Supplementary Materials for DASO: Distribution-Aware Semantics-Oriented Pseudo-Label for Imbalanced Semi-Supervised Learning

Contents

A. Notations **2**

B. Algorithm **3**

C. Detailed Experimental Setup **4**
C.1. Benchmarks . . . 4
C.2. Training Details . . . 4
C.3. Implementation Details . . . 4

D. Additional Experiments **7**
D.1. Comprehensive Comparison with More Baselines . . . 7
D.2. DASO with Label Re-Balancing when $\gamma_l \neq \gamma_u$ . . . 8
D.3. Comparison based on ReMixMatch . . . 8
D.4. Results on Test-Time Logit Adjustment . . . 9
D.5. More Ablation Study . . . 9

E. Detailed Analysis **9**
E.1. Recall and Precision Analysis . . . 9
E.1.1 Detailed comparison for linear pseudo-label and semantic pseudo-label methods . . . 9
E.1.2 DASO with class distribution mismatch on traditional SSL learner . . . 10
E.2. Confusion Matrix on Test Data . . . 11
E.3. Train Curves for Recall and Accuracy . . . 11
E.4. Further Comparison of Feature Representations . . . 12
E.5. Confidence Analysis from Out-of-class Examples . . . 12

F. Overall Framework **13**

A. Notations

In this section, we clarify all the notations with corresponding descriptions introduced in this work.

Notation	Description
DASO	Distribution-Aware Semantic-Oriented (Pseudo-label)
SSL	Semi-Supervised Learning.
K	The number of classes in the labeled data.
$\mathcal{X},\ \mathcal{U}$	Labeled data and unlabeled data.
$N,\ M$	Total number of examples in labeled data and unlabeled data.
N_k, M_k	Number of examples in class k for labeled data and unlabeled data.
$\gamma_l,\ \gamma_u$	Imbalance ratio for labeled data and unlabeled data.
$\hat{m}$	Empirical pseudo-label distribution in probability form; $\hat{m} \in [0,1]^K$.
$\sigma(\cdot)$	Softmax activation.
$\mathcal{H}(y,\ p)$	Cross-entropy between the target y and prediction p.
$\text{sim}(\cdot,\cdot)$	Cosine similarity.
f	A classification model; a feature encoder f_θ^{enc} followed by a linear classifier f_ϕ^{cls}.
$f_{\theta'}^{\text{enc}}$	An EMA encoder (momentum encoder).
ρ	Decay ratio for the momentum encoder.
$\mathbf{Q}$	A dictionary of memory queue; $\{Q_k\}_{k=1}^K$.
L	The maximum queue size for the *balanced* memory queue.
$\mathbf{C}$	A set of class prototypes; $\{c_k\}_{k=1}^K$.
T_{proto}	A temperature factor for the similarity-based classifier.
T_{dist}	A temperature factor for the empirical pseudo-label distribution.
$\hat{p},\ q^{(w)}$ or $\hat{q}$	A linear pseudo-label and semantic pseudo-label.
υ	Class-specific mixup factor for the linear and semantic pseudo-label; $\{\upsilon_k\}_{k=1}^K$.
$\hat{p}'$	A blended pseudo-label.
$\text{PseudoLabel}(\cdot)$	Pseudo-labeler specified by an SSL algorithm.
$\Phi_u(\cdot,\cdot)$	A regularizer for $\mathcal{U}$, specified by an SSL algorithm.
λ_u	The loss weight for $\mathcal{L}_u$.
$\mathcal{L}_{\text{align}}$	Semantic alignment loss.
λ_{align}	The loss weight for $\mathcal{L}_{\text{align}}$.
P	Pre-train steps for applying pseudo-label blending and $\mathcal{L}_{\text{align}}$.
$\mathcal{A}_w$	A set of weak augmentations; horizontal *flip* and/or *crop*.
$\mathcal{A}_s$	A set of strong augmentations; RandAugment [5] followed by Cutout [8].
μ	Unlabeled batch ratio; multiplied to the labeled batch size B.

Table 1. Notations and their descriptions used throughout this work.

B. Algorithm

Algorithm 1 summarizes the blending procedure for the linear and semantic pseudo-labels based on the empirical pseudo-label distribution, and Algorithm 2 represents the whole DASO framework built upon a typical SSL algorithm where the regularizer for the SSL algorithm corresponds to Φ_u.

Algorithm 1 Distribution-aware pseudo-label blending, $\hat{p}' \leftarrow \text{Blend}\left(\hat{p}, \hat{q}, T_{\text{dist}}\right)$.

Input: Linear pseudo-label $\hat{p} \in [0,1]^K$, semantic pseudo-label $\hat{q} \in [0,1]^K$,
Temperature factor for the pseudo-label distribution T_{dist}.
Require: Empirical pseudo-label distribution $\hat{m} = \{\hat{m}_k\}_{k=1}^K$.
Output: Blended pseudo-label $\hat{p}' \in [0,1]^K$.

for $k = 1$ **to** K **do**
  $\upsilon_k \leftarrow \hat{m}_k^{1/T_{\text{dist}}}$ {Temperature scaling for empirical pseudo-label distribution.}
  $\upsilon_k \leftarrow \upsilon_k / \max_k \upsilon_k$ {Normalization for blending.}
end for
$k' \leftarrow \text{argmax}_k \hat{p}_k$ {Class prediction of the linear pseudo-label.}
$\hat{p}' \leftarrow (1 - \upsilon_{k'})\,\hat{p} + \upsilon_{k'}\hat{q}$ {Pseudo-label blending.}

Algorithm 2 Distribution-Aware Semantic-Oriented (DASO) Pseudo-label framework.

