IMWA: Iterative Model Weight Averaging Benefits Class-Imbalanced Learning Tasks

Zitong Huang[1], Ze Chen[2] , Bowen Dong[1], Chaoqi Liang[1], Erjin Zhou[2], Wangmeng Zuo[1✉]

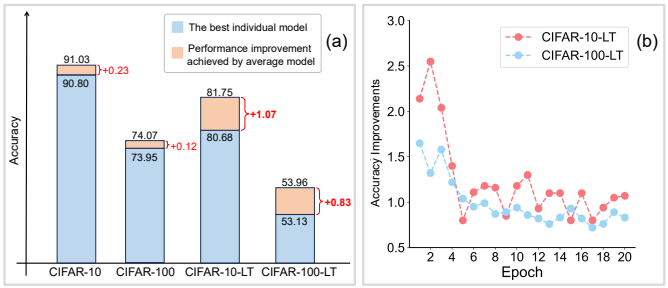
***Abstract*—Model Weight Averaging (MWA) is a technique that seeks to enhance model's performance by averaging the weights of multiple trained models. This paper first empirically finds that 1) the vanilla MWA can benefit the class-imbalanced learning, and 2) performing model averaging in the early epochs of training yields a greater performance improvement than doing that in later epochs. Inspired by these two observations, in this paper we propose a novel MWA technique for class-imbalanced learning tasks named Iterative Model Weight Averaging (IMWA). Specifically, IMWA divides the entire training stage into multiple episodes. Within each episode, multiple models are concurrently trained from the same initialized model weight, and subsequently averaged into a singular model. Then, the weight of this average model serves as a fresh initialization for the ensuing episode, thus establishing an iterative learning paradigm. Compared to vanilla MWA, IMWA achieves higher performance improvements with the same computational cost. Moreover, IMWA can further enhance the performance of those methods employing EMA strategy, demonstrating that IMWA and EMA can complement each other. Extensive experiments on various class-imbalanced learning tasks, *i.e.*, class-imbalanced image classification, semi-supervised class-imbalanced image classification and semi-supervised object detection tasks showcase the effectiveness of our IMWA.**

***Index Terms*—Model Weight Averaging, Class-Imbalanced Learning.**

I. Introduction

Model Weight Averaging (MWA) aims at merging optimized parameters from different models to obtain model parameters with better prediction accuracy. Compared to conventional model ensembling methods (*i.e.*, fusing predictions from multiple pretrained models), MWA methods usually obtain similar or even better performance while lead to less computation cost. Recently, several studies explored the application of MWA in various fields. For example, Model Soups [57] chose candidate weights from dozens of models fine-tuned from pretrained models. Matena *et al*. [39] proposed Fisher Merging, which merges various models' functionalities by calculating a weighted average of their respective parameters using the Fisher information. Other several studies [17], [44] illustrated the effects of MWA on continual learning and out-of-distribution scenarios.

In this work, we empirically deriving two observations from applying MWA on image classification task. **1)** The first observation is that *the degree of class balance in the training set affects the performance improvement by MWA*. As shown in Fig. 1(a), the average model performs better than the best individual model across all selected datasets, while it shows more significant performance improvements on CIFAR-10-LT and CIFAR-100-LT, both of which are class-imbalanced datasets. We infer that two models trained on class-imbalanced datasets exhibit greater model's diversity (See Fig. 1(c)) than that on class-balanced datasets, hence the performance of their average model shows a more pronounced improvement [33]. **2)** The second observation is that *performing model averaging in the early epochs of training yields a greater performance improvement than doing that in later epochs*. Experimentally, we perform model weight averaging at different epochs respectively, and the results are shown in Fig. 1(b). For example, on CIFAR-10-LT (the red line), It achieved over a **2%** performance improvement when performing the weight average between epoch 1 and epoch 3, but achieved only about a 1% improvement near the end of training. This phenomenon can also be observed from the results on CIFAR-100-LT.



	CIFAR-10	CIFAR-100	CIFAR-10-LT	CIFAR-100-LT
L2 Distance	1.34	2.36	**7.28**	**5.79**

(c)

Fig. 1. **Observations** from applying MWA on image classification task. The averaged model is obtained by two trained individual ResNet-34 [15]. **(a) Accuracy** of the best individual model *vs*. average model on both class-balanced (*i.e.*, CIFAR-10 and CIFAR-100) and class-imbalanced (*i.e.*, CIFAR-10-LT and CIFAR-100-LT) datasets, where average model performs better on the class-imbalanced scenario. **(b) Improvements** of the average model versus each epoch, where performing model weight averaging in the early stage of training brings higher improvement. **(c) L2 distance** between two trained individual models is higher in the class-imbalanced datasets.

From these two observations, one can notice that MWA may be more beneficial for class-imbalanced learning tasks. Moreover, performing weight averaging early during training can maximizes the extent of performance improvement. These observations have inspired us that, if we use the early averaged model as a new initialization and continue training, iteratively repeating the "training-average" process, could it yield better

Zitong Huang, Bowen Dong, Chaoqi Liang and Wangmeng Zuo are with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China (e-mail: cswmzuo@gmail.com).

Ze Chen and Erjin Zhou are with the Megvii Research, Megvii Technology Limited, Beijing, China.

results than just performing weight averaging once? To this end, in this paper we propose a novel MWA technique for class-imbalanced learning named **I**terative **M**odel **W**eight **A**veraging (IMWA), which iterates the parallel training and weights average process for many times. Specifically, IMWA splits the whole training stage into several episodes. In each episode, multiple models are trained in parallel from the same initialization model weight, with same training iterations but different data sampling orders, and then they are average into one model. Next, the weight of the average model is treated as a new initialization of the next episodes, which formulates an "iterative" manner. Compared to the vanilla MWA approaches, IMWA enables model to obtain performance improvement from weight average operation at each episode, while incurs almost no additional computational overhead due to the low computational complexity of the average operation.

We select three distinct class-imbalanced vision tasks to assess effectiveness of IMWA: class-imbalanced image classification, class-imbalanced semi-supervised image classification, and semi-supervised object detection. In the context of class-imbalanced semi-supervised classification, the labeled dataset exhibits class imbalance. Furthermore, the model dynamically generates pseudo labels for unlabeled images, resulting in an often uncontrollable number of pseudo labels for each class. while for the setting of semi-supervised object detection tasks, the common object detection datasets (e.g. **MS-COCO**) are naturally class-imbalanced. In each task, we adopt IMWA on several state-of-the-art (SoTA) methods of corresponding tasks and conduct experiments on various benchmarks for evaluation. Furthermore, considering that the latter two tasks often employ the Exponential Moving Average (EMA) technique [34], [42] to stabilize training and obtain better performance, we also explore the synergistic collaboration of IMWA and EMA. Extensive experimental results show that our IMWA is more beneficial than vanilla MWA in improving performance for these methods, meanwhile, the performance improvement achieved by combining IMWA and EMA surpasses that of using IMWA or EMA individually. In conclusion, the contributions of this paper are summarized as follows:

1) We empirically observe that vanilla MWA performs well in class-imbalanced learning, and performing model averaging in the early epochs of training yields a greater performance improvement than doing that in later epochs, which inspire us to propose a novel MWA technique for class-imbalanced learning tasks named IMWA.
2) IMWA executes the multiple model parallel training and weights averaging process in an iterative manner. In addition, we further adopt IMWA to methods that utilize EMA techniques, thus demonstrating that IMWA and EMA complement each other.
3) Extensive experiments on various benchmarks of different class-imbalanced learning tasks demonstrate that the IMWA method is superior to the vanilla MWA and can effectively improve the performance of models.

