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

Supplementary Material

A. Detailed Loss Functions

We detail the loss functions for the training in this section. For the labeled data, we directly adopt the cross entropy $\mathcal{H}(\cdot)$ to calculate the supervised loss L_s. For the unlabeled data, we first follow FixMatch [18] to filter samples with low-confidence pseudo label by a mask $M_u(u_m) = \mathbb{I}(\max(f(A_w(u_m))) > \tau)$. Then, we can obtain M_h and M_l, the masks of high and low entropy for selecting labeled samples (x^s_m, y^s_m) and unlabeled samples u^s_m in data mixing. Given the mixed samples u'_m from CAMmix, we can obtain four types of unsupervised loss (*i.e.* L^h_u, L^l_u, $L^h_{u^s}$, and $L^l_{u^s}$), in which L^h_u, L^l_u and $L^l_{u^s}$ are weighted by the entropy-based class balanced weight $\hat{s}^u$ to form the L_{ecb}. Specifically, L^h_u and L^l_u are supervised by the pseudo label of the original unlabeled data q_m, while $L^h_{u^s}$ and $L^l_{u^s}$ are supervised by the ground truth of the sampled labeled data y^s_m and the pseudo label of the sampled unlabeled data q^s_m, respectively. The final loss function L is weighted by λ reflecting the proportion of area occupied by original and sampled data as in CutMix [23]. Detailed loss functions are as follows:

$$
\begin{aligned}
L_s &= \sum_{n=1}^{B} \mathcal{H}(f(A_w(x_n)), y_n) \\
L^h_u &= \hat{s}^u \sum_{m=1}^{B} M_u(u_m) M_h(u_m) \mathcal{H}(f(u'_m), q_m) \\
L^l_u &= \hat{s}^u \sum_{m=1}^{B} M_u(u_m) M_l(u_m) \mathcal{H}(f(u'_m), q_m) \\
L^h_{u^s} &= \sum_{m=1}^{B} M_h(u_m) \mathcal{H}(f(u'_m), y^s_m) \\
L^l_{u^s} &= \hat{s}^u \sum_{m=1}^{B} M_u(u^s_m) M_l(u_m) \mathcal{H}(f(u'_m), q^s_m) \\
L &= L_s + \lambda(L^h_u + L^l_u) + (1-\lambda)(L^h_{u^s} + L^l_{u^s})
\end{aligned}
\tag{1}
$$

Algorithm 1 Balanced and Entropy-based Mix (BEM)

Input: Labeled dataset X, unlabeled dataset U, model f, effective number of labeled data E^x_c, CAM threshold τ_c, area threshold τ_a, balanced parameter α, number of iterations T.
Require: Weak augmentation A_w, strong augmentation A_s.

for $t = 1$ to T **do**
  $\{(x_n, y_n)\}^B_{n=1} \leftarrow X$, $\{u_m\}^B_{m=1} \leftarrow U$
  Pseudo label $q_m \leftarrow$ `argmax` $f(A_w(u_m))$
  {Update training status}
  Update CBMB according to (x_n, y_n), (u_m, q_m)
  Update class-wise data quantity E_c via Eq. 2, 3
  Update class-wise entropy e_c via Eq. 5, 6
  Update sampling probability $\hat{s}$, $\hat{s}^u$ via Eq. 7
  Update sample-wise entropy e_m, entropy masks M_h, M_l and entropy selection threshold τ_e via Eq. 8, 9, 10
  {Sampling}
  $\{(x^s_m, y^s_m)\}^B_{m=1}, \{u^s_m\}^B_{m=1} \leftarrow$ Sample labeled and unlabeled data from CBMB following $\hat{s}$
  {Selection and CamMix}
  $\{u'_m\}^B_{m=1}$, $\lambda \leftarrow$ `CamMix`$(A_s(u_m),\ A_w(x^s_m),\ y^s_m,\ A_w(u^s_m), f)$ get mixed data and loss weight following M_h, M_l, τ_c and τ_a
  {Compute losses}
  Generate the mask of pseudo label M_u
  $L_s \leftarrow \sum_{n=1}^{B} \mathcal{H}(f(A_w(x_n)), y_n)$
  $L^h_u \leftarrow \hat{s}^u \sum_{m=1}^{B} M_u(u_m) M_h(u_m) \mathcal{H}(f(u'_m), q_m)$
  $L^l_u \leftarrow \hat{s}^u \sum_{m=1}^{B} M_u(u_m) M_l(u_m) \mathcal{H}(f(u'_m), q_m)$
  $L^h_{u^s} \leftarrow \sum_{m=1}^{B} M_h(u_m) \mathcal{H}(f(u'_m), y^s_m)$
  $L^l_{u^s} \leftarrow \hat{s}^u \sum_{m=1}^{B} M_u(u^s_m) M_l(u_m) \mathcal{H}(f(u'_m), q^s_m)$
  $L = L_s + \lambda(L^h_u + L^l_u) + (1-\lambda)(L^h_{u^s} + L^l_{u^s})$
  Update f based on ∇L using SGD
end for
return

B. Detailed Experimental Setup

In this section, we provide additional information about the datasets and implementation details.