Input: A batch of labeled data $\mathcal{X}_B = \{(x_b, y_b)\}_{b=1}^B$ and unlabeled data $\mathcal{U}_B = \{u_b\}_{b=1}^{\mu B}$.
Network for feature encoder f_θ^{enc}, momentum encoder $f_{\theta'}^{\text{enc}}$, and linear classifier f_ϕ^{cls}.
Dictionary of memory queue $\mathbf{Q} = \{Q_k\}_{k=1}^K$, Momentum decay ratio ρ.
Maximum queue size L, temperature factor for the similarity-based classifier T_{proto},
Pre-train steps for pseudo-label blending P, current training step t.

Require: A set of weak augmentations $\mathcal{A}_w$ and strong augmentations $\mathcal{A}_s$.

{**Balanced Prototype Generation.**}
Enqueue $z^{(l)}$ into Q_k, where $z^{(l)} = f_{\theta'}^{\text{enc}}(x)$ and $k \leftarrow y, \quad \forall\,(x,y) \in \mathcal{X}_B$.
Dequeue the earliest elements from Q_k *s.t.* $|Q_k| = L, \quad \forall\, k \in \{1,\dots,K\}$.
$c_k \leftarrow \frac{1}{|Q_k|}\sum_{z_i \in Q_k} z_i, \ \forall\, k \in \{1,\dots,K\}$, {A set of balanced prototypes $\mathbf{C} = \{c_k\}_{k=1}^K$.}

{**Pseudo-label generation.**}
for u **in** $\mathcal{U}_B$ **do**
  $z^{(w)} \leftarrow f_\theta^{\text{enc}}\left(\mathcal{A}_w(u)\right), \quad z^{(s)} \leftarrow f_\theta^{\text{enc}}\left(\mathcal{A}_s(u)\right)$ {feature extraction}
  $\hat{p} \leftarrow \sigma\left(f_\phi^{\text{cls}}\left(z^{(w)}\right)\right), \quad q^{(w)} \leftarrow \sigma\left(\text{sim}\left(z^{(w)}, \mathbf{C}\right)/T_{\text{proto}}\right)$
  $\hat{p}' \leftarrow \text{Blend}\left(\hat{p}, q^{(w)}, T_{\text{dist}}\right)$ **if** $t \geq P$ **else** $\hat{p}$ {Blend pseudo-labels after P train steps.}
end for

{**Compute losses.**}
$\mathcal{L}_{\text{cls}} \leftarrow \mathbb{E}_{(x,y)\in\mathcal{X}_B}\left[\mathcal{H}\left(y, \sigma\left(f(x)\right)\right)\right]$
$\mathcal{L}_{\text{align}} \leftarrow \mathbb{E}_{u\in\mathcal{U}_B}\left[\mathbb{1}\left(t \geq P\right) \cdot \mathcal{H}\left(q^{(w)}, q^{(s)}\right)\right]$ where $q^{(s)} \leftarrow \sigma\left(\text{sim}\left(z^{(s)}, \mathbf{C}\right)/T_{\text{proto}}\right)$.
$\mathcal{L}_u \leftarrow \mathbb{E}_{u\in\mathcal{U}_B}\left[\Phi_u\left(\hat{p}', p^{(s)}\right)\right]$ where $p^{(s)} \leftarrow f_\phi^{\text{cls}}\left(z^{(s)}\right)$.
$\mathcal{L}_{\text{DASO}} \leftarrow \mathcal{L}_{\text{cls}} + \lambda_u\mathcal{L}_u + \lambda_{\text{align}}\mathcal{L}_{\text{align}}$

{**Update parameters.**}
Update θ and ϕ to minimize $\mathcal{L}_{\text{DASO}}$ via SGD optimizer.
$\theta' \leftarrow \rho\theta' + (1-\rho)\theta$ {Update the parameters of momentum encoder.}
$t \leftarrow t + 1$

C. Detailed Experimental Setup

C.1. Benchmarks

In this work, we evaluate both cases of (i) labeled data and unlabeled data shares the same class distribution (*e.g.*, $\gamma_l = \gamma_u$), and (ii) the class distribution of unlabeled data can be different from the labeled data in various degree (*e.g.*, $\gamma_l \neq \gamma_u$).

CIFAR-10 and CIFAR-100. CIFAR benchmarks [13] originally have the same number of examples per class; 5000 and 500 examples in 32×32 sized image for CIFAR-10 and CIFAR-100, respectively. We use the head class size N_1 and imbalance ratio of labels γ_l to craft the *synthetically long-tailed* variants across the level of imbalance and total amount of labels, following the protocol from [12]. The number of examples other than the head class is calculated by $N_k = N_1 \cdot \gamma_l^{-\frac{k-1}{K-1}}$ as proposed by [6]. Note that each N_k, the number of examples in class k is sorted in a descending order (*i.e.*, $N_1 \geq \cdots \geq N_K$). Similarly, the number of examples per class for the unlabeled data can be determined by: $M_k = M_1 \cdot \gamma_u^{-\frac{k-1}{K-1}}$ using the labels, and the true labels are thrown away before training. We call those variants as CIFAR10/100-LT, which consist of labeled and unlabeled splits. We measure the performance on the test data, which have $10k$ examples in total for both data.