II. Related Works

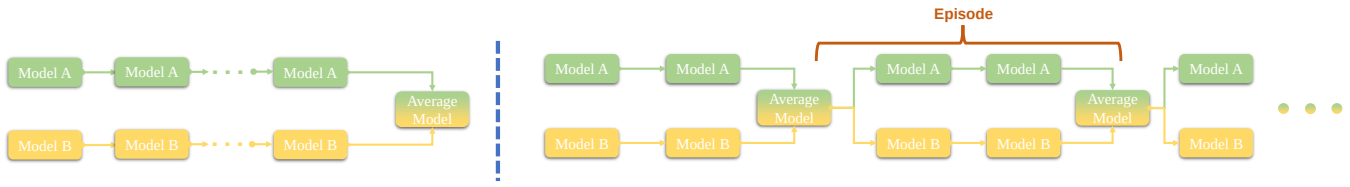
A. Model Weight Averaging

Model weight averaging aims at averaging weights of multiple trained models to improve performance. The existing model averaging approaches can be concluded into two branches. The first kind of methods [14], [17], [39], [44], [47], [57] trained multiple models simultaneously, and then they average the weights of each converged model to a stronger model. And the counterparts [3], [18] showed that simple averaging of multiple checkpoints along one training trajectory of SGD with a typical learning rate schedule, can also improve generalization than conventional training. Then, several following methods [34], [42], [45] involved the Exponential Moving Average (EMA) approach to enhance performance and training stability. Averaging models along the dimension of a individual model's training trajectory can effectively reduce the computational cost during training, but insufficient diversity inevitably lead to performance bottlenecks [33]. In this paper, our proposed IMWA extend the first branch approaches while incorporating the concept of the second branch that "average in the training loop" by iterating the training-averaging process. In addition, IMWA can further enhance those methods based on EMA, indicating that IMWA and the second branch approaches are complementary. Note that some concurrent works [19], [64] also proposed to employ model weight averaging during the training process, While we further analyses the performance of IMWA in class-imbalanced learning tasks and explore the the synergistic collaboration of IMWA and EMA technique.

B. Class-Imbalanced Learning Tasks

Class-imbalanced learning tasks can be summarized into:

1) Class-Imbalanced Image Classification. This task aims to train an image classifier on a class-imbalanced training dataset. Vanilla training with cross-entropy loss may result in overfiting issue on head classes. To address the issue, [8], [29], [63], [67] adjusted sampling rate for each class to re-balanced the training data for model. [1], [43], [48], [49], [66] mitigated the class-imbalanced problem by assigning different class weights for training data. [28], [41], [58], [65] applied single expert learning and knowledge aggregation to alleviate the class-imbalanced problem. [37], [38], [51] proposed to involve the vision-language model, introducing language knowledge and external database to adjust the output scores. Recently, [10], [61] involved ViT to alleviate the class-imbalanced problem by propose masked generative pretraining or prompts tunning. Different from these methods above, IMWA aims to enhance the model's generalization ability from the perspective of averaging model parameters. Therefore, IWMA is plug-and-play approach and applied to these existing methods easily without modifying their origin structure or training details.
2) Semi-Supervised Class-Imbalanced Image Classification. This task aims to train an image classifier on a labeled set and unlabeled set, both of which are suffered from the class-imbalanced problem [42], [55]. In addition, pseudo



(a) Naïve MWA (b) IMWA

Fig. 2. The **illustration** of **(a)** vanilla Model Weight Averaging, and **(b)** our proposed **Iterative Model Weight Averaging (IMWA)**. IMWA splits the whole training stage into several episodes. At each episode, multiple models are trained in parallel from a same initialization model weight, with same training iterations but different data orders, and then they are average into one model. Then, the weight of the average model will be treated as a new initialization for the next episodes, which forms a "iterative" manner.

labels on the unlabeled set are changeable during training, which can also lead to class imbalance. To tackle this problem, [20], [54] assumed that the class distribution in the labeled and unlabeled sets are consistent, and using this assumption to control the generation of pseudo-labels for unlabeled data. [23] added an extra balanced classifier, which is trained to be balanced across all classes by using a mask that re-balances the class distribution. While [42], [55] proposed that the class distribution of labeled set and unlabeled set are different, and present dynamic or adaptive manner to learn the truth distribution. We applied IMWA to several SOTA approaches of this task, which demonstrated that IMWA is also beneficial for semi-supervised learning scenarios in terms of performance. Notice that most approaches use EMA (Exponential Moving Average) technique to maintain a model with better generalization. Therefore, in this paper we also propose a improved version of IMWA for collaborating with EMA, which demonstrating that IMWA and EMA complement each other.

3) Semi-Supervised Object Detection. This task aims to train an object detector on a full-annotated labeled set combining with a huge unlabeled set. Due to the imbalance in the number of instances for different classed in most common object detection datasets, this task also suffers from class-imbalanced problem. Recent mainstream methods presented the teacher-student training framework [4], [5], [21], [24], [26], [33]–[35], [59], [69], where the student is optimized with pseudo-labels generated by teacher model while the weights of teacher model are updated by student model gradually with EMA. In this paper, we also apply our IMWA on existing SSOD approaches to IMWA is benefit not only for image classification task, but also for object detection field.

III. Method

A. Vanilla MWA

The vanilla MWA approach [57] consists of two steps: the parallel training step and the averaging step. Given a set of models $\{\boldsymbol{f}(\mathbf{x};\boldsymbol{\theta}_1,\mathcal{H}_1),...,\boldsymbol{f}(\mathbf{x};\boldsymbol{\theta}_M,\mathcal{H}_M)\}$, $\mathbf{x}$ denotes the input data, M denotes the number of models involved; $\boldsymbol{\theta}_m$ denotes model weights and the $\mathcal{H}_m$ denotes a group of training hyper-parameters, where $m \in [1,...,M]$. Note that all these models share the same network architecture, while their weights $\boldsymbol{\theta}_m$ and corresponding training hyper-parameter $\mathcal{H}_m$ may be different.

During the parallel training step, all models' weights are assigned by the same initialization. Then they are trained in parallel. And during the averaging step, corresponding weights trained models are averaged by Eq. (1),

$$\boldsymbol{\theta}_{\mathrm{MWA}} = \sum_{m=1}^{M} \alpha_m \boldsymbol{\theta}_m, \tag{1}$$

where $\boldsymbol{\theta}_{\mathrm{MWA}}$ denotes the weights of average model, $\boldsymbol{\theta}_m$ denotes the weights of a individual trained model, α_m denotes the coefficient for $\boldsymbol{\theta}_m$ which satisfies $\sum_{m=1}^{M} \alpha_m = 1$, and M denotes the total number of models. In the inference stage, only $\boldsymbol{\theta}_{\mathrm{MWA}}$ will be evaluated to obtain the performance score.

Previous MWA methods [39], [44], [57] typically involve training each model to convergence before averaging their weights. While our empirical findings suggest that balancing the parameters early in the training of each model (for example, during the first few epochs) results in a more significant performance improvement in the averaged model compared to averaging after convergence, as mentioned in Sec. I. This inspires us to consider whether the averaged model at early epoch could serve as a new starting point for training, and then continuously iterate this process of "training" and "weight averaging." To this end, we propose Iterative Model Weight Averaging.

B. Iterative Model Weight Averaging

1) Overview: The key idea of IMWA is employing the vanilla MWA approach during the training process, which treats the average model as a new initialization point to repeat the vanilla MWA process for many times. We refer to each iteration process as an "episode" and split the whole training process into E episodes. The pseudo code of IMWA can be refer to Algorithm 1. And corresponding illustration is shown in Fig 2. Specifically, the process of our IMWA can be divided into the following key components:

2) Initialization: At the beginning of the whole training process, the weights of M individual models are all initialized with the same weights $\boldsymbol{\theta}^{(0)}$:

$$\boldsymbol{\theta}_m^{(0)} \leftarrow \boldsymbol{\theta}^{(0)}, \tag{2}$$

Algorithm 1: Iterative Model Weight Averaging

Input: A specific algorithm $\mathcal{A}(\cdot)$; the initialization weights $\boldsymbol{\theta}^{(0)}$; training hyper-parameters $\{\mathcal{H}_1, ..., \mathcal{H}_M\}$; the number of episode E; the number of training iterations of in each episode T_e; the number of models M; the training dataset $\mathcal{D}$;
Output: the average weights $\boldsymbol{\theta}^{(E)}$ after training;
for $e = 1, \cdots, E$ **do**
  for $m = 1, \cdots, M$ **do**
    Update $\boldsymbol{\theta}_m^{(e-1)} \leftarrow \boldsymbol{\theta}^{(e-1)}$;
    Compute $\boldsymbol{\theta}_m^{(e-1)'} \leftarrow \mathcal{A}(\boldsymbol{\theta}_m^{(e-1)}, \mathcal{D}, \mathcal{H}_m, T_e)$;
  end
  Compute $\boldsymbol{\theta}^{(e)} = \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{\theta}_m^{(e-1)'}$;
end
Return $\boldsymbol{\theta}^{(E)}$

where $m \in \{1, ..., M\}$. Note that $\boldsymbol{\theta}^{(0)}$ can be obtained by random initialization (for CIIC and CISSIC tasks) or pretraining backbone (for SSOD task). Similar to vanilla MWA, we introduce a unique hyper-parameter set $\mathcal{H}_m$ for each model for optimization during training.