Datasets. We evaluate our method in three scenarios, i.e., *1)* the class distribution of labeled data is consistent with the unlabeled data ($\gamma_l = \gamma_u$). *2)* the labeled and unlabeled data fail to share the same distribution ($\gamma_l \neq \gamma_u$). *3)* The test data possesses an imbalanced class distribution.

- **CIFAR10/100-LT** CIFAR-10/100 [11] are originally class-balanced datasets, each containing 500/5000 samples across 10 and 100 classes respectively. All images are 32×32 in size. Following previous work [16], we sample the training data to create imbalanced versions of the datasets. We employ different sampling ratios for labeled and unlabeled data to achieve various data distributions, including $\gamma_l = \gamma_u$ and $\gamma_l \neq \gamma_u$ scenarios. The test set contains $10k$ samples with a balanced class distribution. The CIFAR dataset can be downloaded from https://www.cs.toronto.edu/ kriz/cifar.html.
- **STL10-LT** The STL-10 [4] dataset consists of 5000 class-

Algorithm 2 CamMix

Input: Strong augmentation of unlabeled data $A_s(u_m)$, weak augmentation of sampled labeled data $A_w(x_m^s)$, the label of sampled labeled data y_m^s, weak augmentation of sampled unlabeled data $A_w(u_m^s)$, model f, high entropy mask M_h, low entropy mask M_l, CAM threshold τ_c, area threshold τ_a, functions in skimage `label`$(\cdot)$ and `regionprops`$(\cdot)$, the function of CutMix `Mix`$(\cdot)$.

Output: Mixed data $\{u'_m\}_{m=1}^{B}$, loss weight λ.

1: **for** $m = 1$ to B **do**
2: $\quad q_m^s \leftarrow \texttt{argmax}\, f(A_w(u_m^s))$
3: $\quad CAM_m^u \leftarrow \texttt{GradCAM}(A_w(u_m^s), q_m^s)$
4: $\quad S_m^u \leftarrow \texttt{int}(CAM_m^u > \tau_c)$
5: $\quad P_m^u \leftarrow \texttt{max}(\texttt{regionprops}(\texttt{label}(S_m^u)))$ get largest connected region
6: $\quad$ **if** the area ratio of $P_m^u < \tau_a$ **then**
7: $\quad\quad bbox_m^u \leftarrow$ Random crop of $A_w(u_m^s)$
8: $\quad$ **else**
9: $\quad\quad bbox_m^u \leftarrow$ The bounding box of P_m^u
10: $\quad$ **end if**
11: $\quad bbox_m^x \leftarrow$ Calculate the bounding box for $(A_w(x_m^s), y_m^s)$ using a similar method in steps 3-10.
12: $\quad u'_m \leftarrow \texttt{Mix}(A_s(u_m),\ A_w(x_m^s)$ or $A_w(u_m^s))$ following $bbox_m^x$, $M_h(u_m)$, $bbox_m^u$, $M_l(u_m)$
13: $\quad (1-\lambda_m) \leftarrow$ The area ratio of $bbox_m^x$ or $bbox_m^u$
14: **end for**
15: $\lambda \leftarrow$ The average of λ_m
16: **return** Mixed data $\{u'_m\}_{m=1}^{B}$, loss weight λ

balanced labeled data and 1000k unlabeled data with an unknown distribution. To make an imbalanced version of the dataset, we only sample the labeled data, while the distribution of unlabeled data naturally differs from that of labeled data, i.e.,$\gamma_l \neq \gamma_u$. All images are 96×96 in size and the dataset can be downloaded from https://cs.stanford.edu/ acoates/stl10/.

- **ImageNet-127** ImageNet-127 [5] is naturally an imbalanced dataset, thus it doesn't require any further processing. Moreover, it has an imbalanced test set, which can validate scenario *3)*. To conserve computation resources, all images are down-sampled to 32×32 or 64×64 in size and the dataset can be downloaded from https://image-net.org/download-images.

