseudo labels in multiple rounds. ABC [26] uses an auxiliary balanced classifier, trained by down-sampling majority classes. DASO [31] mitigates class bias by adaptively blending linear and semantic pseudo labels. ACR [47], the current state-of-the-art method, proposes a dual-branch network and dynamic logit adjustment. However, none of these methods utilizes data mixing to further enhance their performance as in SSL. CoSSL [14] uses MixUp at the feature level for minority classes and decouples representation learning and classifier learning. However,

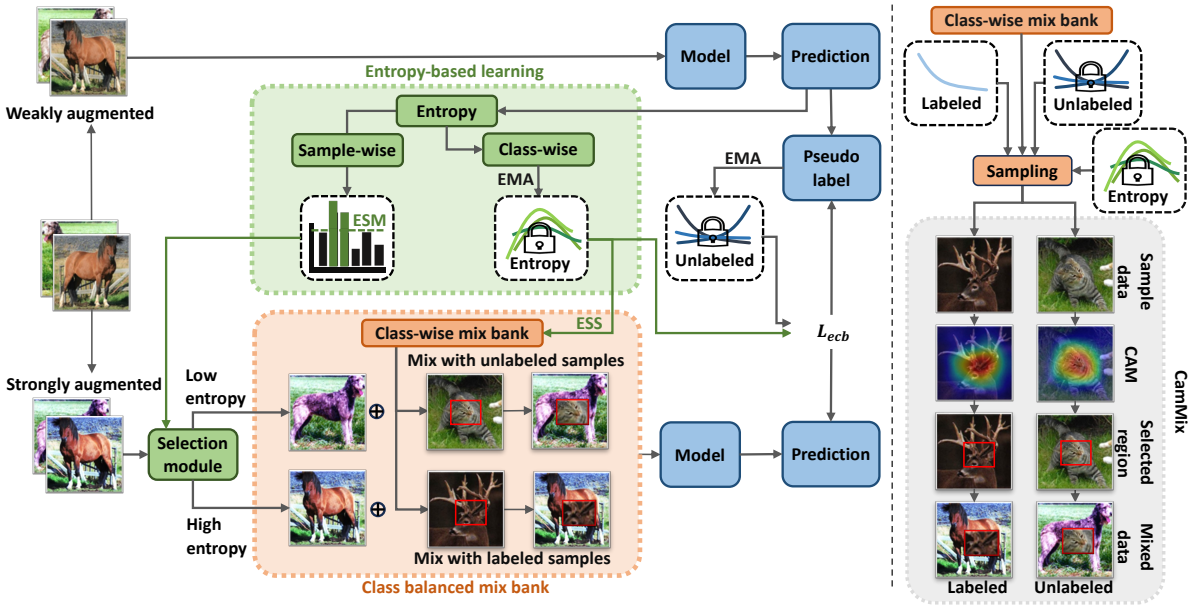


Figure 2. **Left**: The overview of Balanced and Entropy-based Mixing (BEM), incorporating with FixMatch [38] as an example in this figure. BEM consists of two sub-modules: class balanced mix bank (CBMB) and entropy-based learning (EL). CBMB re-balances data quantity through the proposed CamMix, guided by a class-balanced sampling function. EL further re-balances class-wise uncertainty using three techniques: entropy-based sampling strategy (ESS), entropy-based selection module (ESM) and entropy-based class balanced loss (L_{ecb}). **Right**: The sampling and CamMix process of BEM. The sampling process considers both the class distribution of data quantity and uncertainty, which are estimated on the fly. CamMix extracts the bounding box from the high response area of the CAM to form mixed data. (The lock icon denotes the unknown distribution that needs estimation, and the $\oplus$ icon denotes the process of CamMix.).

this feature mixing method assumes identical class distributions for labeled and unlabeled data and requires a complex training approach. Our study considers non-ideal class distributions and designs a simple image-level mixing method in an end-to-end training framework.

3. Preliminaries

Semi-supervised learning. In SSL, the training data consists of labeled data $X = \{(x_n, y_n)\}_{n=1}^{N}$ and unlabeled data $U = \{u_m\}_{m=1}^{M}$. Here, x_n and u_m are training samples, y_n is the ground truth, N and M denote the quantity of labeled data and unlabeled data, respectively. A representative framework of SSL is FixMatch [38], which utilizes unlabeled data with the *Weak and Strong Augmentation*. For an unlabeled sample u_m, it first takes a *weakly*-augmented version of u_m as the input of the model $f(\cdot)$ to compute the prediction. Then, it uses $q_m = \text{argmax}(f(A_w(u_m)))$ as one-hot pseudo label, while applying the prediction from a *strongly*-augmented of u_m to calculate the cross entropy loss L_u:

$$L_u = \sum_{m=1}^{B} \mathbb{I}(\max(f(A_w(u_m))) > \tau) \underbrace{\mathcal{H}(f(A_s(u_m)), q_m)}_{L_{cls}}, \tag{1}$$

where B denotes the batch size, $\mathcal{H}(\cdot)$ is cross entropy and L_{cls} denotes original classification loss term. $\mathbb{I}(\max(q_m) > \tau)$ is the mask to filter low-confidence pseudo label with a threshold of τ, abbreviated as $M_u(\cdot)$ in the following part. A_w and A_s denotes the *weak augmentation* (*e.g.*, random crop and flip) and *strong augmentation* (*e.g.*, RandAugment [9] and Cutout [12]), respectively.