3) Episode: Similar to the vanilla MWA, each episode consists of two steps: parallel training step and weight average step. At the parallel training step, M models are trained in parallel via a specific algorithm $\mathcal{A}(\cdot)$ on a certain dataset $\mathcal{D}$ for T_e iterations, where $T_e = \frac{T}{E}$ and T denotes the total number of training iterations of whole training stage.

Formally, we leverage Eq. (3) to optimize each model $\boldsymbol{f}$:

$$\boldsymbol{\theta}_m^{(e-1)'} \leftarrow \mathcal{A}(\boldsymbol{\theta}_m^{(e-1)}, \mathcal{D}, \mathcal{H}_m, T_e), \tag{3}$$

where the superscript e represents the index of episode. At the weight average step, we average these trained weights $\boldsymbol{\theta}_m^{(e-1)'}$ by Eq. (4):

$$\boldsymbol{\theta}^{(e)} = \sum_{m=1}^{M} \alpha_m \boldsymbol{\theta}_m^{(e-1)'}, \tag{4}$$

where the coefficient α_m referred in vanilla MWA is set to $\frac{1}{M}$ to avoid cumbersome manual tuning.

4) Iterative loop: After obtaining the average weight $\boldsymbol{\theta}^{(e)}$, we re-assign it to each model:

$$\boldsymbol{\theta}_m^{(e)} \leftarrow \boldsymbol{\theta}^{(e)}. \tag{5}$$

Now each model has updated with new weights $\boldsymbol{\theta}_m^{(e)}$, then conduct the next episode training.

5) Evaluation: Upon whole training completion, we take the average model of the last episode $\boldsymbol{\theta}^{(E)}$ as the final model and then evaluate its performance on the test set.

C. Collaboration with EMA

Now we have introduced the process of IMWA in the previous sub-section, where we assume that only optimized model $\boldsymbol{f}$ is involved to the selected SOTA algorithm $\mathcal{A}(\cdot)$. However, many recent works [34], [40], [42], [55] of CISSIC and SSOD adopt the EMA strategy to improved effectiveness and stability. In these works, an extra EMA model $\boldsymbol{f}(\mathbf{x}; \boldsymbol{\omega}, \mathcal{H})$ is updated at the t iteration with the optimized model $\boldsymbol{f}$ by Eq. (6)

$$\boldsymbol{\omega}^t = \lambda \boldsymbol{\omega}^{t-1} + (1-\lambda)\boldsymbol{\theta}^t, \tag{6}$$

where λ is the coefficient and usually set to close to 1. Involving the EMA model necessitates modifications to our IMWA. To extend our IMWA to these adopting EMA methods, we propose the following modifications:

1) To merge the EMA strategy into the baseline method $\mathcal{A}(\cdot)$, we modify the Eq. (3) to Eq. (7):
$$\boldsymbol{\theta}_m^{(e-1)'}, \boldsymbol{\omega}_m^{(e-1)'} \leftarrow \mathcal{A}(\boldsymbol{\theta}_m^{(e-1)}, \boldsymbol{\omega}_m^{(e-1)}, \mathcal{D}, \mathcal{H}_m, T_e). \tag{7}$$
2) Then we extend Eq. (4) and Eq. (5) for the weights of EMA models:
$$\boldsymbol{\omega}^{(e)} = \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{\omega}_m^{(e-1)'}, \tag{8}$$
$$\boldsymbol{\omega}_m^{(e)} \leftarrow \boldsymbol{\omega}^{(e)}. \tag{9}$$
3) Upon whole training completion, we take the average EMA model of the last episode $\boldsymbol{\omega}^{(E)}$ as the final model and then evaluate its performance on the test set.

TABLE I
PERFORMANCE OF OUR IMWA APPLYING TO BCL [71] AND LIVT [61] ON THREE CIIC BENCHMARKS. "IN-LT" IS THE ABBREVIATION FOR "IMAGENET-LT" AND "INAT-18" IS THE ABBREVIATION FOR INATURALIST 2018.

Methods	IN-LT	iNat18	Places-LT
ACE [2]	56.6	72.9	-
PaCo [7]	58.2	73.2	41.2
TADE [65]	**58.8**	72.9	40.9
TSC [30]	52.4	69.7	-
GCL [29]	54.5	71.0	40.6
TLC [25]	55.1	-	-
NCL [28]	57.7	**74.2**	**41.5**
Bread [32]	44.0	70.3	39.3
DOC [53]	55.0	71.0	-
DLSA [60]	57.5	72.8	39.0
BCL [71]	56.0	71.8	39.4
BCL+IMWA	**57.3**	**72.5**	**40.3**
LiVT [61]	58.2	75.8	35.1
LiVT+IMWA	**59.0**	**76.6**	**36.7**

IV. EXPERIMENTS

A. Evaluation Tasks and Experimental Setup

This paper selects three distinct class-imbalanced vision tasks to assess effectiveness of IMWA: class-imbalanced image classification (CIIC), class-imbalanced semi-supervised image classification (CISSIC), and semi-supervised object detection (SSOD). The problem definitions of these tasks are presented below:

- CIIC aims to training an image classifier with a class-imbalanced dataset $\mathcal{D}_{\text{CI}}$. In formal, $\mathcal{D}_{\text{CI}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_{\text{CI}}}$. $\mathbf{x}_i \in \mathbb{R}^{w \times h \times 3}$ denotes an RGB image, where w and h

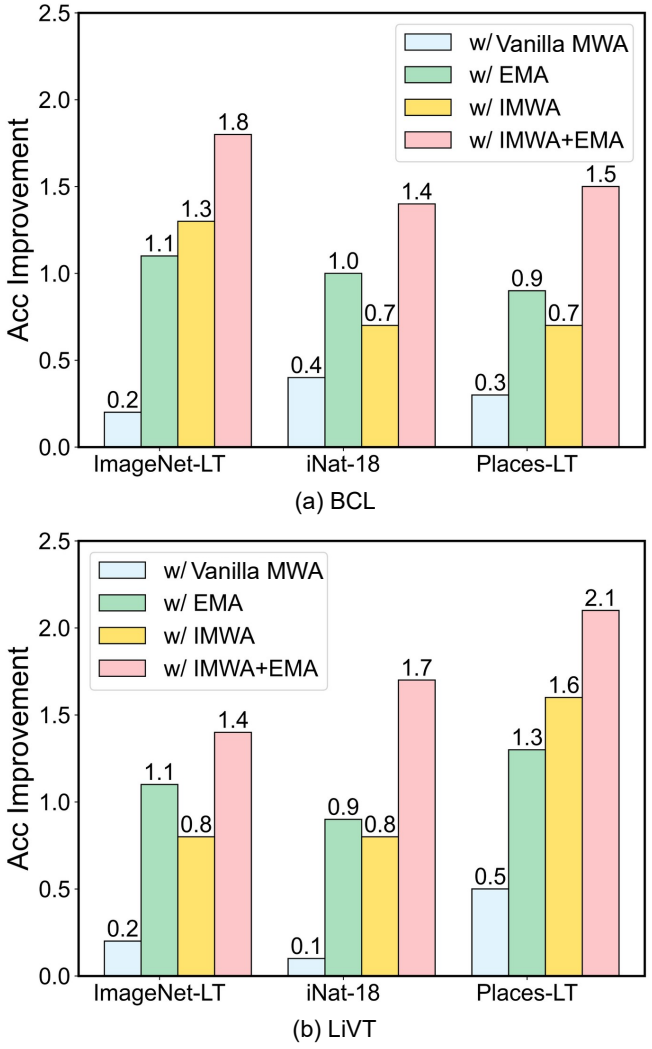


Fig. 3. **Comparison** of our IMWA and other MWA approaches in terms of achieving performance improvements for **(a)** BCL [71] and **(b)** LiVT [61] on **three CIIC benchmarks**.

indicate the width and height of image. $y_i \in \{1, ..., C\}$ denotes the label of $\mathbf{x}_i$ where C is the total number of class in $\mathcal{D}_{\text{CI}}$. $N_{\text{CI}} = \sum_{c=1}^{C} n_c$ denotes the total number of training pairs, where n_c denotes the of number training pairs belonging to the class c. In this paper we assume the $\{n_c\}_{c=1}^{C}$ obeys a long-tailed distribution, which is commonly adopted in previous works. We set $n_1 \geq n_2 \geq \cdots \geq n_C$ and imbalance ratio $\gamma = \frac{n_1}{n_C}$ to represent the degree of imbalance.