Implementation details. Following previous training protocol [16], we conduct our experiments on CIFAR10-LT, CIFAR100-LT and STL10-LT using Wide ResNet-28-2 [24], and on ImageNet-127 using ResNet-50 [7]. We train the model with a batch size of 64 for $250k$ iterations, with an evaluation every 500 iterations. We use SGD with momentum as our optimizer and adopt a cosine learning rate decay strategy by setting the learning rate to $\eta cos(\frac{7\pi t}{16T})$, where η is the initial learning rate, t is the current iteration number and T is the total number of iterations. We set the balance parameter $\alpha = 0.5$ on CIFAR10-LT, CIFAR100-LT and STL10-LT, and set it to 0.2 on ImageNet-127. We set all EMA update weights as $\lambda = \lambda_d = \lambda_e = \lambda_\tau = 0.999$. The CAM threshold τ_c and area threshold τ_a are set to 0.8 and 0.1, respectively. The epoch number for starting to estimate the data quantity and entropy of unlabeled data is set to 5. We designate the final block as the CAM layer. We adopt Softmax$(\cdot)$ as the mapping function $\delta(\cdot)$. Our experiments are conducted on one NVIDIA Tesla V100 with the CentOS 7 system, using PyTorch 1.11.0 and Torchvision 0.12.0.

C. Pseudo-code for Our BEM Algorithm

We define the pseudo-code for our BEM and CamMix algorithm in Alg. 1 and 2, respectively.

D. Additional Experiment Results

In this section, we conduct a series of additional experiments to further demonstrate the effectiveness of our BEM.

More results with re-balancing methods when $\gamma_l \neq \gamma_u$. We present the results of combining with FixMatch and ACR under $\gamma_l \neq \gamma_u$ setup in Tab. 2. As shown in Tab. 1, we further combine our BEM with more re-balancing methods, including LA and ABC. Without incorporating any re-balancing method, BEM's performance is weaker than DASO in some settings, particularly in the reversed setting. After combining two re-balancing methods, BEM outperforms DASO in almost all settings. Further integration with ACR achieves the state-of-the-art results in all scenarios with an average 31.5% performance gain. In summary, our method needs to combine with re-balancing methods to enhance the re-balancing ability in challenging scenarios, and it in turn complements these methods.

More results on CIFAR100-LT. We also conduct experiments on CIFAR100-LT under $\gamma_l \neq \gamma_u$ setup in Tab. 2. Results show that our BEM outperforms DASO in almost all settings. By integrating with ACR, we can achieve the best results in all scenarios (32.7% accuracy gain). It further demonstrates that the complementation of BEM can boost the performance of most re-balancing methods.

Fine-grained results. In this experiment, we present the fine-grained results in Tab. 3. We compare our BEM with DASO and ACR in three settings. Our method surpasses DASO in all scenarios and further enhances the state-of-the-art method (ACR). In particular, our method significantly improves the performance of few-shot classes at the cost of negligible drop on head classes in the consistent setting. Moreover, in all settings, our method shows a large improvement in medium classes, which is brought by entropy-based learning.

BEM on balanced datasets. To verify the effect of our BEM on balanced datasets, we conduct experiments on balanced datasets with combinations of different SSL methods, in-

Table 1. Comparison of test accuracy with combinations of different baseline models 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 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 [18]	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/DASO [16]	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
w/LA [15]+DASO [16]	84.6±2.04	86.8±0.76	**72.6** ±0.38	78.5±1.31	72.7±1.45	79.7±0.44	**66.8**±0.62	75.7±0.50
w/LA [15]+BEM(Ours)	**85.3**±0.31	**88.5**±0.65	70.9±1.69	**79.8**±1.37	**72.9**±0.38	**81.8**±0.76	65.7±0.25	**76.8**±1.87
w/ABC [12]+DASO [16]	85.2±1.56	88.4±0.82	70.1±1.25	79.8±0.21	71.8±1.17	78.4±0.58	**67.3**±2.06	75.9±0.43
w/ABC [12]+BEM(Ours)	**85.9**±0.33	**89.0**±0.67	**71.2**±0.58	**80.1**±0.96	**73.1**±1.68	**81.4**±1.29	66.4±1.93	**76.7**±1.12
w/ACR [21]	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/ACR [21]+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

Table 2. Comparison of test accuracy with combinations of different baseline models under $\gamma_l \neq \gamma_u$ setup on CIFAR100-LT. The γ_l is fixed to 10. The best results for each diversion are in **bold**.