STL-10. To generate STL10-LT: a *long-tailed* variant of STL-10 [4], we follow the same process as explained in above. Besides the $5k$ labeled examples, STL-10 contains additional $100k$ unlabeled examples from a similar but broader distribution compared to the labeled data. Since the information about the class distribution of the unlabeled data is not known, we only construct the imbalanced labeled data and use the whole $100k$ unlabeled examples for training.

Semi-Aves. We also consider Semi-Aves benchmark [20] for more realistic scenarios. *Semi-Aves* includes $1k$ species of birds sampled from the *iNaturalist-2018* [23] with *long-tailed* class distribution. Moreover, only 200 species are considered *in-class*, and the other 800 species correspond to the *out-of-class* (*i.e.*, novel, open-set) categories for the unlabeled data. For *in-class* examples, about $4k$ examples are labeled ($\mathcal{X}$), and the other $27k$ examples are unlabeled ($\mathcal{U}_{\text{in}}$). Note that the class distribution of labeled data does not match that of $\mathcal{U}_{\text{in}}$ ($\gamma_l \neq \gamma_u$), as illustrated in [20]. The *out-of-class* unlabeled data ($\mathcal{U}_{\text{out}}$) have $122k$ examples in total. *Semi-Aves* benchmark provides $2k$ images and $8k$ images (*i.e.*, 10 images and 40 images per class) for the validation and test data, respectively. We combine the labeled training data and validation data, $6k$ in total, for the labeled training data in our experiments, following [19]. As note, we do not make any distinction between $\mathcal{U}_{\text{in}}$ and $\mathcal{U}_{\text{out}}$ when learning on the whole unlabeled data ($\mathcal{U} = \mathcal{U}_{\text{in}} + \mathcal{U}_{\text{out}}$).

C.2. Training Details

CIFAR10/100-LT and STL10-LT. Following the training protocol in [12], we train a Wide ResNet-28-2 [25] with 1.5M parameters for $250k$ iterations. We set the batch size of the labeled data as 64, and the network is optimized via Nesterov SGD with momentum 0.9 and weight decay 5e-4. For the methods with using only labels, the base learning rate is set to 0.1 with linear warm-up applied during the first 2.5% of the total train steps, and it decays after 80% and 90% of the training phase by a factor of 100, respectively, following [3]. For SSL methods, we set the base learning rate as 0.03, which is fixed during the training. For the exponential moving average (EMA) network parameters for evaluation, the decay ratio ρ is set to 0.999. We further clarify the details for each method, such as hyper-parameters in Appendix C.3. We measure the performance every 500 iterations (*e.g.*, considered as 1 epoch), and report the median value in last 20 evaluations.

Semi-Aves. We train ResNet-34 [10] with 21.3M parameters pre-trained on ImageNet [7]. For the Supervised method, we train for 90 epochs of the labeled data, while we train 90 epochs of unlabeled data for SSL methods, using SGD optimizer with momentum 0.9. The base learning rate is set to 0.1 and 0.04 for the Supervised and SSL method each, with the linear warm-up for the first 5 epochs and it decays after 30 and 60 epochs, by a factor of 10. We set the labeled batch size as 256. All training images are randomly cropped and re-scaled to 224×224 size with random horizontal flip. The EMA decay ratio is $\rho = 0.9$. The hyper-parameters of the individual method is described in Appendix C.3.

C.3. Implementation Details

DASO. T_{dist}, for scaling the empirical pseudo-label distribution, is chosen out of $\{0.3, 0.5, 1.0, 1.5\}$. Specifically, for CIFAR10-LT, $T_{\text{dist}} = 1.5$ in case of $\gamma_l = \gamma_u$, while $T_{\text{dist}} = 0.3$ in the case of $\gamma_l \neq \gamma_u$. For the other hyper-parameters, $T_{\text{proto}} = 0.05$, $L = 256$, and $\lambda_{\text{align}} = 1$, which are kept unchanged during experiments. The ablation study for those parameters is provided in Appendix D.5. We start applying DASO with $\mathcal{L}_{\text{align}}$ after a few pre-training steps $P = 5000$ to avoid unconfident predictions in the early stage of training. For empirical pseudo-label distribution $\hat{m}$, we accumulate the class predictions of the final pseudo-labels $\hat{p}'$ every 100 iterations on CIFAR10/100-LT and STL10-LT. For Semi-Aves, we set $P = 20$ epochs and update $\hat{m}$ every epoch. For the EMA decay ratio ρ for prototype generation, we simply use the same

parameter of the one for evaluation. Table 2 summarizes the training details of DASO.

parameter	CIFAR10-LT	CIFAT100-LT	STL10-LT	Semi-Aves
lr	0.03			0.04
B	64			256
μ	2			5
SGD momentum	0.9			0.9
Nesterov	True			True
weight decay	5e-4			3e-4
L	256			256
ρ	0.999			0.9
T_{proto}	0.05			0.05
λ_{align}	1.0			1.0
P	5000 *steps*			20 *epochs*
T_{dist}	$\{1.5, 0.3\}$	0.3	0.3	0.5

Table 2. A complete list of training details for DASO framework.