Long-tailed semi-supervised learning. In LTSSL, a dataset with a long-tailed distribution is characterized by the minority of classes possessing a large number of samples, while the majority of classes contain only a few samples. Given C classes across the dataset, N_c represents the quantity of labeled data for class c. Without loss of generality, we assume that $N_1 \geq N_2 \geq \cdots \geq N_C$ and the imbalanced ratio is denoted by $\gamma_l = N_1/N_C$. Similarly, we can denote the quantity of unlabeled data as M_c for class c and the imbalanced ratio as $\gamma_u = \max_c M_c / \min_c M_c$.

4. Balanced and Entropy-based Mix (BEM)

The Balanced and Entropy-based Mix (BEM) is a plug-and-play method based on the existing SSL framework. Fig. 2 shows the overview of BEM, incorporating FixMatch [38] as an example. Specifically, our entropy-based learning (EL) takes the prediction of the weakly augmented samples as input to perform entropy-based sampling and selection. Then, strongly augmented samples are mixed with data from class balanced mix bank (CBMB) using our CamMix. Based on the estimated distribution of data quantity and uncertainty,

we employ the entropy-based class balance (ECB) loss L_{ecb} to train the overall framework. Please refer to Appendix C for the pseudo-code of BEM.

4.1. CamMix

Most data mixing methods, such as MixUp [52] and CutMix [49], lack the localization ability for class re-balancing. Although SaliencyMix [40] has initial localization ability, it still tends to extract excessive redundant context. To this end, we propose CamMix to replace the saliency map of SaliencyMix with Class Activation Map (CAM) [55] to achieve more accurate localization. Specifically, we feed images into the prediction model (*i.e.* ResNet50 [16]) to generate the CAM, where the last layer of the third block of ResNet50 is used as the CAM layer. The resulting CAM is used to extract the largest connected region using a threshold of τ_c. Finally, we obtain the bounding box of this region and paste the corresponding patch onto the original image. The pseudo-code of CamMix can be found in the Appendix C.

4.2. Class Balanced Mix Bank (CBMB)

Previous in-batch data mixing methods used in SSL are limited to increasing the data quantity of tail classes, thus failing to re-balance the class distribution. To address this issue, we further propose a class balanced mix bank (CBMB) that stores samples for each class and adequately selects samples to be mixed based on a prior-based class-balancing rule. In essence, the more frequent a class, the more samples are used in the data mixing process. As noted in [10], there is overlap in the data, necessitating the use of the effective number E_c to measure the realistic class distribution of data quantity:

$$E_c = \frac{1-\beta^{N_c}}{1-\beta}, \tag{2}$$

where N_c represents the data quantity of class c, while the hyper-parameter β is set to 0.999 in our experiments.

The effective number of labeled data, denoted as E_c^x, can be obtained directly using Eq. 2. As the class distribution of unlabeled data is unknown, we estimate it using a simple yet effective approach. Specifically, at each iteration t, we obtain the class distribution of the pseudo label d_c^{ut} and update the class distribution of the entire unlabeled dataset d_c^u, using an Exponential Moving Average (EMA) approach once the training status stabilizes.

$$d_c^u \leftarrow \lambda_d d_c^u + (1-\lambda_d) d_c^{ut}, \tag{3}$$

where λ_d denotes the EMA weight. To obtain the effective number of unlabeled data for each class E_c^u, we substitute the class-wise data quantity $N_c^u = M d_c^u$ into Eq. 2, where M is the quantity of entire unlabeled dataset. Then, we obtain the effective number of total data for each class by $E_c = E_c^x + E_c^u$ and perform our CamMix using the initial sampling function as follows:

$$s_c = \frac{F_c}{\sum_{c=1}^{C} F_c}, \tag{4}$$

where $F_c = 1/E_c$ and s_c denotes the sampling probability for class c. By accurately estimating the class distribution of the dataset, we can enhance the precision of mixed data sampling. Our data mixing achieves class balance among training samples, equivalent to re-sampling during training.

4.3. Entropy-based Learning (EL)

In the previous section, we re-balance training samples to initially alleviate the long-tail distribution problem. However, class balance does not only depend on data quantity. Class-wise uncertainty, which can be quantified by entropy, is also vital for class performance as it reflects training difficulty. Thus, we propose an entropy-based learning approach to re-balance class-wise entropy, including entropy-based sampling strategy (ESS), entropy-based selection module (ESM) and entropy-based class balanced (ECB) loss.