	CIFAR100-LT($\gamma_l \neq \gamma_u$)			
	$\gamma_u = 1$(uniform)		$\gamma_u = 1/10$(reversed)	
Algorithm	$N_1 = 50$ $M_1 = 400$	$N_1 = 150$ $M_1 = 300$	$N_1 = 50$ $M_1 = 400$	$N_1 = 150$ $M_1 = 300$
FixMatch [18]	45.5±0.71	58.1±0.72	44.2±0.43	57.3±0.19
w/DASO [16]	53.9±0.66	61.8±0.98	**51.0**±0.19	60.0±0.31
w/BEM(Ours)	**54.8**±0.55	**63.6**±0.91	50.8±0.25	**60.7**±0.12
w/LA [15]+DASO [16]	54.7±0.40	62.4±1.06	51.1±0.12	60.5±0.23
w/LA [15]+BEM(Ours)	**56.5**±0.43	**64.1**±0.87	**51.7**±0.20	**61.3**±0.17
w/ABC [12]+DASO [16]	53.4±0.53	62.4±0.61	**51.2**±0.19	60.8±0.39
w/ABC [12]+BEM(Ours)	**55.2**±0.35	**64.7**±0.87	51.1±0.10	**61.4**±0.29
w/ACR [21]	66.0±0.25	73.4±0.22	57.0±0.46	67.6±0.12
w/ACR [21]+BEM(Ours)	**68.1**±0.34	**75.9**±0.49	**58.0±0.28**	**68.4±0.13**

Table 3. Fine-grained results on CIFAR10-LT with $N_1 = 1500, M_1 = 3000, \gamma_l = 100$.

	Consistent($\gamma_u = 100$)				Uniform($\gamma_u = 1$)				Reversed($\gamma_u = 1/100$)			
Algorithm	Many	Medium	Few	All	Many	Medium	Few	All	Many	Medium	Few	All
DASO	**95.1**	78.6	60.4	78.1	89.6	84.4	85.7	86.3	84.0	71.6	68.2	74.3
BEM	94.7	78.0	67.0	79.8	91.7	88.1	90.7	89.4	82.3	80.2	73.3	78.7
ACR	93.9	81.6	75.3	83.4	92.8	90.6	97.9	93.5	90.7	83.8	**96.4**	89.7
ACR+BEM	92.3	**83.3**	**81.9**	**85.4**	**95.4**	**93.1**	**98.0**	**95.3**	**90.9**	**84.9**	95.8	**89.9**

cluding MeanTeacher, FixMatch, FlexMatch and SoftMatch. Specifically, we set $\alpha = 0$, meaning that we only consider the differences in class-wise uncertainty distribution. As shown in Tab. 6, our BEM enhances the performance of all baseline models, particularly the MeanTeacher, where our method gains an average of 21.4%, 26.9% and 25.0% improvement for three datasets. This demonstrates the potential of class-wise uncertainty re-balancing in enhancing model performance for balanced datasets.

Table 4. Ablation study on different sampling strategies. EFF. denotes the effective number.

	CBMB	ESS	EFF.	C10	STL10
Random				72.1	65.0
Quantity-based	✓		✓	74.9	66.5
Entropy-based		✓		74.4	65.9
w/o effective number	✓	✓		75.2	67.3
Ours	✓	✓	✓	**75.7**	**68.3**

Table 5. Ablation study on updating strategies of entropy selection threshold τ_e.

	C10	STL10
Baseline	67.8	56.1
$\tau_e = 0.1$	74.7	66.6
$\tau_e = 0.2$	75.2	67.2
$\tau_e = 0.4$	75.1	66.9
$\tau_e = 0.6$	74.4	66.4
w/ ours	**75.7**	**68.3**

Ablation study on sampling strategies. To evaluate the effect of our sampling strategy, we conduct a series of experiments by replacing the sampling function. Results are summarized in Tab. 4. Random sampling only improves performance slightly. Then, we split the class-balanced entropy-based sampling function and find that the results drop on both datasets. Further, we replace the effective number with the common number. Results indicate the effective number more accurately measures the class distribution of datasets.

Ablation on the updating strategy of entropy threshold τ_e. As shown in Tab. 5, we perform experiments to validate the effect of the entropy threshold τ_e updating strategy. When we filter the entropy mask with fixed thresholds, the

Table 6. Comparison of test accuracy on balanced datasets with combinations of different SSL methods, including MeanTeacher, FixMatch, FlexMatch and SoftMatch.