Supverised. The only labeled data is trained via standard cross-entropy loss $\mathcal{H}$. The training protocol and hyper-parameters (total iterations, learning rate, optimizer, and etc.) are described in Appendix C.2.

Re-weighting with the Effective Number of Samples [6]. The per-class weights are applied to the cross-entropy loss based on the effective number of samples.

$$E_{N_k} = \frac{1-\beta^{N_k}}{1-\beta}, \tag{1}$$

where N_k corresponds to the number of samples in class k, and then the weight for class k is set to be proportional to the inverse of the effective number E_{N_k}. β is a hyper-parameter, which is set to 0.999 during the experiments.

LDAM-DRW [3]. Decision boundary of the classifier takes up more margin in rare classes, using LDAM loss:

$$\mathcal{L}_{LDAM} = -\log \frac{e^{z_{y_k}-\Delta_{y_k}}}{e^{z_{y_k}-\Delta_{y_k}} + \sum_{j\neq y_k} e^{z_j}}, \text{ where } \Delta_k \propto \frac{1}{N_k^{1/4}}. \tag{2}$$

Then it adopts deferred re-weighting scheme (DRW) to apply re-balancing algorithm in later stage of training. Following DRW scheme, we apply re-weighting objective Eq. (2) after 200k iterations.

cRT [11]. After training the entire network under imbalanced distribution, the classifier is re-trained with the parameters of the feature encoder fixed for a balanced objective. We first train a model with cross-entropy loss. In classifier re-training phase, we simply re-weight the cross-entropy loss with the weights based on the effective number of samples [6] for $100k$ iterations. The learning rate schedule under re-training phase is proportionally adjusted.

Logit Adjustment (LA) [16]. Logits are adjusted by enforcing a large margin for the minority classes compared to the majority ones in either two ways: *post-hoc adjustment* or *logit-adjusted* cross-entropy, based on the class frequency of labels. In this work, we adopt the latter strategy. Before measuring cross-entropy for the labeled data, each logit is adjusted by:

$$p_k \leftarrow p_k + \tau \log n_k, \tag{3}$$

where $p = f(x)$ and n_k denotes the class label frequency value in class k. $\tau = 1$ is a temperature scaling factor.

PseudoLabel [14]. The one-hot pseudo-label $\hat{p}$ from $p = f(u)$ regularizes the unlabeled example. Only the predictions with the highest probability value above a certain threshold τ contribute to the regularizer. We set τ to 0.95.

$$\Phi_u(\hat{p},\, p) = \mathbb{1}\left(\max_k p_k \geq \tau\right) \mathcal{H}\left(\hat{p},\, p\right), \tag{4}$$

where $\hat{p} = \text{OneHot}(\text{argmax}_k\, p_k)$. We set the loss weight $\lambda_u = 1$ and apply linear ramp-up with the ratio of 0.4; λ_u linearly increases starting from 0 and attains the maximum value ($\lambda_u = 1$) at 40% of the total iterations.

MeanTeacher [21]. The momentum encoder $f^{\text{EMA}} = f^{\text{cls}}_{\phi'} \circ f^{\text{enc}}_{\theta'}$ generates the target for the prediction of unlabeled data, where ϕ' and θ' are the momentum-updating network parameters of linear classifier and feature encoder, respectively.

$$\Phi_u(\hat{p},\, p) = \|\sigma(\hat{p}) - \sigma(p)\|^2\,, \text{ where } \hat{p} = f^{\text{EMA}}(\mathcal{A}_w(u)) \text{ and } p = f(\mathcal{A}_w(u)). \tag{5}$$

We set the EMA decay ratio $\rho = 0.999$. λ_u is set to 50, applying the linear ramp-up with the ratio of 0.4.

MixMatch [2]. Pseudo-label is produced from the multiple augmentations of the same image with entropy regularization. Then the model learns mixup [26] images and (pseudo-) labels over the whole labeled and unlabeled data. We use the number of augmentations as 2, temperature scaling factor as 0.5, and the sampling hyper-parameter for mixup regularization α as 0.5. We also apply linear ramp-up strategy for λ_u, where it attains its maximum value 100 with the ratio of 0.016.

ReMixMatch [1]. It adds up two techniques of *Augmentation Anchoring* and *Distribution Alignment* over MixMatch [2]. We use the advanced augmentation as RandAugment [5] followed by Cutout [8]. Considering the computational cost, we set the number of advanced augmentations as $\mu = 2$. For the others, we set the temperature scaling factor for pseudo-labels as 0.5, and α as 0.75. The weights for pre-mixup loss and rotation loss are both set to 0.5. For λ_u, the linear ramp-up ratio is set to 0.016 with $\lambda_u = 1.5$. We apply weak augmentations for convenience for the labeled data, instead of advanced augmentation.