- CISSIC aims to training an image classifier with a labeled dataset $\mathcal{D}^l_{\text{CISS}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N^l_{\text{CISS}}}$ and an unlabeled dataset $\mathcal{D}^u_{\text{CISS}} = \{\mathbf{x}_i\}_{i=1}^{N^u_{\text{CISS}}}$, where N^l_{CISS} and N^u_{CISS} denotes the number of images in $\mathcal{D}^l_{\text{CISS}}$ and $\mathcal{D}^u_{\text{CISS}}$ respectively. Similar to CIIC, both $\mathcal{D}^l_{\text{CISS}}$ and $\mathcal{D}^u_{\text{CISS}}$ are class-imbalanced. we set γ^l and γ^u as the imbalance ratios for these two sets.
- SSOD aims to training an object detection model in a semi-supervised manner, in which a set of full-labeled images $\mathcal{D}^l_{\text{SSOD}} = \{(\mathbf{x}_i, \mathcal{B}_i)\}_{i=1}^{N^l_{\text{SSOD}}}$ combination with a set of unlabeled images $\mathcal{D}^u_{\text{SSOD}} = \{\mathbf{x}_j\}_{j=1}^{N^u_{\text{SSOD}}}$ are available. The $\mathcal{B}_i = \{(\mathbf{b}_k, y_k)\}_{k=1}^{K_i}$ denotes the box-level annotations of $\mathbf{x}_i$ consisting of K_i bounding box labels, where $\mathbf{b}_k$ is the k-th bounding box coordinates, and $y_k \in \{1, ..., C\}$ is the class label of $\mathbf{b}_k$.

For each evaluation task, we apply IMWA to several recent SOTA methods , and then select various widely used benchmarks to evaluate the performance. Specifically:

- For CIIC, we apply IMWA to LiVT [61] and BCL [71]. We evaluate our IMWA for CIIC on ImageNet-LT [9], [36], iNaturalist 2018 (iNat18) [52] and Places-LT [68] benchmarks.
- For CISSIC, we apply IMWA to ACR [55] and DASO [42]. We evaluate our IMWA for CISSIC on CIFAR-10/100-LT [22], STL-10 [6] and ImageNet-127 [16] benchmarks.
- For SSOD, we apply IMWA to Dense Teacher [69] , Soft Teacher [59] and Unbiased Teacher [34]. We evaluate our IMWA for SSOD on MS-COCO [31], PASCAL VOC [12] benchmarks.

B. Implementation Details

In this subsection we introduce some implementation details of our experiments. As mentioned above, we apply our IMWA to several SOTA methods for each task. For each method, we deploy our IMWA on its official implementation and follow its default training configurations (*e.g.*total training iteration, batch size, data augmentation, and other hyper-parameters) for fair comparison. For the hyper-parameters of our IMWA, we set E to 20 and M to 2 for all experiments in default. For the configurations of training components $\{\mathcal{H}_1, ..., \mathcal{H}_M\}$, we simply maintain an independent training dataloader for each individual model. Therefore, each model is trained with different sequence of sample data and optimizing direction during training period. For CIIC task and CISSIC task, $\boldsymbol{\theta}^{(0)}$ is initialized from the scratch. While for SSOD task, the backbone (feature extractor) of $\boldsymbol{\theta}^{(0)}$ is pretrained from ImageNet. The experiments of applying IMWA on Soft Teacher [59] are deployed on 8 $\times$ Tesla V100, and the other experiments are deployed on 8 $\times$ RTX 2080 Ti.

Besides comparing with these SOTA methods, we compare our IMWA with these performing the vanilla MWA and EMA approaches. For vanilla MWA approach, we perform it for all three tasks, just like the experiments of our IMWA. While as for EMA, because the methods for CISSIC and SSOD inherently incorporate EMA techniques (Please refer to their works for details), thus our experiments for these two tasks mainly explore the effect of synergistic collaboration between IMWA and EMA. To further present the comparison of IMWA and EMA, we conducted extra experiments for EMA on CIIC task. In addition, for each task, we also compare our IMWA with other recent influential methods to further show the effectiveness of IMWA.

TABLE II
PERFORMANCE OF OUR IMWA APPLYING TO DASO [42] AND ACR [55] UNDER CONSISTENT CLASS DISTRIBUTIONS ON **CISSIC BENCHMARKS**, *i.e.*, **CIFAR10-LT** AND **CIFAR100-LT**.

Algorithm	CIFAR10-LT				CIFAR100-LT			
	$\gamma=\gamma_l=\gamma_u=100$		$\gamma=\gamma_l=\gamma_u=150$		$\gamma=\gamma_l=\gamma_u=10$		$\gamma=\gamma_l=\gamma_u=20$	
	$N_1=500$ $M_1=4000$	$N_1=1500$ $M_1=3000$	$N_1=500$ $M_1=4000$	$N_1=1500$ $M_1=3000$	$N_1=50$ $M_1=400$	$N_1=150$ $M_1=300$	$N_1=50$ $M_1=400$	$N_1=150$ $M_1=300$
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 [41]	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 + LA [41]	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 [20]	76.6 ±0.92	80.8 ±0.62	68.2 ±0.94	76.7 ±1.13	50.5 ±0.78	59.9 ±0.32	44.4 ±0.65	53.8 ±0.43
w/ CReST+ [54]	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 [42]	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
FixMatch + ABC [23]	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 [42]	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
FixMatch [41]	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/ DASO [42]	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/ DASO+IMWA	**76.9 ±0.14**	**79.5 ±0.37**	**71.3 ±0.88**	**75.9 ±0.51**	**50.5 ±0.36**	**60.1 ±0.32**	**44.0 ±0.07**	**53.9 ±0.27**
w/ ACR [55]	81.6 ±0.19	84.3 ±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/ ACR+IMWA	**82.0 ±0.27**	**84.9 ±0.55**	**77.7 ±1.26**	**81.3 ±0.12**	**56.4 ±0.45**	**66.2 ±0.05**	**48.5 ±0.44**	**59.3 ±0.19**

TABLE III
PERFORMANCE OF OUR IMWA APPLYING TO DASO [42] AND ACR [55] UNDER INCONSISTENT CLASS DISTRIBUTIONS ON **CISSIC BENCHMARKS**, *i.e.*, **CIFAR10-LT** AND **STL-LT**.

Algorithm	CIFAR10-LT ($\gamma_l \neq \gamma_u$)				STL10-LT (γ_u = *N/A*)			
	$\gamma_u=1$ (uniform)		$\gamma_u=1/100$ (reversed)		$\gamma_l=10$		$\gamma_l=20$	
	$N_1=500$ $M_1=4000$	$N_1=1500$ $M_1=3000$	$N_1=500$ $M_1=4000$	$N_1=1500$ $M_1=3000$	$N_1=150$ $M=100k$	$N_1=450$ $M=100k$	$N_1=150$ $M=100k$	$N_1=450$ $M=100k$
FixMatch [45]	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 [20]	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 [54]	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+ [54]	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 [42]	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/ DASO+IMWA	**87.8 ±0.59**	**89.3 ±0.84**	**71.7 ±0.65**	**80.5 ±0.29**	**70.4 ±1.33**	**79.1 ±0.27**	**66.5 ±2.01**	**76.0 ±0.96**
w/ ACR [55]	92.1 ±0.18	93.5 ±0.18	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+IMWA	**92.8 ±0.34**	**94.4 ±0.14**	**85.5 ±0.36**	**90.7 ±0.28**	**78.6 ±0.40**	**83.9 ±0.26**	**76.4 ±0.59**	**81.9 ±0.12**

C. Datasets Details

1) Datasets of Class-Imbalanced image classification: We follow LiVT [61] and BCL [71] to perform our IMWA on ImageNet-LT, iNaturalist 2018 and Places-LT benchmarks.

- **ImageNet-LT:** ImageNet-LT is a long-tailed version ($\gamma = 256$) benchmark of vanilla ImageNet [9] by sampling a subset following the Pareto distribution with power value $\alpha = 0.6$. It consists of 115.8K images of 1000 classes in total with 1280 to 5 images per class.
- **iNaturalist 2018:** iNaturalist 2018 [52] is a large-scale dataset containing 437.5K images from 8,142 classes over multiple natural species and suffers from extremely class-imbalanced distribution ($\gamma = 512$).
- **Places-LT:** Places-LT is a long-tailed variant of the large-scale scene classification dataset Places [68], which contains 62.5K images from 365 categories of scenes in total with 4,980 to 5 ($\gamma = 996$).

2) Datasets of Class-Imbalanced Semi-Supervised Image Classification: We follow ACR [55] and DASO [42] to perform our IMWA on CIFAR-10/100-LT [22], STL-10-LT [6] and ImageNet-127 [13]. Given a imbalance ratio γ^l of labeled set, we set the number of labeled samples for class c as N_c; given the imbalance ratio of unlabeled set γ^u, we set the number of labeled samples for class c as $N_c =$ for class c as $M_c = M_1 \cdot \gamma^{u-\frac{c-1}{C-1}}$ with given M_1.