	CIFAR-10			CIFAR-100			STL-10	
Algorithm	40	250	4000	400	2500	10000	40	1000
MeanTeacher[20]	29.81±1.60	62.54±3.30	91.90±0.21	18.89±1.44	54.83±1.06	68.25±0.23	28.28±1.45	66.10±1.37
w/BEM(Ours)	**43.13**±2.55	**74.31**±1.79	**92.65**±0.23	**30.92**±3.69	**60.73**±2.14	**72.54**±0.19	**37.31**±2.59	**78.74**±1.38
FixMatch [18]	92.53±0.28	95.14±0.05	95.79±0.08	53.58±0.82	72.97±0.16	77.80±0.12	64.03±4.14	93.75±0.33
w/BEM(Ours)	**93.96**±0.37	**95.37**±0.03	**95.93**±0.11	**55.24**±0.93	**73.12**±0.14	**77.95**±0.11	**66.45**±3.29	**93.98**±0.65
FlexMatch [26]	95.03±0.06	95.03±0.09	95.81±0.01	60.06±1.62	73.51±0.20	78.10±0.09	70.85±0.01	94.23±1.62
w/BEM(Ours)	**95.08**±0.09	**95.21**±0.04	**95.98**±0.01	**60.83**±0.98	**73.94**±0.18	**78.72**±0.11	**72.11**±0.03	**94.39**±1.54
SoftMatch [2]	95.09±0.12	95.18±0.09	95.96±0.02	62.90±0.77	73.34±0.25	77.97±0.03	78.58±3.48	94.27±0.24
w/BEM(Ours)	**95.11**±0.08	**95.37**±0.06	**96.12**±0.07	**63.13**±0.92	**73.56**±0.08	**78.14**±0.08	**79.09**±3.87	**94.43**±0.38

Table 7. Ablation study on α.

	C10	STL10
1.0	74.7	67.0
0.7	75.5	67.3
0.5	**75.7**	68.3
0.3	74.4	**68.5**
0	73.8	67.5

Table 8. Ablation study on τ_c.

	C10	STL10
0.9	73.0	65.8
0.8	**75.7**	**68.3**
0.6	74.4	67.3
0.4	71.5	64.6
0.2	69.3	61.3

performance decreases and becomes unstable. Our EMA updating strategy achieves the best result, indicating that it adaptively adjusts the threshold following the training status of the model.

Ablation study on parameter α. As shown in Tab. 7, we verify the effect of α to balance the effective number and entropy in Eq. 7. Results show the best α on CIFAR10-LT and STL10-LT are 0.5 and 0.3, respectively. The visualization of sampling rate and class accuracy can be seen in Appendix E.

Ablation study on CAM threshold τ_c. In Tab. 8, we study the effect of CAM threshold τ_c on selected region. Results show that 0.8 is the best threshold on both datasets. It indicates that the precise selection of relevant regions is more advantageous for re-balancing long-tailed datasets.

Ablation study on the adding weight β. We conduct experiments to test the impact of the adding weight β in the equation $e_c = \beta e_c^u + (1-\beta)e_c^x$. The results in Tab. 9 indicate that weight addition has minimal impact. So we remove this parameter to simplify the number of hyperparameters.

Table 9. Ablation study on the adding weight β.

β	C10	STL10
0.7	75.1	67.7
0.5	75.7	**68.3**
0.3	**75.8**	68.1
0.1	74.9	67.9

Table 10. More comparison with class-wise data mixing methods.

	C10	STL10
FixMatch	67.8	56.1
w/UniMix [22]	72.9	66.0
w/MiSLAS [29]	73.4	66.2
w/Ours	**75.7**	**68.3**

Table 11. Comparison with AREA on supervised learning.

	C10-LT		C100-LT	
γ	200	50	200	50
CE	65.7	74.8	34.8	43.9
AREA [3]	**75.0**	82.7	**43.9**	**51.8**
Ours	74.7	**83.0**	40.3	49.7

More comparison with class-wise data mixing methods. We conduct additional experiments to compare our BEM with other class-wise data mixing methods [22, 29]. The results in Tab. 10 show that BEM outperforms them. We infer that these class-wise mixup methods are limited in not considering the uncertainty issue in LTSSL.

Comparison with AREA. We compare our BEM with AREA [3], which is a fully supervised learning method in long-tailed learning. Our BEM is different from AREA in three aspects: **1) Motivation:** AREA does not consider class-wise uncertainty. It optimizes the re-weighting strategy, which only focuses on data quantity, by exploring the spanned space of each class and relations between samples. While we propose to re-balance the class distribution of both data quantity and uncertainty, which is more suitable for LTSSL. **2) Task:** AREA focuses only on the class imbalance issue in the supervised learning diagram. While our method is specifically designed for LTSSL to further address the issue of uncertainty in unlabeled sample predictions, which can not be achieved by AREA. We also apply our BEM to supervised learning. Tab. 11 shows that BEM is competitive with AREA, demonstrating its flexibility and superiority. **3) Design:** AREA is based on the re-weighting strategy, using the effective area as class-wise weights in cross-entropy loss. While BEM is primarily based on re-sampling, where we use class-wise data quantity and uncertainty as sampling criteria for CamMix.