FixMatch [18]. One-hot pseudo-labels are generated from weakly augmented images as the same with PseudoLabel [14], then they provide the targets for the predictions from strong augmentations of the same images to the cross-entropy loss $\mathcal{H}$:

$$\Phi_u(\hat{p},\, p^{(s)}) = \mathbb{1}\left(\max_k p_k^{(w)} \geq \tau\right) \mathcal{H}\left(\hat{p},\, p^{(s)}\right), \tag{6}$$

where $\hat{p} = \text{OneHot}\left(\text{argmax}_k\, p_k^{(w)}\right)$ with $p^{(w)} = f\left(\mathcal{A}_w(u)\right)$ and $p^{(s)} = f\left(\mathcal{A}_s(u)\right)$. We use RandAugment [5] for the advanced augmentation. For fair comparisons to ReMixMatch [1], we use the unlabeled batch ratio μ as 2. For the other hyper-parameters, λ_u is set to 1 without applying linear ramp-up strategy.

USADTM [9]. It combines *unsupervised semantic aggregation* (USA); a clustering objective in unlabeled data and *deformable template matching* (DTM); assigning a semantic pseudo-label to each unlabeled example solely from feature-space. The semantic pseudo-label is determined by the agreement of two different distance measure from a sample to each class prototypes constructed from the labeled data. In our experiments, we use the loss weight for the mutual information loss $\alpha = 0.1$ and $\tau = 0.85$ for the confidence threshold, following [9]. We note that [9] keeps some confident unlabeled examples to treat them as labeled examples to enforce cross-entropy loss due to the limited labels (*i.e.*, 4 labels per class). This would also help generally in *imbalanced* SSL, but we do not adopt this strategy in our experiments in order to fairly comparing with other SSL methods focusing on the aspect of *pseudo-labeling* method.

BOSS [17]. This originally proposes to apply three techniques altogether on FixMatch [18] to achieve state-of-the-art performance on CIFAR-10 benchmark under one label per class: *prototype (single-example per class) refining*, *pseudo-label re-balancing*, and *self-training iterations*. We only adopt *pseudo-label re-balancing* method from the original paper for fairly comparing under *imbalanced* SSL. *Pseudo-label re-balancing* includes adjusting loss weights and confidence thresholds based on the class distribution of predicted pseudo-labels on top of the FixMatch loss:

$$\Phi_u(\hat{p},\, p^{(s)}) = \mathbb{1}\left(\max_k p_k^{(w)} \geq \tau_k\right) \frac{1}{Z \cdot \hat{c}_k} \mathcal{H}\left(\hat{p},\, p^{(s)}\right), \tag{7}$$

where τ_k is the class-dependent confidence threshold defined as:

$$\tau_k = \tau - \Delta \cdot \left(1 - \frac{\hat{c}_k}{\max_k \hat{c}_k}\right), \tag{8}$$

and $\hat{c}_k$ is the number of predicted pseudo-labels in the current batch for class k. We fix $\Delta = 0.25$ during the experiments. Note that the scale of Φ_u is adjusted by a factor of Z to consistently maintain the relative scale of λ_u.

DARP [12]. The class distribution of the predicted pseudo-labels is explicitly adjusted to the *given* class priors via solving a convex optimization problem. In our experiments, we use the class prior as the class label frequency in case of $\gamma_l = \gamma_u$ for CIFAR10-LT and CIFAR100-LT, and in case of Semi-Aves benchmark. In other cases, *i.e.*, $\gamma_l \neq \gamma_u$, we estimate the distribution of the unlabeled data (*e.g.*, M_k) using held-out validation set, following [12]. We start applying DARP at $100k$ iterations of training with refining pseudo-labels every 10 steps. We use $\alpha = 2.0$ for removing the noisy entries.

Algorithm	Method type			CIFAR10-LT $\gamma = \gamma_l = \gamma_u = 100$		CIFAR100-LT $\gamma = \gamma_l = \gamma_u = 10$		STL10-LT $\gamma_l = 10$, γ_u: *unknown*	
	SSL	LB	PB	$N_1 = 500$ $M_1 = 4000$	$N_1 = 1500$ $M_1 = 3000$	$N_1 = 50$ $M_1 = 400$	$N_1 = 150$ $M_1 = 300$	$N_1 = 150$ $M = 100k$	$N_1 = 450$ $M = 100k$
Supervised				47.3 ±0.95	61.9 ±0.41	29.6 ±0.57	46.9 ±0.22	40.2 ±1.80	60.4 ±1.91
w/ LDAM-DRW [3]		✓		50.1 ±1.55	65.7 ±1.49	28.4 ±0.32	46.2 ±0.46	41.8 ±3.05	62.1 ±1.39
w/ cRT [11]		✓		49.5 ±1.05	65.8 ±0.47	30.1 ±0.50	48.0 ±0.43	40.8 ±1.95	61.6 ±1.83
w/ LA [16]		✓		53.3 ±0.44	70.6 ±0.21	30.2 ±0.44	48.7 ±0.89	42.8 ±1.78	63.1 ±1.13
PseudoLabel [14]	✓			47.8 ±1.06	63.4 ±0.81	30.7 ±0.18	47.8 ±0.40	42.3 ±0.83	60.4 ±1.11
USADTM [9]	✓			72.9 ±0.74	73.3 ±0.39	48.7 ±1.00	58.2 ±0.79	68.9 ±1.83	77.1 ±0.74
FixMatch [18]	✓			67.8 ±1.13	77.5 ±1.32	45.2 ±0.55	56.5 ±0.06	56.1 ±2.32	72.4 ±0.71
w/ CB re-weight [6]	✓	✓		72.2 ±1.28	80.9 ±1.52	46.0 ±0.27	58.3 ±0.46	58.9 ±2.79	74.7 ±0.55
w/ LA [16]	✓	✓		75.3 ±2.45	<u>82.0</u> ±0.36	47.3 ±0.42	58.6 ±0.36	63.4 ±2.99	75.9 ±1.25
w/ BOSS [17]	✓		✓	70.3 ±0.87	76.5 ±0.66	50.0 ±0.39	59.3 ±0.22	66.4 ±2.09	76.0 ±0.85
w/ DARP [12]	✓		✓	74.5 ±0.78	77.8 ±0.63	49.4 ±0.20	58.1 ±0.44	66.9 ±1.66	75.6 ±0.45
w/ CReST [24]	✓		✓	73.4 ±3.10	76.6 ±1.23	44.3 ±0.77	57.1 ±0.58	61.7 ±2.51	71.6 ±1.17
w/ CReST+ [24]	✓		✓	76.3 ±0.86	78.1 ±0.42	44.5 ±0.94	57.1 ±0.65	61.2 ±1.27	71.5 ±0.96
w/ DASO (Ours)	✓		✓	76.0 ±0.37	79.1 ±0.75	49.8 ±0.24	59.2 ±0.35	70.0 ±1.19	<u>78.4</u> ±0.80
w/ CB re-weight + DASO (Ours)	✓	✓	✓	<u>77.3</u> ±0.86	81.2 ±0.77	<u>50.3</u> ±0.18	<u>60.1</u> ±0.12	<u>70.2</u> ±1.05	77.8 ±0.58
w/ LA + DASO (Ours)	✓	✓	✓	**77.9** ±0.88	**82.5** ±0.08	**50.7** ±0.51	**60.6** ±0.71	**71.3** ±1.81	**79.0** ±0.58