Entropy-based Sampling Strategy. To consider class-wise uncertainty in the sampling process, we define the class-wise entropy e_c^x and e_c^u for the entire labeled and unlabeled dataset, and update them in EMA manner by using the average entropy e_c^{xt} and e_c^{ut} at each training iteration t as follows:

$$\begin{aligned} e_c^{xt} &= \frac{1}{N_c^t} \sum_{n=1}^{N_c^t} \sum_{c=1}^{C} -f_c(A_w(x_n)) \log(f_c(A_w(x_n))) \\ e_c^{ut} &= \frac{1}{M_c^t} \sum_{m=1}^{M_c^t} \sum_{c=1}^{C} -f_c(A_w(u_m)) \log(f_c(A_w(u_m))), \end{aligned} \tag{5}$$

$$\begin{aligned} e_c^x &\leftarrow \lambda_e e_c^x + (1-\lambda_e) e_c^{xt} \\ e_c^u &\leftarrow \lambda_e e_c^u + (1-\lambda_e) e_c^{ut}, \end{aligned} \tag{6}$$

where N_c^t and M_c^t represent the data quantity within one batch belonging to class c according to the ground truth and pseudo label respectively, and λ_e denotes the EMA weight. It's worth noting that we start estimating the entropy of data once the training status stabilizes. Then, we obtain the total class-wise entropy, *i.e.* $e_c = e_c^u + e_c^x$, and subsequently compute the final sampling probability $\hat{s}_c$:

$$\hat{s}_c = \delta(\alpha s_c + (1-\alpha) s_c'), \tag{7}$$

where s_c' is the normalization of e_c, denoted as $s_c' = e_c / \sum_{c=1}^{C} e_c$, the hyper-parameter α is used to balance between the effective number and entropy. The convex function $\delta(\cdot)$ is utilized to map the sampling function better according to FlexMatch [51]. Finally, we can obtain a more comprehensive sampling function $\hat{s}_c$ for CamMix.

Entropy-based Selection Module. Previous work [4, 5, 44] in SSL primarily uses unlabeled samples for data mixing. Yet, some pseudo labels possess high uncertainty [1, 45], especially for challenging samples or in early training stages, causing confirmation bias [1]. Our data mixing approach allows the selection of both labeled and unlabeled data. We suggest augmenting high-uncertainty unlabeled data with confident labeled data. However, this beneficial mixing may under-utilize unlabeled data when the regions of labeled data cover unlabeled ones, leaving them unexploited in training. This trade-off is a crucial consideration in data mixing. Thus, we utilize sample-wise entropy e_m as the selection indicator between labeled and unlabeled samples in data mixing as:

$$e_m = \sum_{c=1}^{C} -f_c(A_w(u_m)) \log(f_c(A_w(u_m))). \tag{8}$$

We then define $M_h(\cdot)$ and $M_l(\cdot)$ as the masks of high and low entropy for selecting labeled and unlabeled samples respectively. They are also used to mask the unsupervised loss as in the Appendix A. These masks can be expressed as:

$$\begin{aligned} M_h(u_m) &= \mathbb{I}(e_m > \tau_e) \\ M_l(u_m) &= \mathbb{I}(e_m < \tau_e), \end{aligned} \tag{9}$$

where τ_e is the selection threshold of the entropy mask, updated in EMA manner:

$$\tau_e \leftarrow \lambda_\tau \tau_e + (1 - \lambda_\tau) e^t, \tag{10}$$

where λ_τ denotes the EMA weight, e^t is the average entropy of unlabeled data at each training iteration t, *i.e.* $e^t = \frac{1}{B}\sum_{m=1}^{B} e_m$. In the early training stages, we select more labeled samples for mixing due to the uncertainty of model prediction on some unlabeled data. As training progresses and predictions become more reliable, the utilization of unlabeled data increases.

Entropy-based class balanced loss. We further apply the class balanced loss, which is first introduced in [10] to re-balance the class distribution by utilizing the weighted loss based on the class-wise effective number as $L_{cb} = L_{cls}/E_c$. By normalizing $1/E_c$ as Eq. 4, we can obtain $L_{cb} = s_c L_{cls}$.

Moreover, to tackle the class-wise uncertainty problem in LTSSL, we propose entropy-based class balanced loss L_{ecb} on the unlabeled data. L_{ecb} uses $\hat{s}_c$ to measure both the effective number and uncertainty as:

$$L_{ecb} = \hat{s}_c^u L_{cls}, \tag{11}$$

where $\hat{s}_c^u$ is calculated by Eq. 7, but only based on unlabeled data. Finally, L_{ecb} can be weighted towards both tail classes and high uncertainty classes, further re-balancing the training process. Unlike previous entropy-based losses [15, 35], our loss focuses on class-wise uncertainty instead of sample-wise, making it ideally suited for the LTSSL problem characterized by large category gaps. A detailed description of the loss functions can be found in the Appendix A.

5. Experiments

5.1. Experimental setup

Datasets. We perform evaluation experiments of our proposed method on widely used long-tailed datasets, including CIFAR10-LT [24], CIFAR100-LT [24], STL10-LT [8] and ImageNet-127 [14]. To create imbalanced versions of the datasets, we randomly discard training samples to maintain the pre-defined imbalance ratio. With the imbalance ratio γ_l and maximum number N_1 of labeled samples, we can calculate the number of labeled samples for class c as $N_c = N_1 \times \gamma_l^{-\frac{c-1}{C-1}}$. Similarly, using the parameters γ_u and M_1, we can determine the class distribution of unlabeled data quantity as in the labeled samples. For a detailed introduction to the datasets, please refer to the Appendix B.

Implementation Details. Following DASO [31], we apply our method to various baseline frameworks, including FixMatch [38], FixMatch + LA [30], FixMatch + ABC [26] and FixMatch + ACR [47]. We compare our method with recent re-balancing methods like DARP [23], CReST/CReST+ [46] and DASO [31]. For a fair comparison, our code is developed based on DASO and ACR, implemented with Pytorch [34]. We conduct our experiments on CIFAR10-LT, CIFAR100-LT and STL-10 using Wide ResNet-28-2 [50], and on ImageNet-127 using ResNet-50 [16]. The top-1 accuracy on the test set is used as the evaluation metric. The mean and standard deviation of three independent runs are reported. Due to the page limitation, detailed training settings are provided in the Appendix B.