- **CIFAR-10-LT:** Following ACR [55] and and DASO [42], we perform our IMWA under $N_1 = 500, M_1 = 4000$ and $N_1 = 1500, M_1 = 3000$. Meanwhile, we set $\gamma^l = \gamma^u = 100$ and $\gamma^l = \gamma^u = 150$ respectively. We also perform IMWA on *uniform* ($\gamma^l = 100$) and *reversed* ($\gamma^u \in \{1, 1/100\}$) mentioned in DASO [42] to simulate various class distribution of unlabeled data.

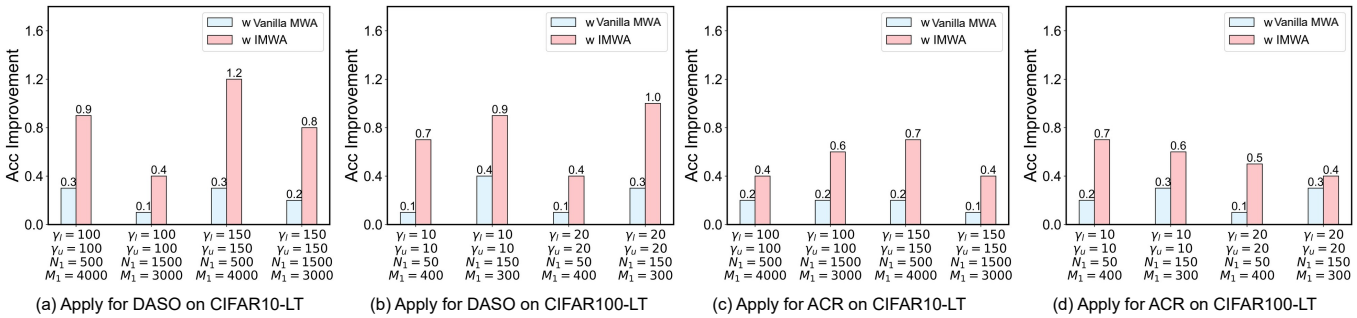


Fig. 4. **Comparison** of our IMWA and vanilla MWA approaches in terms of achieving performance improvements for DASO [42] and ACR [55] under consistent class distributions on **CISSIC benchmarks**, *i.e.*, **CIFAR10-LT** and **CIFAR100-LT**.

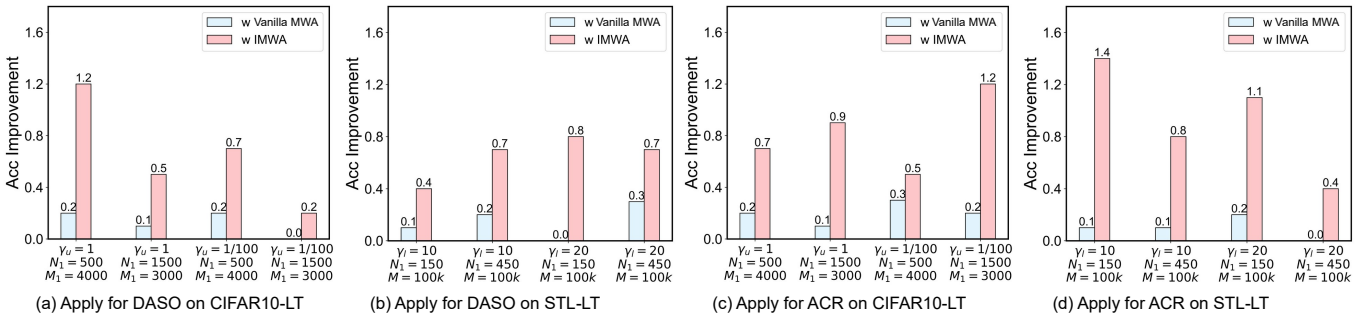


Fig. 5. **Comparison** of our IMWA and vanilla MWA approaches in terms of achieving performance improvements for DASO [42] and ACR [55] under inconsistent class distributions on CISSIC benchmarks, *i.e.*, **CIFAR10-LT** and **STL-LT**.

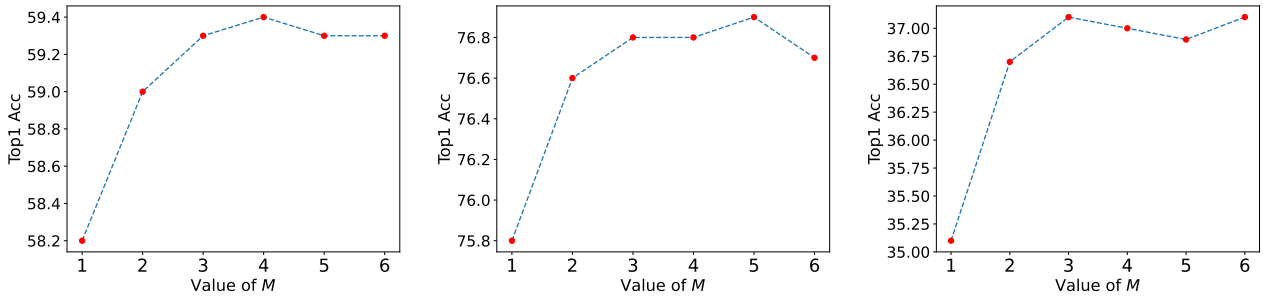


Fig. 6. **Ablation study** of different value of M by LiVT+IMWA on **three CIIC benchmarks**.

- **CIFAR-100-LT:** We test our IMWA under $N_1 = 50, M_1 = 400$ and $N_1 = 150, M_1 = 300$. The imbalance ratio is set to $\gamma^l = \gamma^u = 10$ and $\gamma^l = \gamma^u = 20$. For the uniform and reversed unlabeled data class distributions, we set $\gamma^l = 10$ and $\gamma^u \in \{1, 1/10\}$.
- **STL-10-LT:** Following DASO, we test our IMWA under $\gamma^l \in \{10, 20\}$. Note that the ground-truths of unlabeled set are unknown, therefore we simply use all the unlabeled data instead of sampling.
- **ImageNet-127:** ImageNet127 is a naturally class-imbalanced dataset in long-tailed distribution, hence we do not need to construct the datasets manually. Following ACR, we downsample the image size to 32 × 32 and 64 × 64.

3) Datasets of Semi-Supervised Object Detection: We follow Unbiased Teacher [34], Soft Teacher [59] and Dense Teacher [69] to perform our IMWA on PASCAL VOC [12] and MS-COCO [31] benchmarks.

- **MS-COCO:** MS-COCO [31] dataset is a large scale visual recognition benchmark including 80 object categories with a total of 123k labeled images for object detection task, where the `train2017` set and `val2017` set contain 118k images and 5k images respectively. To build the semi-supervised setting, we follow the data split protocol of [34] to sample 1%, 2%, 5% and 10% images of the `train2017` set as the labeled set. And the rest part of images are treated as the unlabeled set. `val2017` is viewed as the test set to evaluate our method.

TABLE IV
PERFORMANCE OF OUR IMWA APPLYING TO DASO [42] AND ACR [55] ON CISSIC BENCHMARK, *i.e.*, IMAGENET-127 UNDER DIFFERENT INPUT SIZE.

Methods	32 × 32	64 × 64
FixMatch [45]	29.8	42.3
w/ DARP [20]	30.5	42.5
w/ DARP+cRT [20]	39.7	51.0
w/ CReST+ [54]	32.5	44.7
w/ CReST++LA [41]	40.9	55.9
w/ CoSSL [13]	43.7	53.9
w/ TRAS [56]	46.2	54.1
w/ DASO [42]	44.5	53.7
w/ DASO+vanilla MWA	44.8	54.3
w/ DASO+IMWA	**45.9**	**55.3**
w/ ACR [55]	57.2	63.6
w/ ACR+vanilla MWA	57.5	64.4
w/ ACR+IMWA	**59.2**	**65.7**

TABLE V
PERFORMANCE OF OUR IMWA APPLYING TO UNBIASED TEACHER [34], SOFT TEACHER AND [59] AND DENSE TEACHER [69] ON SSOD BENCHMARK, *i.e.*, MS-COCO, IN TERMS OF $AP_{50:95}$. *UBT.* IS THE ABBREVIATION FOR UNBIASED TEACHER; *ST.* IS THE ABBREVIATION FOR SOFT TEACHER; *DT.* IS THE ABBREVIATION FOR DENSE TEACHER.