Table 3. Comparison of accuracy (%) with different methods and their combinations on CIFAR10-LT, CIFAR100-LT, and STL10-LT under different label sizes with class imbalance. SSL denotes semi-supervised learning. LB and PB correspond to re-balancing for labels and pseudo-labels, respectively. Our DASO shows consistent performance gain over the baseline FixMatch [18], and adding label re-balancing to our method shows the best performance among the baselines. CIFAR10/100-LT benchmarks represent the $\gamma_l = \gamma_u$ setup, and STL10-LT corresponds to $\gamma_l \neq \gamma_u$ setup. We indicate the best results in bold and the second-best results with underlined.

CReST [24]. Self-training is adopted where a SSL algorithm is *iteratively re-trained* with adding some acceptable pseudo-labeled samples to the labeled data. The relative ratio of pseudo-labeled samples that will be added to the labeled set in next generation for each class k is defined as: $\mu_k = (N_{K+1-k}/N_1)^\alpha$, where N_k is the label size for class k, suggesting that minority-class pseudo-labels are more likely to be added. In CReST+, it adds the progressive distribution alignment (PDA) to the CReST method. To fairly compare with other baselines with $250k$ of the maximum iterations in total, we divide the whole iterations to 5 generations, where each generation trains $50k$ iterations for CIFAR10/100-LT and STL10-LT. For Semi-Aves, we divide the whole 90 epochs to 3 generations of 30 epochs. For CIFAR10/100-LT and STL10-LT, we set $\alpha = 1/3$ and $t_{\text{min}} = 0.5$, and $\alpha = 0.7$ and $t_{\text{min}} = 0.5$ for Semi-Aves respectively similar to [24].

ABC [15]. It trains an auxiliary balanced classifier (ABC) built upon a whole SSL learner (*e.g.*, FixMatch [18]). In particular, ABC shares the feature extractor with the existing pipeline, and learns the re-weighted versions of both cross-entropy with labels and *consistency regularization* from unlabeled data. The re-weight mechanism is performed by the balanced batch of labeled data and unlabeled data, where the batched images corresponding to each labels and predicted pseudo-labels are dropped with a probability sampled from Bernoulli distribution. Here, the parameter for Bernoulli is inversely proportional to the class frequency of the labels and pseudo-labels respectively. The ABC classifier is opted during inference.

D. Additional Experiments

D.1. Comprehensive Comparison with More Baselines

Experiments from the main paper evaluated DASO and other baseline methods specifically designed for *re-balancing the biased pseudo-labels* under class-imbalanced labels and distribution mismatch between $\mathcal{X}$ and $\mathcal{U}$. In Table 3, we introduce more diverse baseline methods for comparisons across different benchmarks including both $\gamma_l = \gamma_u$ and $\gamma_l \neq \gamma_u$ cases. As following, we term SSL methods as SSL, label re-balancing methods as LB, and the re-balancing methods for pseudo-labels as PB from Table 3. We consider *LDAM-DRW* [3], *classifier re-training (cRT)* [11], and *class re-weighting with effective number of samples (CB re-weight)* [6] for LB, respectively. For SSL methods, we additionally introduce *PseudoLabel* [14] and *USADTM* [9]. We further consider *BOSS* [17] as PB. The implementation details on those methods are explained in Appendix C.3. Note that we extensively compare PB methods based on other than FixMatch in Table 5.