5.2. Results on CIFAR10/100-LT and STL10-LT.

We first consider the $\gamma_l = \gamma_u$ situation which is the most common scenario in SSL. Then, we investigate the performance of the methods by setting $\gamma_l \neq \gamma_u$, including uniform ($\gamma_u = 1$) and reversed ($\gamma_u = 1/100$) scenarios.

In case of $\gamma_l = \gamma_u$. As shown in Tab. 1, we compare our method with existing re-balancing methods under various baseline settings. When setting FixMatch as the baseline, our BEM shows superior performance improvement in most scenarios. When further adding LA to FixMatch for label re-balancing, our BEM outperforms all other configurations. When integrating ABC into FixMatch for pseudo label re-balancing, our BEM can benefit the baseline more than the DASO. Finally, we also demonstrate that our methods can complement ACR, achieving the SOTA performance with an average gain of 18.35% over FixMatch for CIFAR10-LT. In summary, our BEM achieves consistent and significant gain under all baseline settings, showing its great adaptability. The main reason is that, unlike most previous methods with pseudo label or logit adjustment, we directly re-balance the class distribution through data mixing, a vital technique missing in them, thus complementing these methods.

In case of $\gamma_l \neq \gamma_u$. In real-world datasets, the class dis-

Table 1. Comparison of test accuracy with combinations of different baseline frameworks under $\gamma_l = \gamma_u$ setup on CIFAR10-LT and CIFAR100-LT. The best results for each diversion are in **bold**.

	CIFAR10-LT				CIFAR100-LT			
	$\gamma = \gamma_l = \gamma_u = 100$		$\gamma = \gamma_l = \gamma_u = 150$		$\gamma = \gamma_l = \gamma_u = 10$		$\gamma = \gamma_l = \gamma_u = 20$	
Algorithm	$N_1 = 500$ $M_1 = 4000$	$N_1 = 1500$ $M_1 = 3000$	$N_1 = 500$ $M_1 = 4000$	$N_1 = 1500$ $M_1 = 3000$	$N_1 = 50$ $M_1 = 400$	$N_1 = 150$ $M_1 = 300$	$N_1 = 50$ $M_1 = 400$	$N_1 = 150$ $M_1 = 300$
Supervised	47.3±0.95	61.9±0.41	44.2±0.33	58.2±0.29	29.6±0.57	46.9±0.22	25.1±1.14	41.2±0.15
w/LA [30]	53.3±0.44	70.6±0.21	49.5±0.40	67.1±0.78	30.2±0.44	48.7±0.89	26.5±1.31	44.1±0.42
FixMatch [38]	67.8±1.13	77.5±1.32	62.9±0.36	72.4±1.03	45.2±0.55	56.5±0.06	40.0±0.96	50.7±0.25
w/DARP [23]	74.5±0.78	77.8±0.63	67.2±0.32	73.6±0.73	49.4±0.20	58.1±0.44	43.4±0.87	52.2±0.66
w/CReST+ [46]	**76.3**±0.86	78.1±0.42	67.5±0.45	73.7±0.34	44.5±0.94	57.4±0.18	40.1±1.28	52.1±0.21
w/DASO [31]	76.0±0.37	79.1±0.75	**70.1**±1.81	75.1±0.77	49.8±0.24	**59.2**±0.35	43.6±0.09	52.9±0.42
w/BEM (ours)	75.8±1.13	**80.3**±0.62	69.7±0.91	**75.7**±0.22	**50.4**±0.34	59.0±0.23	**44.1**±0.18	**54.3**±0.36
FixMatch+LA [30]	75.3±2.45	82.0±0.36	67.0±2.49	78.0±0.91	47.3±0.42	58.6±0.36	41.4±0.93	53.4±0.32
w/DARP [23]	76.6±0.92	80.8±0.62	68.3±0.94	76.7±1.13	50.5±0.78	59.9±0.32	44.4±0.65	53.8±0.43
w/CReST [46]	76.7±1.13	81.1±0.57	70.9±1.18	77.9±0.71	44.0±0.21	57.1±0.55	40.6±0.55	52.3±0.20
w/DASO [31]	77.9±0.88	82.5±0.08	70.1±1.68	79.0±2.23	50.7±0.51	60.6±0.71	44.1±0.61	55.1±0.72
w/BEM (ours)	**78.6**±0.97	**83.1**±0.13	**72.5**±1.13	**79.9**±1.02	**51.3**±0.26	**61.9**±0.57	**44.8**±0.21	**56.1**±0.54
FixMatch+ABC [26]	78.9±0.82	83.8±0.36	66.5±0.78	80.1±0.45	47.5±0.18	59.1±0.21	41.6±0.83	53.7±0.55
w/DASO [31]	**80.1**±1.16	83.4±0.31	70.6±0.80	80.4±0.56	**50.2**±0.62	60.0±0.32	**44.5**±0.25	55.3±0.53
w/BEM (ours)	79.8±0.82	**83.9**±0.34	**70.7**±0.78	**80.8**±0.67	50.0±0.15	**60.9**±0.42	44.4±0.18	**55.5**±0.84
FixMatch+ACR [47]	81.6±0.19	84.1±0.39	77.0±1.19	80.9±0.22	55.7±0.12	65.6±0.16	48.0±0.75	58.9±0.36
w/BEM (ours)	**83.5**±0.33	**85.5**±0.28	**78.1**±0.99	**83.8**±1.12	**55.8**±0.32	**66.3**±0.24	**48.6**±0.45	**59.8**±0.37