Methods	1%	2%	5%	10%
STAC [46]	13.97	18.25	24.38	28.64
Instant Teaching [70]	18.05	22.45	26.75	30.40
ISMT [62]	18.88	22.43	26.37	30.52
Humble Teacher [50]	16.96	21.72	27.70	31.61
Li *et al*. [27]	19.02	23.34	28.40	32.23
UBT. [34]	19.61	24.13	27.47	31.23
UBT.+vanilla MWA	19.97	24.55	27.59	31.64
***UBT.*+IMWA**	**20.46**	**24.56**	**28.21**	**32.04**
ST. [59]	20.57	24.85	30.17	33.66
ST.+vanilla MWA	20.74	25.09	30.42	34.01
***ST.*+IMWA**	**21.60**	**25.49**	**30.93**	**34.54**
DT. [69]	18.33	25.05	30.91	34.51
DT.+vanilla MWA	18.78	25.32	31.34	34.69
***DT.*+IMWA**	**19.06**	**27.12**	**32.43**	**35.87**

- **PASCAL VOC:** PASCAL VOC 2007 and 2012 datasets [12] contain 9,963 labeled images and 22,531 labeled images respectively including 20 categories. Following [34], [59], [69], we treat the `VOC 07 trainval set` as the labeled set and the `VOC 12 trainval set` as the unlabeled set. `VOC 07 test set` is viewed as the test set to evaluate our method.

D. Main Results

Here we report the experimental results for three class-imbalanced learning tasks.

1) Comparison on Class-Imbalanced Image Classification (CIIC): Tab. I shows the comparison results with previous state-of-the-art (SoTA) methods, where "+IMWA" means performing experiments with our IMWA. According to Tab. I, BCL+IMWA outperforms BCL **1.3%**, **0.7%** and **0.9%** in terms of Top-1 Acc on ImageNet-LT, iNaturalist 2018 and Places-LT respectively. LiVT+IMWA outperforms LiVT **0.8%**, **0.8%** and **1.6%** in terms of Top-1 Acc on ImageNet-LT, iNaturalist 2018 and Places-LT respectively. Especially, LiVT+IMWA achieves the state-of-the-art performance on ImageNet-LT and iNaturalist 2018 benchmarks. Note that BCL employs a CNN backbone and LiVT employs a ViT [11] backbone, which implies that our IMWA is effective for different types of backbones. Fig. 3 shows the comparison results with vanilla MWA and EMA for CIIC task on three benchmarks. For comparison on ImageNet, BCL and LiVT with vanilla MWA only improve about 0.2% in terms of Top1-Acc, while IMWA obtains **1.8%** and **1.4%** improvements respectively. *Notice that vanilla MWA has the same total training iterations with IMWA*, thus our IMWA can achieve higher improvement than vanilla MWA does without extra computation cost. We also observe that our IMWA shows comparable improvements with EMA (1.3% vs. 1.1% for BCL, 0.8% vs. 1.1% for LiVT), while performing both IMWA with EMA results in further performance improvement (**1.8%** for BCL and **1.4%** for LiVT), which implies that EMA and our IMWA complement each other. We can also observe the similar comparison results on iNat-18 and Places-LT datasets.

TABLE VI
PERFORMANCE OF OUR IMWA APPLYING TO UNBIASED TEACHER [34], SOFT TEACHERAND [59] AND DENSE TEACHER [69] ON SSOD BENCHMARK, *i.e.*, PASCAL VOC. *UBT.* IS THE ABBREVIATION FOR UNBIASED TEACHER; *ST.* IS THE ABBREVIATION FOR SOFT TEACHER; *DT.* IS THE ABBREVIATION FOR DENSE TEACHER.

Methods	AP_{50}	$AP_{50:95}$
UBT. [34]	81.71	54.09
UBT.+vanilla MWA	81.76	54.13
***UBT.*+IMWA**	**82.49**	**55.88**
ST. [59]	84.34	52.51
ST.+vanilla MWA	84.29	52.57
***ST.*+IMWA**	**85.17**	**52.71**
DT. [69]	80.27	56.73
DT.+vanilla MWA	80.29	56.81
***DT.*+IMWA**	**81.15**	**56.99**

TABLE VII
ABLATION STUDY OF DIFFERENT VALUE OF E. EXPERIMENT IS CONDUCTED BY ACR ON CIIC BENCHMARK, *i.e.*, IMAGENET-127 UNDER IMAGE SIZE TO 32×32.

Method	E	Top1 Acc
ACR+IMWA	-	57.2
	1	57.5
	2	57.7
	5	58.7
	10	**59.0**
	20	**59.2**
	50	**59.0**
	100	**59.1**
	200	**59.1**
	125000	57.7
	250000	57.6

2) Comparison on Class-Imbalanced Semi-Supervised Image Classification (CISSIC) and on Semi-Supervised Object Detection (SSOD): Similar to CIIC, we apply IMWA to the CISSIC and SSOD tasks. Tab. II,Tab. III,Tab. IV, Fig. 4 and 5 show the extensive experimental comparison results in terms of Top-1 Acc on CISSIC tasks; Tab. V to Tab. VI show the extensive experimental comparison results in terms of $AP_{50:95}$ and AP_{50} on SSOD tasks. Let's take the experimental results on the SSOD task as an example. According to these results, we analyze the effectiveness of our IMWA over the baseline methods in four-fold as follows:

1) Our IMWA achieves consistent improvements for different proportion of labeled set. Let's take the results of Dense Teacher for an example, as shown in Tab. V. Over each proportion of labeled setting (1%,2%,5% and 10% labeled ratio), Dense Teacher + IMWA achieve **0.73%** to **2.52%** $AP_{50:95}$ improvements consistently.
2) Our method achieves stable improvements for various baseline methods. As shown in Tab. VI, with respect to their corresponding baseline methods, our methods obtain **0.73%** to **1.03%** $AP_{50:95}$ improvements on 1% labeled images, **0.43%** to **2.07%** $AP_{50:95}$ on 2% labeled images, **0.74%** to **2.52%** $AP_{50:95}$ on 5% labeled images, **0.81%** to **1.36%** $AP_{50:95}$ on 10% labeled images.
3) Our method achieves stable improvements for different datasets. Tab. VI shows the experimental results on PASCAL VOC benchmark. Our methods based on three baseline achieve similar margin of improvements with that of MS-COCO.
4) Our method achieves stable outperforming comparing with the vanilla MWA. Extensive comparison from Tab. V and Tab. VI show that the improvement brought by vanilla MWA is significantly lower than that of our IMWA, which indicates that our IMWA is superior to vanilla MWA.

Similar conclusions can be drawn from the experimental results on CISSIC in Tab. II, III, IV and Fig. 4, 5 as well. Based on the above analysis, our IMWA can bring stable improvement over various class-imbalanced learning tasks and evaluation benchmark. Moreover, from the point of view of implementation, IMWA can be easily adapted into any existing CISSIC and SSOD without modifying models' structure or adding complicated tricks, which shows the sufficient efficiency of IMWA.

E. Ablation Studies

In this section we conduct ablation studies to explore the effect of M and E, which are the hyper-parameters of our IMWA.

1) Effect of the number of models M: M is a hyper-parameters of our IMWA, which presents the number of models involving training. Intuitively, the larger the value of M, the better the IMWA effect. The existing MWA appoarch [44], [57] also propose that increasing the number of models can enhance performance. However, as M increases, the computational cost of the whole training process also increases. Therefore, it is essential to search a suitable M to balance the performance and computational cost. Here we conduct ablation study by tuning the M from 2 to 6 for LiVT+IMWA on three benchmarks, and the results are shown in Fig. 6. $M = 1$ represents the setting without our IMWA. When $M = 2$ or $M = 3$, the performance of IMWA are increase obviously. As the M proceed to increase, there are no significant improvement in terms of performance. Note that a large M will aggravate the both cost of computation and memory. Thus, we suppose $\boldsymbol{M = 2}$ or $\boldsymbol{M = 3}$ are trade-off decisions in implementation between performance and efficiency.

TABLE VIII
GPU MEMORY COST FOR UNBIASED TEACHER+IMWA UNDER DIFFERENT M ON **SSOD BENCHMARK**, *i.e.*, **PASCAL VOC**.