We observe in Table 3 that applying LB improves the performance for *Supervised* and semi-supervised (SSL, PB) learning methods in general. This suggests that the bias of pseudo-label can be reduced by LB methods. In particular, the performance of DASO can be further pushed by additionally applying LB methods, as noted from *CB re-weight + DASO* and *LA + DASO*. This verifies that DASO is complementary to the existing LB methods, where the source for the performance improvement of DASO itself comes from the ability to *truly* alleviate the bias of pseudo-labels, not just re-balancing the labels.

D.2. DASO with Label Re-Balancing when $\gamma_l \neq \gamma_u$

We further evaluate DASO combined with other re-balancing techniques: LA [16] and ABC [15], when the class distribution of unlabeled data significantly differs from the labeled data (*e.g.*, $\gamma_l \neq \gamma_u$). In this setup, we conduct experiments with STL10-LT, as shown in Table 4.

	STL10-LT ($M = 100k$)			
	$\gamma_l = 10$		$\gamma_l = 20$	
Algorithm	$N_1 = 150$	$N_1 = 450$	$N_1 = 150$	$N_1 = 450$
FixMatch [18]	56.1 ±2.32	72.4 ±0.71	47.6 ±4.87	64.0 ±2.27
w/ DASO (Ours)	70.0 ±1.19	78.4 ±0.80	**65.7** ±1.78	75.3 ±0.44
FixMatch w/ LA [16]	64.4 ±1.35	75.9 ±1.25	51.5 ±3.23	67.4 ±1.04
w/ DASO (Ours)	**71.7** ±1.09	**79.0** ±0.58	65.6 ±1.43	**75.8** ±0.81
FixMatch + ABC [15]	66.3 ±1.00	77.1 ±0.56	59.3 ±2.66	73.0 ±0.91
w/ DASO (Ours)	69.6 ±0.94	77.9 ±0.89	64.5 ±2.81	74.7 ±0.16

Table 4. Comparison of accuracy (%) with the combination of various re-balancing methods on $\gamma_l \neq \gamma_u$ setup. DASO somewhat obtains performance gain when even combined with either LA [16] or ABC [15] on FixMatch. We indicate the best results as bold.

We observe that both LA [16] and ABC [15], are beneficial upon baseline FixMatch. Moreover, the performance can be further pushed when DASO is applied on top of those methods. However, the performances show marginal improvements compared to the FixMatch w/ DASO. This opens a new challenge that calls for the design of an unified re-balancing approach of labels and unlabeled data, which can also well address the potentially unknown unlabeled data.

D.3. Comparison based on ReMixMatch

To verify the efficacy of DASO as a *generic framework*, we further compare the pseudo-label re-balancing (PB) methods based on ReMixMatch [1]. In particular, we provide the results as the same way when DASO is integrated with FixMatch [18] from the main paper. Table 5 shows the results. We compare each method on CIFAR10/100-LT and STL10-LT, varying the imbalance ratio while the amount of labels used are fixed by N_1. Note that for CIFAR benchmarks, $\gamma = \gamma_l = \gamma_u$.

	CIFAR10-LT $N_1 = 500, M_1 = 4000.$		CIFAR100-LT $N_1 = 50, M_1 = 400.$		STL10-LT $N_1 = 150.$	
Algorithm	$\gamma = 100$	$\gamma = 150$	$\gamma = 10$	$\gamma = 20$	$\gamma_l = 10$	$\gamma_l = 20$
ReMixMatch [1]	70.9 ±2.37	64.7 ±0.95	52.3 ±0.91	46.5 ±0.30	54.4 ±2.15	46.5 ±1.93
w/ DARP [12]	72.2 ±2.72	65.7 ±1.20	52.8 ±0.65	47.0 ±0.17	61.2 ±2.62	59.5 ±2.56
w/ CReST+ [24]	75.6 ±1.60	65.9 ±2.20	49.9 ±0.80	44.5 ±1.04	64.1 ±1.68	49.2 ±0.90
w/ DASO (Ours)	**76.8** ±0.81	**68.5** ±0.98	**53.6** ±0.81	**47.8** ±0.69	**75.0** ±0.95	**68.5** ±5.14

Table 5. Comparison of accuracy (%) with various pseudo-label re-balancing (PB) methods upon different baseline SSL learner, ReMixMatch [1]. DASO outperforms all the other methods by a significant margin, which is consistent with the results when the baseline SSL learner was FixMatch from the main paper. We indicate the best results as bold.

As can be seen, DASO achieves the best results among the baselines for comparison. From CIFAR benchmarks (*e.g.*, $\gamma_l = \gamma_u$), DASO outperforms both DARP [12] and CReST+ [24] that leverages the assumption of $\gamma_l = \gamma_u$ explicitly; for example, they utilize the actual class distribution of unlabeled data. As note, while CReST+ is beneficial for ReMixMatch when trained on CIFAR10-LT, but it performs worse in CIFAR100-LT results. This might come from the limited amount of labels and the repeated training with re-initializing models via self-training. For STL10-LT cases, the improvements from both DARP and CReST+ can be limited due to the mismatch of class distributions between the labeled data and unlabeled

data. In contrary, DASO significantly surpasses the other methods without the access to the class distribution of either labels or unlabeled data. To summarize, DASO can improve typical baseline SSL methods under imbalanced data *in general*.