Table 2. Comparison of test accuracy with combinations of different baseline frameworks under $\gamma_l \neq \gamma_u$ setup on CIFAR10-LT and STL10-LT. The γ_l is fixed to 100 for CIFAR10-LT, and the γ_l is set to 10 and 20 for STL10-LT. The N/A denotes the class distribution of data quantity is unknown. The best results for each diversion are in **bold**.

	CIFAR10-LT($\gamma_l \neq \gamma_u$)				STL10-LT($\gamma_u = N/A$)			
	$\gamma_u = 1$(uniform)		$\gamma_u = 1/100$(reversed)		$\gamma_l = 10$		$\gamma_l = 20$	
Algorithm	$N_1 = 500$ $M_1 = 4000$	$N_1 = 1500$ $M_1 = 3000$	$N_1 = 500$ $M_1 = 4000$	$N_1 = 1500$ $M_1 = 3000$	$N_1 = 150$ $M = 100k$	$N_1 = 450$ $M = 100k$	$N_1 = 150$ $M = 100k$	$N_1 = 450$ $M = 100k$
FixMatch [38]	73.0±3.81	81.5±1.15	62.5±0.94	71.8±1.70	56.1±2.32	72.4±0.71	47.6±4.87	64.0±2.27
w/DARP [23]	82.5±0.75	84.6±0.34	70.1±0.22	80.0±0.93	66.9±1.66	75.6±0.45	59.9±2.17	72.3±0.60
w/CReST [46]	83.2±1.67	87.1±0.28	70.7±2.02	**80.8**±0.39	61.7±2.51	71.6±1.17	57.1±3.67	68.6±0.88
w/CReST+ [46]	82.2±1.53	86.4±0.42	62.9±1.39	72.9±2.00	61.2±1.27	71.5±0.96	56.0±3.19	68.5±1.88
w/DASO [31]	86.6±0.84	88.8±0.59	**71.0**±0.95	80.3±0.65	**70.0**±1.19	78.4±0.80	**65.7**±1.78	75.3±0.44
w/BEM(ours)	**86.8**±0.47	**89.1**±0.75	70.0±1.72	79.1±0.77	68.3±1.15	**81.2**±1.42	61.6±0.98	**76.0**±1.51
FixMatch+ACR [47]	92.1±0.18	93.5±0.11	85.0±0.09	89.5±0.17	77.1±0.24	83.0±0.32	75.1±0.70	81.5±0.25
w/BEM(ours)	**94.3**±0.14	**95.1**±0.56	**85.5**±0.21	**89.8**±0.12	**79.3**±0.34	**84.2**±0.56	**75.9**±0.15	**82.3**±0.23

tribution of unlabeled data remains unknown or inconsistent with labeled data. For CIFAR10-LT, we consider two extreme scenarios: uniform and reversed. For STL10-LT, where the class distribution of unlabeled data is unknown, we set $\gamma_l \in \{10, 20\}$ and $N_1 \in \{150, 450\}$.

As shown in Tab. 2, our methods yield an average improvement of 14.1% and 11.1% over FixMatch in two scenarios for CIFAR10-LT. However, our method is less effective than DASO under the reversed setting. We speculate that data mixing methods cannot achieve thorough re-balancing in challenging scenarios, unlike approaches from the prediction perspective. However, integrating ACR results in the best performance on CIFAR10-LT, even under the reversed setting, with an average gain of 22.9% and 30.9% over FixMatch. Similarly, for STL-10, our method enhances the performance of FixMatch and achieves the best performance when combined with ACR. This highlights the value of our BEM for re-balancing methods. Further comparisons of our method with more re-balancing methods can be found in the Appendix D.

Table 3. Comparison of test accuracy with combinations of different SSL learners, including MeanTeacher, FlexMatch and SoftMatch.

	C10-LT		C100-LT	STL10-LT
	$N_1 = 1500$ $M_1 = 3000$		$N_1 = 150$ $M_1 = 300$	$N_1 = 450$ $M = 100k$
Algorithm	$\gamma_u = 100$	$\gamma_u = 1$	$\gamma_u = 10$	$\gamma_u = N/A$
MeanTeacher[39]	68.6±0.88	46.4±0.98	52.1±0.09	54.6±0.17
w/BEM(Ours)	**73.5**±0.56	**81.3**±1.67	**60.1**±0.43	**75.3**±0.59
FlexMatch [51]	79.2±0.92	82.2±0.23	62.1±0.86	74.9±0.42
w/BEM(Ours)	**81.2**±0.50	**88.0**±0.17	**68.4**±0.79	**81.2**±0.92
SoftMatch [7]	79.6±0.46	78.3±0.86	62.8±0.33	75.5±0.74
w/BEM(Ours)	**82.0**±0.38	**84.5**±0.25	**68.9**±1.08	**82.8**±0.49