Method	M	GPU memory cost / GB
Unbiased Teacher+IMWA	1	6.63
	2	7.35
	3	8.06
	4	8.76
	5	9.45
	6	10.47

2) Effect of the number of episode E: E represents the number of episodes in our IMWA. Larger value of E means that the average operation between models will be more frequent. Intuitively, the more frequent the average operation, the higher the performance will be obtained, as each average operation could potentially lead to a performance improvement. However, when E increases, the training iterations of each episode will be reduced, which will lead to the low diversity among the models to be averaged. Existing work [44] has proved that the effect of MWA is positively correlated with the differences between models. Therefore, we conduct ablation study to explore the best choice of E to balance the average frequency and models' diversity. We perform ACR+IMWA on ImageNet-127 under E from 1 to 250000, where 250000 is the default total training iteration of ACR [55]. Tab. VII shows the ablation results of various E. The first row represent the ACR w/o IMWA setting and the second row ($E = 1$) represents the vanilla MWA approach. The experimental results show that too large or too small E lead to slight improvement compared with baseline (w/o IMWA setting). While E is in the range of **5** to **200**, the improvement of performance is more noticeable. The above results indicate that there is relatively flexible room for tunning E, only necessary to avoid extremely large or extremely small values.

F. Discussion

In this section, we further give some discussion about our IMWA.

1) Discussion about Memory Consumption: Due to training multiple models simultaneously in IMWA, it inevitably introduces additional computational overhead like vanilla MWA. To save the GPU memory cost during training, in our implementation of IMWA, we equivalently transfer the "parallel

TABLE IX
COMPARISON OF IMWA WITH BASELINE METHODS TRAINED WITH $2 \times T$ ITERATIONS ON **SSOD BENCHMARK**, *i.e.*, **PASCAL VOC**. EXPERIMENTS ARE CONDUCTED BY UNBIASED TEACHER, SOFT TEACHER AND DENSE TEACHER.

Methods	2×	IMWA	AP_{50}
Unbiasd Teacher			81.71
	✓		81.74(+0.03)
		✓	**82.49 (+0.78)**
Soft Teacher			84.34
	✓		84.41(+0.07)
		✓	**85.17 (+0.83)**
Dense Teacher			80.27
	✓		80.32(+0.05)
		✓	**81.15 (+0.88)**

training" into a "serial training" manner to avoid the large cost of GPU memory. Specifically, in one episode, we can train each model $\boldsymbol{f}$ sequentially, and then save its trained weights into a dictionary (a data structure of python). After all the M models are well trained in this episode, we average their weights and then start the next episode. By this way, the GPU memory cost from simultaneous forward and backward propagation of M pairs detectors can be avoided, while only the trained weights need to be stored in GPU memory. We take Unbiased Teacher+IMWA under different model number M as an example. Tabel. VIII shows the results of actual GPU memory cost. For adding more one pair of model (optimized model and EMA model), the corresponding memory cost will increase by $(10.47 - 6.63)/5 = 0.76$ GB on average for Unbiased Teacher, *which are acceptable GPU memory costs for training them on each RTX 2080Ti (about 11 GB memory)*.

2) Comparison with Baseline under Longer Training Schedule: In our IMWA, M models are trained in parallel, each of which is trained in total of T iterations. Therefore, the whole training iterations of IMWA is $M \times T$, which is about M times more than that of baseline method (only T iterations). For a fair comparison, we train the baseline methods with the same training iterations as IMWA (*e.g.*$M \times T$ iterations. Here we choose to conduct experiments on SSOD task as an example. Specifically, we train Unbiased Teacher, Soft Teacher and Dense Teacher for $M \times T$ iterations respectively, and compare with w/ IMWA under $M = 2$ for fair comparison. Table IX shows the experimental results. Simply training the baseline method with $M\times$ iterations shows less improvements, while our IMWA obtains more obvious improvements. *From this point of view, training a SSOD method in IMWA pipeline will benefit more than simply training baseline method with more iterations*.

3) Discussion about the imbalance ratio: In this section we aim to exploring how the imbalance ratio of training set impact the effectiveness of our IMWA. We conduct experiments on CIFAR-10-LT with WideResNet-28 for CIIC task under various imbalance ratio $\gamma \in \{1, 5, 10, 20, 50, 100, 200\}$. In particular, the experiment with $\gamma = 1$ is equivalent to a balanced setting. From the Tab. X, we can observe that the

TABLE X
IMPACT OF DATASET IMBALANCE RATIO r FOR IMWA. EXPERIMENTS ARE CONDUCTED BY WIDERESNET-28 USING VANILLA SUPERVISED MANNER ON **CIIC BENCHMARK**, *i.e.*, **CIFAR-10-LT**.

Imbalance ratio	w/o IMWA	w/ IMWA
1	84.2	84.4 (+0.2)
5	72.5	73.4 (+0.9)
10	61.5	63.8 (+**2.3**)
20	53.5	54.5 (+1.0)
50	42.5	44.0 (+1.5)
100	36.8	37.8 (+1.0)
200	33.1	33.4 (+0.3)

performance improvement are not significant when the γ is too small or too large, which suggests that our IMWA is better suited for scenarios with class imbalance but not severe. For example, when $\gamma = 10$, IMWA brings absolutely **2.3%** improvement in terms of Top 1 Accuracy. We will explore how to enhance the performance of IMWA in scenarios with either class balance or severe class imbalance in our future work.

4) Comparison with concurrent works: We have surveyed two very recent work [19], [64] whose idea are similar with our IMWA. In general, there are three differences between our IMWA and these works. First, we evaluate the effectiveness of IMWA on various class-imbalanced learning task, with analyzing the reason of performing better in a class-imbalanced setting. While these concurrent only evaluate their methods on standard image classification task or out-of-distribution scenario. Second, we further explore the synergistic collaboration of our IMWA on EMA-based methods and empirically demonstrate that IMWA and EMA can be mutual promotion, which is something that these concurrent works have not done. Third, in our implementation, the improvement can be obtained without adopting different data augmentation or enumerate various hyper-parameters (*e.g.*learning rate) for each individual model, which these concurrent works need. In summary, both IMWA and these concurrent works demonstrate the effectiveness of MWA with an "iterative" manner. However, in our work, we further illustrate that IMWA can be applied to a broader range of scenarios and mutual promotion with other MWA technique.

V. CONCLUSION

In this paper, we empirically explore the current model weight averaging (MWA). Specifically, the vanilla MWA can benefit the class-imbalanced learning, and performing model averaging in the early epochs of training yields a greater performance improvement than doing that in later epochs. Based on this discovery, we propose a novel MWA technique named IMWA for class-imbalanced learning tasks, which present the vanilla MWA for multiple times during the training process to further obtain performance improvement. We apply IMWA on several existing SOTA methods of three class-imbalanced image recognize tasks to evaluate its effectiveness. In addition, we also explore the synergistic collaboration of IMWA and EMA. Extensive experiments results on various benchmarks

demonstrate that IMWA can achieve larger performance improvements for these SOTA methods than vanilla MWA does, while IMWA and EMA can complement each other.

REFERENCES

[1] Shaden Alshammari, Yu-Xiong Wang, Deva Ramanan, and Shu Kong. Long-tailed recognition via weight balancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6897–6907, 2022.

[2] Jiarui Cai, Yizhou Wang, and Jenq-Neng Hwang. Ace: Ally complementary experts for solving long-tailed recognition in one-shot. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 112–121, 2021.

[3] Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems*, 34:22405–22418, 2021.

[4] Binbin Chen, Weijie Chen, Shicai Yang, Yunyi Xuan, Jie Song, Di Xie, Shiliang Pu, Mingli Song, and Yueting Zhuang. Label matching semi-supervised object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14381–14390, 2022.

[5] Binghui Chen, Pengyu Li, Xiang Chen, Biao Wang, Lei Zhang, and Xian-Sheng Hua. Dense learning based semi-supervised object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4815–4824, 2022.

[6] 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.

[7] Jiequan Cui, Zhisheng Zhong, Shu Liu, Bei Yu, and Jiaya Jia. Parametric contrastive learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 715–724, 2021.

[8] 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.

[9] 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.

[10] Bowen Dong, Pan Zhou, Shuicheng Yan, and Wangmeng Zuo. Lpt: Long-tailed prompt tuning for image classification. *arXiv preprint arXiv:2210.01033*, 2022.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[12] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010.

[13] 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.

[14] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pages 3259–3269. PMLR, 2020.

[15] 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 (CVPR)*, June 2016.

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

[17] Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights. *arXiv preprint arXiv:2208.05592*, 2022.

[18] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

[19] Samyak Jain, Sravanti Addepalli, Pawan Kumar Sahu, Priyam Dey, and R Venkatesh Babu. Dart: Diversify-aggregate-repeat training improves generalization of neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16048–16059, 2023.

[20] 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.