E. Additional Visualization Analysis

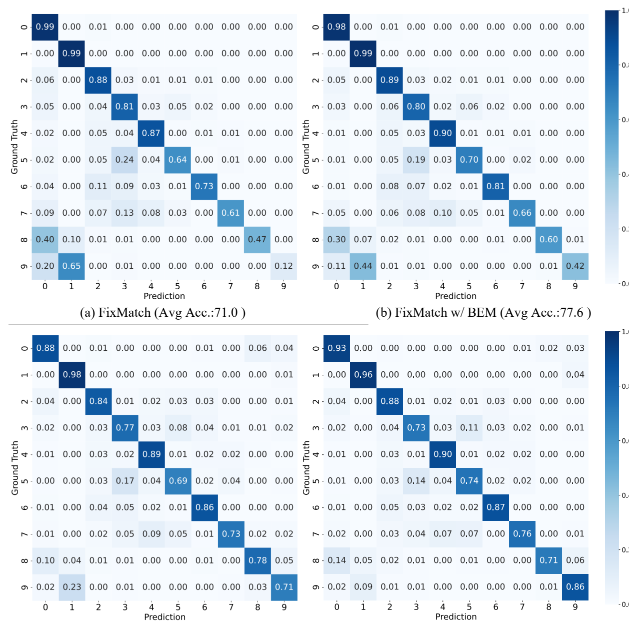
In this section, we provide additional visualization analysis to better understand our approach.

Visualization of confusion matrices on test set. We compare the confusion matrices of the prediction from the test set. We conduct experiments on CIFAR10-LT in the consistent scenario and apply our BEM to FixMatch and ACR, respectively. As shown in Fig. 1, the prediction of FixMatch is significantly biased towards the head classes, resulting in poor performance of the tail classes. Our method greatly alleviates this bias, improving both the tail performance and overall performance. ACR achieves good results in various classes, and our method further improves the performance of the tail classes, demonstrating the superiority and versatility of our method.

Visualization of precision and recall on the test set. We analyze the precision and recall on the test set to further verify the effect of our BEM. As shown in Fig. 2, we apply our method to FixMatch and ACR. The results show that the recall of tail classes achieves significant gains by combining our BEM with both models.

Visualization of train curves and test accuracy class distribution. We further assess the effect of BEM on FixMatch and ACR by plotting training curves and class-wise test accuracy. As shown in Fig. 3(a), the low entropy ratio increases, suggesting a large fraction of unlabeled data is used in the mixing as the training state becomes stable. As shown in Fig. 3(b), our method greatly improves the tail class performance of FixMatch and ACR.

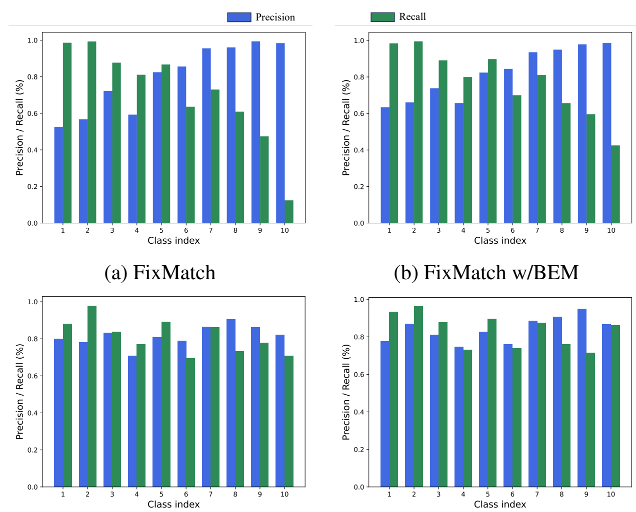
Visualization of the class distribution of sampling rate and accuracy under different α. We present the ablation study on α in Tab. 7. In addition, we further visualize the class distribution of sampling rate and accuracy under various α. Fig. 4 (a) shows that as α increases, the sampling rate of tail classes improves. When α is small, the sampling function pays attention not only to tail classes but also to middle



(a) FixMatch (Avg Acc.:71.0) (b) FixMatch w/ BEM (Avg Acc.:77.6)

(c) ACR (Avg Acc.:81.3) (d) ACR w/ BEM (Avg Acc.:83.5)

Figure 1. The confusion matrices of the test set on CIFAR10-LT under $\gamma_l = \gamma_u$ setup.