BEM on the SSL learner. We further validate the adaptability of BEM with various SSL learners, including MeanTeacher [39], FlexMatch [51] and SoftMatch [7]. Notably, FlexMatch and SoftMatch outperform FixMatch on balanced datasets. For SoftMatch, we only apply L_{cb} considering its training process already re-weights the loss based on class-wise confidence. Following DASO, we set $\gamma_l = 100$ for CIFAR10-LT and $\gamma_l = 10$ for CIFAR100-LT and STL10-LT. As depicted in Tab. 3, our method enhances the performance of all SSL learners under each setting. Specially, MeanTeacher initially underperforms on the Long-Tailed dataset but achieves gains of 41.1%, 15.4%, and 37.9% on three datasets by applying BEM. SoftMatch, the state-of-the-art SSL method, also gains an additional 5.5%, 9.7% and 9.7% improvement with our BEM.

5.3. Results on ImageNet-127.

ImageNet127, initially introduced in [21] and later employed by CReST [46] for imbalanced SSL, is a naturally imbalanced dataset with an imbalance ratio $\gamma \approx 286$. It groups the 1000 classes of ImageNet [11] into 127 classes, based on the WordNet hierarchy. Due to resource constraints, we down-sample the origin ImageNet127 images to 32×32 or 64×64 pixel images [14] and randomly select 10% of training samples as the labeled set. Given the long-tailed test set, we set $\alpha = 0.2$ to reduce sampling and loss weight bias towards tail classes, favoring high uncertainty classes instead. Tab. 4 demonstrates the superiority of our method over FixMatch, even without other re-balancing techniques. When combined with ACR, our method achieves the best results for both image sizes (95.3% and 51.1% absolute gains over FixMatch). This shows the applicability of our BEM to long-tailed test datasets and its ability to enhance previous re-balancing methods.

5.4. Comprehensive analysis of the method.

We perform comprehensive ablation studies to further understand how our method enhances baseline frameworks. Following DASO, we use CIFAR10-LT (C10) with $N_1 = 500$, $\gamma = 100$ and STL10-LT (STL10) with $N_1 = 150$, $\gamma_l = 10$ to cover both $\gamma_l = \gamma_u$ and $\gamma_l \neq \gamma_u$ cases. Our baseline framework is FixMatch. More results are provided in the Appendix D.

Table 4. Comparison of test accuracy with combinations of different baseline frameworks on ImageNet-127.

Algorithm	32×32	64×64
FixMatch [38]	29.7	42.3
w/DARP [23]	30.5	42.5
w/DARP+cRT [23]	39.7	51.0
w/CReST+ [46]	32.5	44.7
w/CReST++LA [30]	40.9	55.9
w/CoSSL [14]	43.7	53.9
w/TRAS [48]	46.2	54.1
w/BEM(Ours)	53.3	58.2
w/ACR [47]	57.2	63.6
w/ACR+BEM(Ours)	**58.0**	**63.9**

Table 5. Ablation study on different mixing strategies. Apart from BEM, all other methods perform mixing within the same batch.

Algorithm	C10	STL10
FixMatch [38]	67.8	56.1
w/MixUp [52]	69.9	63.2
w/CutMix [49]	70.2	62.5
w/SaliencyMix [40]	70.8	64.0
w/CamMix(Ours)	71.9	64.8
w/BEM(Ours)	**75.7**	**68.3**

Ablation study on different mixing strategies. We compare our mixing method with existing techniques including MixUp [52], CutMix [49] and SaliencyMix [40] to demonstrate its effectiveness in Tab. 5. First, we mix data within the same batch. SaliencyMix outperforms CutMix and MixUp on both datasets, and our CamMix surpasses SaliencyMix, indicating better localization ability. By further optimizing the in-batch mixing method, BEM achieves the best results.

Ablation study on each component of BEM. We verify each component in BEM by either removal or standard component replacement in Tab. 6. The accuracy on both datasets reduces sharply when replacing CamMix with CutMix. It highlights the importance of semantic region selection. We then remove CBMB and implement random sampling, resulting in a maximum performance decrease of 5.1% and 4.9%, respectively. This suggests our CBMB effectively tackles the long-tail problem. Removing ESS, denoted as setting $\alpha = 1$, also leads to a decline in the model's performance. When we remove ESM and merely use unlabeled data mixing, it results in a 4.4% performance decrease on STL10. This implies initial training phase guidance from confident labeled data resolves the problem of pseudo label errors especially when $\gamma_l \neq \gamma_u$. Finally, the removal of the ECB loss also

Table 6. Ablation study on each component of BEM.

	C10	STL10
BEM(Ours)	**75.7**	**68.3**
w/o CamMix	74.0	66.6
w/o CBMB	72.1	65.0
w/o ESS	74.7	67.0
w/o ESM	75.3	65.3
w/o ECB Loss	74.9	67.2

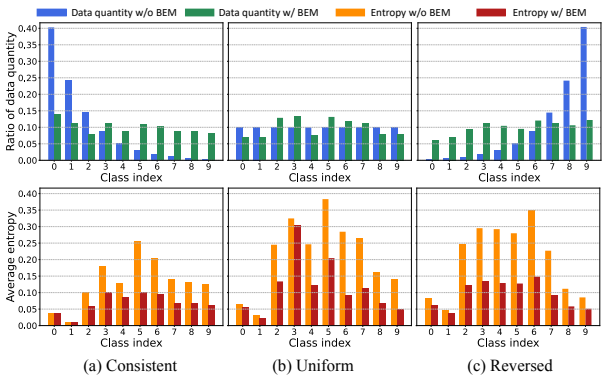


Figure 3. Class distribution of data quantity and entropy in three settings. Each mixed data is calculated as containing two classes.

causes a performance drop on both datasets.