[21] JongMok Kim, JooYoung Jang, Seunghyeon Seo, Jisoo Jeong, Jongkeun Na, and Nojun Kwak. Mum: Mix image tiles and unmix feature tiles for semi-supervised object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14512–14521, 2022.

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

[23] 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.

[24] Aoxue Li, Peng Yuan, and Zhenguo Li. Semi-supervised object detection via multi-instance alignment with global class prototypes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9809–9818, 2022.

[25] 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.

[26] Gang Li, Xiang Li, Yujie Wang, Wu Yichao, Ding Liang, and Shanshan Zhang. Dtg-ssod: Dense teacher guidance for semi-supervised object detection. *Advances in neural information processing systems*, 35:8840–8852, 2022.

[27] Hengduo Li, Zuxuan Wu, Abhinav Shrivastava, and Larry S Davis. Rethinking pseudo labels for semi-supervised object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1314–1322, 2022.

[28] 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.

[29] Mengke Li, Yiu-ming Cheung, and Yang Lu. Long-tailed visual recognition via gaussian clouded logit adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6929–6938, 2022.

[30] Tianhong Li, Peng Cao, Yuan Yuan, Lijie Fan, Yuzhe Yang, Rogerio S Feris, Piotr Indyk, and Dina Katabi. Targeted supervised contrastive learning for long-tailed recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6918–6928, 2022.

[31] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[32] Bo Liu, Haoxiang Li, Hao Kang, Gang Hua, and Nuno Vasconcelos. Breadcrumbs: Adversarial class-balanced sampling for long-tailed recognition. In *European Conference on Computer Vision*, pages 637–653. Springer, 2022.

[33] Hao Liu, Bin Chen, Bo Wang, Chunpeng Wu, Feng Dai, and Peng Wu. Cycle self-training for semi-supervised object detection with distribution consistency reweighting. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6569–6578, 2022.

[34] 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. *arXiv preprint arXiv:2102.09480*, 2021.

[35] Yen-Cheng Liu, Chih-Yao Ma, and Zsolt Kira. Unbiased teacher v2: Semi-supervised object detection for anchor-free and anchor-based detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9819–9828, 2022.

[36] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2537–2546, 2019.

[37] Alexander Long, Wei Yin, Thalaiyasingam Ajanthan, Vu Nguyen, Pulak Purkait, Ravi Garg, Alan Blair, Chunhua Shen, and Anton van den Hengel. Retrieval augmented classification for long-tail visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6959–6969, 2022.

[38] Teli Ma, Shijie Geng, Mengmeng Wang, Jing Shao, Jiasen Lu, Hongsheng Li, Peng Gao, and Yu Qiao. A simple long-tailed recognition baseline via vision-language model. *arXiv preprint arXiv:2111.14745*, 2021.

[39] Michael S Matena and Colin A Raffel. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716, 2022.

[40] ML Menéndez, JA Pardo, L Pardo, and MC Pardo. The jensen-shannon divergence. *Journal of the Franklin Institute*, 334(2):307–318, 1997.

[41] 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.

[42] 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.

[43] Seulki Park, Jongin Lim, Younghan Jeon, and Jin Young Choi. Influence-balanced loss for imbalanced visual classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 735–744, 2021.

[44] Alexandre Rame, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Patrick Gallinari, and Matthieu Cord. Diverse weight averaging for out-of-distribution generalization. *arXiv preprint arXiv:2205.09739*, 2022.

[45] 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.

[46] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020.

[47] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[48] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11662–11671, 2020.

[49] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. *Advances in Neural Information Processing Systems*, 33:1513–1524, 2020.

[50] Yihe Tang, Weifeng Chen, Yijun Luo, and Yuting Zhang. Humble teachers teach better students for semi-supervised object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3132–3141, 2021.

[51] Changyao Tian, Wenhai Wang, Xizhou Zhu, Jifeng Dai, and Yu Qiao. Vl-ltr: Learning class-wise visual-linguistic representation for long-tailed visual recognition. In *European Conference on Computer Vision*, pages 73–91. Springer, 2022.

[52] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018.

[53] Hualiang Wang, Siming Fu, Xiaoxuan He, Hangxiang Fang, Zuozhu Liu, and Haoji Hu. Towards calibrated hyper-sphere representation via distribution overlap coefficient for long-tailed learning. In *European Conference on Computer Vision*, pages 179–196. Springer, 2022.

[54] 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.

[55] 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.

[56] 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.

[57] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pages 23965–23998. PMLR, 2022.

[58] Liuyu Xiang, Guiguang Ding, and Jungong Han. Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 247–263. Springer, 2020.

[59] Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu. End-to-end semi-supervised object detection with soft teacher. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3060–3069, 2021.

[60] Yue Xu, Yong-Lu Li, Jiefeng Li, and Cewu Lu. Constructing balance from imbalance for long-tailed image recognition. In *European Conference on Computer Vision*, pages 38–56. Springer, 2022.

[61] Zhengzhuo Xu, Ruikang Liu, Shuo Yang, Zenghao Chai, and Chun Yuan. Learning imbalanced data with vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15793–15803, 2023.

[62] Qize Yang, Xihan Wei, Biao Wang, Xian-Sheng Hua, and Lei Zhang. Interactive self-training with mean teachers for semi-supervised object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5941–5950, 2021.

[63] Sihao Yu, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Zizhen Wang, and Xueqi Cheng. A re-balancing strategy for class-imbalanced classification based on instance difficulty. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 70–79, 2022.

[64] Jiangtao Zhang, Shunyu Liu, Jie Song, Tongtian Zhu, Zhengqi Xu, and Mingli Song. Lookaround optimizer: k steps around, 1 step average. *arXiv preprint arXiv:2306.07684*, 2023.

[65] Yifan Zhang, Bryan Hooi, Lanqing Hong, and Jiashi Feng. Test-agnostic long-tailed recognition by test-time aggregating diverse experts with self-supervision. *arXiv e-prints*, pages arXiv–2107, 2021.

[66] 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.

[67] 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.

[68] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

[69] Hongyu Zhou, Zheng Ge, Songtao Liu, Weixin Mao, Zeming Li, Haiyan Yu, and Jian Sun. Dense teacher: Dense pseudo-labels for semi-supervised object detection. In *European Conference on Computer Vision*, pages 35–50. Springer, 2022.

[70] Qiang Zhou, Chaohui Yu, Zhibin Wang, Qi Qian, and Hao Li. Instant-teaching: An end-to-end semi-supervised object detection framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4081–4090, 2021.

[71] Jianggang Zhu, Zheng Wang, Jingjing Chen, Yi-Ping Phoebe Chen, and Yu-Gang Jiang. Balanced contrastive learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6908–6917, 2022.

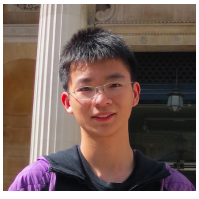
VI. Biography Section

Zitong Huang is currently pursuing the Ph.D. degree with Harbin Institute of Technology, Harbin, China. His research interests include computer vision, deep learning, continual learning and object detection.

Ze chen is currently employed at Megvii Technology Limited, having graduated with a master's degree from Shanghai Jiao Tong University. His research interests encompass general object detection, the practical applications of deep learning models, and generative modeling.

Bowen Dong is the PhD Candidate of Harbin Institute of Technology and Hong Kong Polytechnic University, and his supervisors are Professor Wangmeng Zuo and Professor Lei Zhang. His main research directions are weakly supervised learning, few-shot learning and transfer learning. He has published several papers in top conferences such as ICCV, ECCV, CVPR, and ICLR, and has served as a reviewer for conferences or journals such as CVPR, ICCV, ECCV, and TIP.

Chaoqi Liang is currently pursuing a Bachelor of Science degree at Harbin Institute of Technology in Harbin, China. His research interests include computer vision, large language models, semi-supervised learning, and computational biology.



Erjin Zhou is the Director of the research group at Megvii Research Institute, responsible for building and leading the team to conduct research on facial and human body detection and recognition, portrait generation, general object detection, and action recognition technologies. Erjin Zhou also leads the team in the research and development of algorithm production tools. His research achievements have been applied to Megvii's cloud-based identity verification solution, as well as industry solutions for intelligent building access, smart phone AI solutions, and financial industry identity verification.

Wangmeng Zuo received the Ph.D. degree from the Harbin Institute of Technology in 2007. He is currently a Professor in the School of Computer Science and Technology, Harbin Institute of Technology. His research interests include image enhancement and restoration, image and face editing, object detection, visual tracking, and image classification. He has published over 100 papers in top tier journals and conferences. His publications have been cited more than 55,000 times. He also serves as Associate Editors for IEEE T-PAMI and IEEE T-IP.