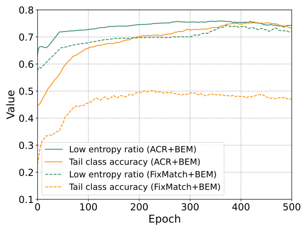
(a) FixMatch (b) FixMatch w/BEM

(c) ACR (d) ACR w/BEM

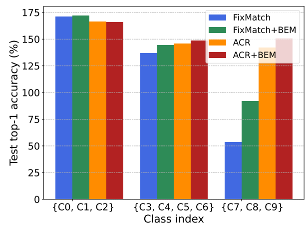
Figure 2. The precision and recall of the test set on CIFAR10-LT under $\gamma_l = \gamma_u$ setup.

classes with high uncertainty. In Fig. 4 (b), we can see that when $\alpha = 0.5$, both the tail class and the middle class with high uncertainty have relatively high accuracy, indicating it achieves the balance of data quantity and uncertainty.

More visualization of data mixing. We provide the intermediate images of the data mixing on STL10 in Fig. 5. To further illustrate the effectiveness of our CamMix, we also present additional visualization results on CIFAR10 in Fig. 5.

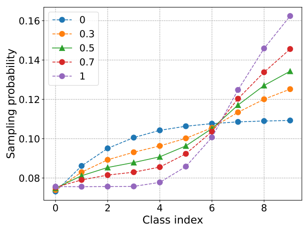


(a) Train curves

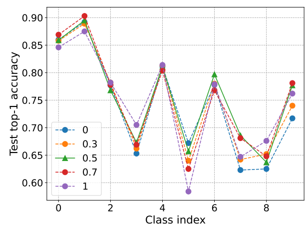


(b) Class-wise test accuracy

Figure 3. **(a):** Train curves for tail low entropy ratio and tail class accuracy. **(b):** Class distribution of test accuracy over different methods. C0 and C9 are the head and tail classes, respectively.



(a) Class-wise sampling rate



(b) Class-wise test accuracy

Figure 4. Class distribution of sampling rate and test accuracy under various α on CIFAR10-LT ($\gamma_l = \gamma_u = 100$) using FixMatch.

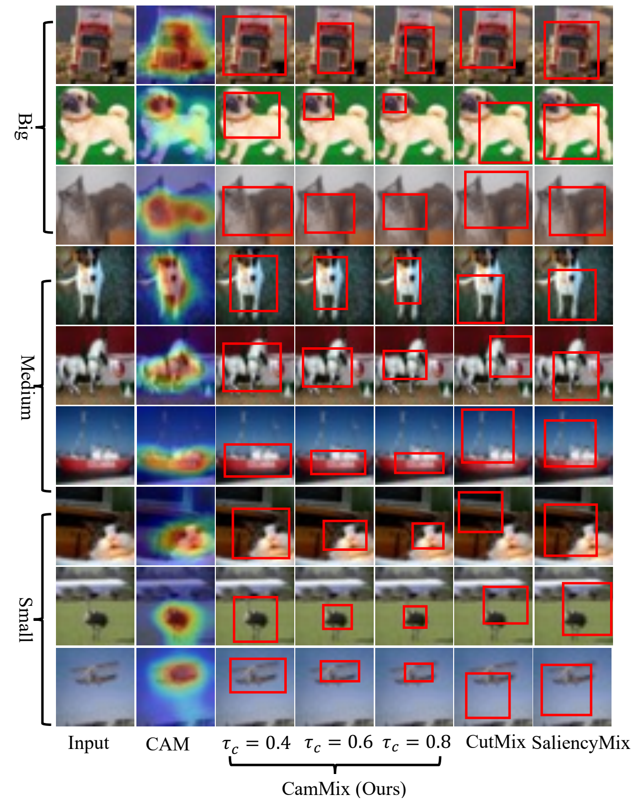
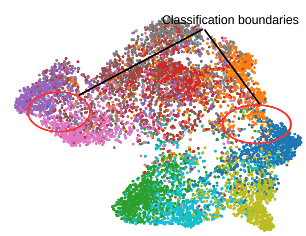
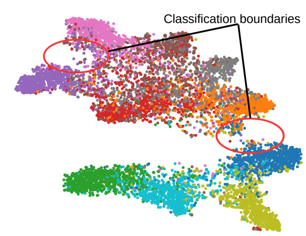


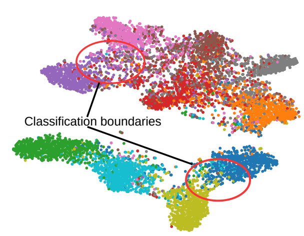
Figure 5. The visualization of data mixing process for CutMix, SaliencyMix, and CamMix on CIFAR10-LT. The red box indicates the image area selected by data mixing.