Furthermore, we conduct a qualitative analysis of the performance enhancement achieved by BEM on CIFAR10-LT, setting $\gamma_l = \gamma_u = 100$, $N_1 = 500$ and $M_1 = 4000$. More visualization analysis can be seen in the Appendix E.

Visualization of the class distribution of unlabeled data quantity and entropy. To verify the effect of BEM on the re-balancing training process, we visualize the class distribution of data quantity and entropy. Fig. 3 reveals our method's effect on re-balancing data quantity across all settings. Moreover, our approach notably diminishes uncertainty via entropy and re-balances class-wise entropy, particularly in uniform settings where higher entropy classes engage more training samples, thus lowering uncertainty.

Visualization of T-SNE. Additionally, we visualize the learning representation on the balanced test set using t-distributed stochastic neighbor embedding (t-SNE) [41]. We apply our method to FixMatch and ACR respectively. The results in Fig. 4 suggest that our method generates clearer classification boundaries for representations.

Visualization of data mixing. As shown in Fig. 5, we compare intermediate images from various data mixing methods on STL10 due to the high-resolution input. Five images with different target sizes are selected to visualize. CutMix shows strong randomness and tends to miss the class content, especially when the target is small (see in (e)). Although SaliencyMix has initial target localization ability, it often fails to accurately locate key areas and tends to include numerous

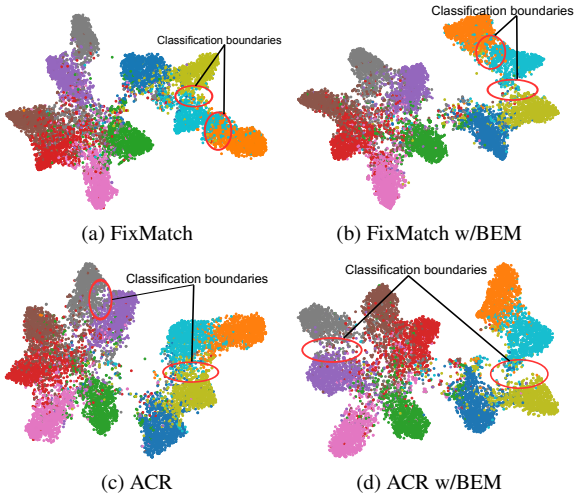


Figure 4. Comparison of t-SNE visualization with combinations of FixMatch and ACR.

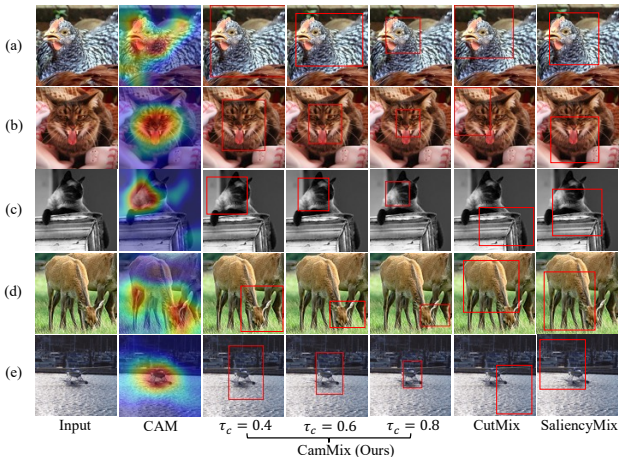


Figure 5. The visualization of data mixing process for CutMix, SaliencyMix, and CamMix on STL10-LT. The red box indicates the image area selected by data mixing.

redundant contexts (see in (c) and (e)). CamMix shows the best localization ability, accurately locating the class content based on CAM. As τ_c increases, localization accuracy improves and inclusion of redundant context decreases.

6. Conclusion

In this work, we introduce a novel approach, Balanced and Entropy-based Mix (BEM), to enhance long-tailed semi-supervised learning by re-balancing the training process. Specially, we re-balance data quantity using the class balanced mix bank and re-balance class-wise uncertainty through the entropy-based learning approach. As the first method to leverage data mixing in LTSSL, BEM significantly boosts the accuracy of various LTSSL frameworks across multiple benchmarks, offering a complementary technique for other re-balancing methods.

References

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020. 2, 5

[2] Samy Bengio. Sharing representations for long tail computer vision problems. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 1–1, 2015. 1

[3] David Berthelot, Colin Raffel, Aurko Roy, and Ian Goodfellow. Understanding and improving interpolation in autoencoders via an adversarial regularizer. *arXiv preprint arXiv:1807.07543*, 2018. 1

[4] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019. 1, 2, 5

[5] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019. 1, 2, 5

[6] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019. 2

[7] Hao Chen, Ran Tao, Yue Fan, Yidong Wang, Jindong Wang, Bernt Schiele, Xing Xie, Bhiksha Raj, and Marios Savvides. Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning. *arXiv preprint arXiv:2301.10921*, 2023. 1, 7

[8] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 5

[9] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 3

[10] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019. 1, 2, 4, 5

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 7

[12] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 3

[13] Fei Du, Peng Yang, Qi Jia, Fengtao Nan, Xiaoting Chen, and Yun Yang. Global and local mixture consistency cumulative learning for long-tailed visual recognitions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15814–15823, 2023. 2

[14] Yue Fan, Dengxin Dai, Anna Kukleva, and Bernt Schiele. Cossl: Co-learning of representation and classifier for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14574–14584, 2022. 2, 5, 7

[15] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17, 2004. 1, 5

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4, 5

[17] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 558–567, 2019. 1

[18] Feng Hong, Jiangchao Yao, Zhihan Zhou, Ya Zhang, and Yanfeng Wang. Long-tailed partial label learning via dynamic rebalancing. In *The Eleventh International Conference on Learning Representations*, 2023. 1

[19] Feng Hong, Jiangchao Yao, Yueming Lyu, Zhihan Zhou, Ivor Tsang, Ya Zhang, and Yanfeng Wang. On harmonizing implicit subpopulations. In *The Twelfth International Conference on Learning Representations*, 2024. 1

[20] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*, 2017. 1

[21] Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016. 7

[22] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019. 2

[23] Jaehyung Kim, Youngbum Hur, Sejun Park, Eunho Yang, Sung Ju Hwang, and Jinwoo Shin. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. *Advances in neural information processing systems*, 33: 14567–14579, 2020. 5, 6, 7

[24] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1, 5

[25] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016. 1

[26] Hyuck Lee, Seungjae Shin, and Heeyoung Kim. Abc: Auxiliary balanced classifier for class-imbalanced semi-supervised learning. *Advances in Neural Information Processing Systems*, 34:7082–7094, 2021. 1, 2, 5, 6

[27] Bolian Li, Zongbo Han, Haining Li, Huazhu Fu, and Changqing Zhang. Trustworthy long-tailed classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6970–6979, 2022. 2

[28] Han Li, Bowen Shi, Wenrui Dai, Hongwei Zheng, Botao Wang, Yu Sun, Min Guo, Chenglin Li, Junni Zou, and Hongkai Xiong. Pose-oriented transformer with uncertainty-guided refinement for 2d-to-3d human pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1296–1304, 2023.

[29] Jun Li, Zichang Tan, Jun Wan, Zhen Lei, and Guodong Guo. Nested collaborative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6949–6958, 2022. 2

[30] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020. 1, 2, 5, 6, 7

[31] Youngtaek Oh, Dong-Jin Kim, and In So Kweon. Daso: Distribution-aware semantics-oriented pseudo-label for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9786–9796, 2022. 1, 2, 5, 6

[32] Yassine Ouali, Céline Hudelot, and Myriam Tami. An overview of deep semi-supervised learning. *arXiv preprint arXiv:2006.05278*, 2020. 1

[33] Seulki Park, Youngkyu Hong, Byeongho Heo, Sangdoo Yun, and Jin Young Choi. The majority can help the minority: Context-rich minority oversampling for long-tailed classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6887–6896, 2022. 2

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 5

[35] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8050–8058, 2019. 5

[36] Dvir Samuel and Gal Chechik. Distributional robustness loss for long-tail learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9495–9504, 2021. 2

[37] Heereen Shim, Stijn Luca, Dietwig Lowet, and Bart Vanrumste. Data augmentation and semi-supervised learning for deep neural networks-based text classifier. In *Proceedings of the 35th annual ACM symposium on applied computing*, pages 1119–1126, 2020. 1

[38] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020. 1, 2, 3, 5, 6, 7

[39] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017. 7

[40] AFM Uddin, Mst Monira, Wheemyung Shin, TaeChoong Chung, Sung-Ho Bae, et al. Saliencymix: A saliency guided data augmentation strategy for better regularization. *arXiv preprint arXiv:2006.01791*, 2020. 2, 4, 7

[41] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 8

[42] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine learning*, 109(2):373–440, 2020. 1

[43] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International conference on machine learning*, pages 6438–6447. PMLR, 2019. 1

[44] Vikas Verma, Kenji Kawaguchi, Alex Lamb, Juho Kannala, Arno Solin, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. *Neural Networks*, 145:90–106, 2022. 1, 2, 5

[45] Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, et al. Freematch: Self-adaptive thresholding for semi-supervised learning. *arXiv preprint arXiv:2205.07246*, 2022. 2, 5

[46] Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Yuille, and Fan Yang. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10857–10866, 2021. 1, 2, 5, 6, 7

[47] Tong Wei and Kai Gan. Towards realistic long-tailed semi-supervised learning: Consistency is all you need. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3469–3478, 2023. 1, 2, 5, 6, 7

[48] Tong Wei, Qian-Yu Liu, Jiang-Xin Shi, Wei-Wei Tu, and Lan-Zhe Guo. Transfer and share: semi-supervised learning from long-tailed data. *Machine Learning*, pages 1–18, 2022. 7

[49] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 2, 4, 7

[50] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 5

[51] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34:18408–18419, 2021. 1, 4, 7

[52] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 2, 4, 7

[53] Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *Proceedings of the IEEE/CVF*

conference on computer vision and pattern recognition, pages 2361–2370, 2021. 1

[54] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 1

[55] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 4

[56] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9719–9728, 2020. 2