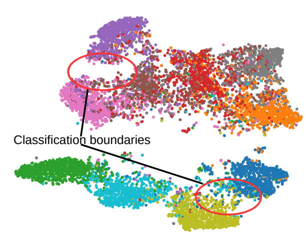
(a) FixMatch



(b) FixMatch w/BEM



(c) ACR



(d) ACR w/BEM

Figure 6. Comparison of t-SNE visualization with combinations of FixMatch and ACR on the test set of STL10-LT when $\gamma_l \neq \gamma_u$.

We select three images for each target size. Based on the results from the two datasets, we can draw the following conclusions: 1) CutMix has a high degree of randomness and often selects the context region. 2) The localization ability of SaliencyMix needs to be optimized. The selection region is not precise and tends to choose numerous redundant areas. 3) CamMix greatly improves the localization ability due to the accuracy of CAM and excludes irrelevant redundant areas as τ_c value decreases.

More visualization of t-SNE As displayed in Fig. 4, we show the t-SNE of learning representations from the test data on CIFAR10-LT. We further conduct experiments on STL10-LT to visualize the learning representations when $\gamma_l \neq \gamma_u$. Results in Fig. 6 show that our method generates clearer classification boundaries for representations when $\gamma_l \neq \gamma_u$. Specially, the classification ability of FixMatch is relatively poor, with most clusters gathered together. Our method greatly enhances its classification ability.

F. Limitation and Future Work

A potential limitation is that the proposed BEM is restricted by only exploring the data mixing for the LTSSL classification task, while ignoring its further application for other vision tasks, such as object detection [1, 6, 30], semantic segmentation [8, 17, 19, 28] and others [13, 27]. It is worth noting that the application of semi-supervised learning for long-tailed objection detection [14, 25] and semantic segmentation [9, 10] is not trivial but much harder than the pure classification task, as it requires further predict object location or semantic mask. In the future, we will extend our BEM to more complex vision tasks to further demonstrate its effectiveness and adaptability.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 6

[2] 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. 4

[3] Xiaohua Chen, Yucan Zhou, Dayan Wu, Chule Yang, Bo Li, Qinghua Hu, and Weiping Wang. Area: adaptive reweighting via effective area for long-tailed classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19277–19287, 2023. 4

[4] 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. 1

[5] 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

[6] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 6

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 6

[9] Ruifei He, Jihan Yang, and Xiaojuan Qi. Re-distributing biased pseudo labels for semi-supervised semantic segmentation: A baseline investigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6930–6940, 2021. 6

[10] Hanzhe Hu, Fangyun Wei, Han Hu, Qiwei Ye, Jinshi Cui, and Liwei Wang. Semi-supervised semantic segmentation via adaptive equalization learning. *Advances in Neural Information Processing Systems*, 34:22106–22118, 2021. 6

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

[12] 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. 3

[13] 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. 6

[14] Yen-Cheng Liu, Chih-Yao Ma, Zijian He, Chia-Wen Kuo, Kan Chen, Peizhao Zhang, Bichen Wu, Zsolt Kira, and Peter Vajda. Unbiased teacher for semi-supervised object detection. In *International Conference on Learning Representations*, 2021. 6

[15] 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. 3

[16] 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, 3

[17] Bowen Shi, Dongsheng Jiang, Xiaopeng Zhang, Han Li, Wenrui Dai, Junni Zou, Hongkai Xiong, and Qi Tian. A transformer-based decoder for semantic segmentation with multi-level context mining. In *European Conference on Computer Vision*, pages 624–639. Springer, 2022. 6

[18] 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, 3, 4

[19] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7262–7272, 2021. 6

[20] 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. 4

[21] 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. 3

[22] Zhengzhuo Xu, Zenghao Chai, and Chun Yuan. Towards calibrated model for long-tailed visual recognition from prior perspective. *Advances in Neural Information Processing Systems*, 34:7139–7152, 2021. 4

[23] 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. 1

[24] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 2

[25] Yuhang Zang, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Semi-supervised and long-tailed object detection with cascadematch. *International Journal of Computer Vision*, 131 (4):987–1001, 2023. 6

[26] 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. 4

[27] Hongwei Zheng, Han Li, Bowen Shi, Wenrui Dai, Botao Wang, Yu Sun, Min Guo, and Hongkai Xiong. Actionprompt: Action-guided 3d human pose estimation with text and pose prompting. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pages 2657–2662. IEEE, 2023. 6

[28] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021. 6

[29] Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16489–16498, 2021. 4

[30] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 6