D.4. Results on Test-Time Logit Adjustment

In the main paper, we have considered Logit Adjustment (LA) [16] as applying logit-adjusted cross-entropy loss during training. This point is also explained in Appendix C.3. On the other hand, we also consider adjusting the logits during inference also present in [16]; we denote this type of LA as *LA (inf)*. In Table 6, we report the results obtained from LA [16] by this strategy when the class distribution of labeled data and unlabeled data are identical ($\gamma = \gamma_l = \gamma_u$).

	CIFAR10-LT $N_1 = 1500, M_1 = 3000$		CIFAR100-LT $N_1 = 50, M_1 = 300$	
Algorithm	$\gamma = 100$	$\gamma = 150$	$\gamma = 10$	$\gamma = 20$
FixMatch [18]	77.5 ±1.32	72.4 ±1.03	56.5 ±0.06	50.7 ±0.25
FixMatch w/ LA [16]	82.0 ±0.36	78.0 ±0.91	58.6 ±0.36	53.4 ±0.32
FixMatch w/ LA + CReST+ [24]	81.1 ±0.57	77.9 ±0.71	57.1 ±0.55	52.3 ±0.20
FixMatch w/ LA + DASO (Ours)	82.5 ±0.08	79.0 ±2.23	**60.6** ±0.71	55.1 ±0.72
FixMatch w/ LA (inf) [16]	82.8 ±1.43	79.2 ±1.15	58.7 ±0.63	53.3 ±0.43
FixMatch w/ LA (inf) + CReST+ [24]	82.9 ±0.24	80.3 ±0.56	57.8 ±0.47	53.3 ±0.83
FixMatch w/ LA (inf) + DASO (Ours)	**84.5** ±0.55	**81.8** ±0.83	60.5 ±0.49	**55.2** ±0.47

Table 6. Comparison of accuracy (%) with different strategies of applying Logit Adjustment (LA) [16]: either train-time (noted as LA) or during inference (noted as LA (inf)). We observe large gains compared to baseline FixMatch when LA is applied during inference.

D.5. More Ablation Study

We conduct several ablation studies on the hyper-parameters in DASO framework. As the same with the ablation study conducted from the main paper, we consider FixMatch [18] with DASO on CIFAR10-LT with $N_1 = 500$, $\gamma = 100$ (denoted as C10) and STL10-LT with $N_1 = 150$, $\gamma_l = 10$ (denoted as STL10) respectively. Table 7 compares different values of the queue size L for constructing the *balanced* prototypes. Table 8 tests different temperature factor T_{proto} for the similarity-based classifier. Finally, Table 9 shows the effect of different loss weights λ_{align} for the semantic alignment loss. We shaded rows that correspond to the hyper-parameter of the complete DASO framework. We also indicate the best results in bold.

	C10	STL10
FixMatch	68.25	55.53
$L = 128$	73.77	69.17
$L = 256$	**75.97**	**70.21**
$L = 512$	75.03	69.96
$L = 1024$	74.36	69.64
$L = 2048$	73.50	69.99

Table 7. Ablation study on L, the *balanced* queue size.

	C10	STL10
FixMatch	68.25	55.53
$T_{\text{proto}} = 0.02$	73.84	68.19
$T_{\text{proto}} = 0.05$	**75.97**	**70.21**
$T_{\text{proto}} = 0.2$	70.53	66.62
$T_{\text{proto}} = 0.5$	52.36	60.92
$T_{\text{proto}} = 1.0$	46.47	57.40

Table 8. Ablation study on T_{proto} for semantic pseudo-label.

	C10	STL10
FixMatch	68.25	55.53
$\lambda_{\text{align}} = 0$	70.98	61.64
$\lambda_{\text{align}} = 0.5$	73.78	69.01
$\lambda_{\text{align}} = 1$	**75.97**	70.21
$\lambda_{\text{align}} = 1.5$	74.59	**71.51**
$\lambda_{\text{align}} = 2$	74.57	71.12

Table 9. Ablation study on λ_{align}, which is a weight for $\mathcal{L}_{\text{align}}$.

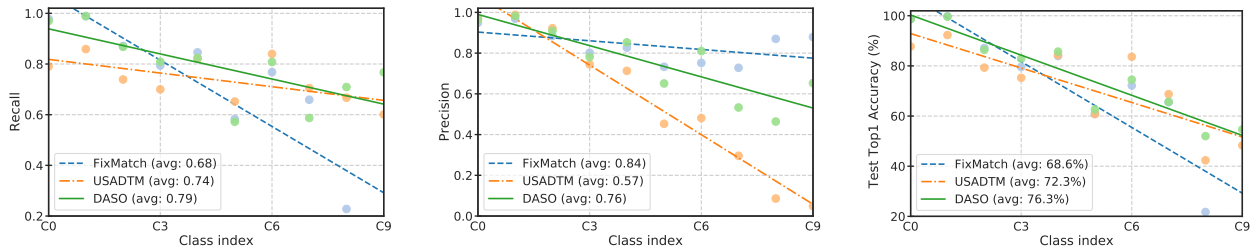
As note, we do not tune the hyper-parameters above (L, T_{proto}, λ_{align}) depending on different benchmarks across different imbalance ratio. For example, in STL10-LT case, using λ_{align} value higher than 1 seems effective, but the result of 70.21% obtained from $\lambda_{\text{align}} = 1$ already performs well.

E. Detailed Analysis

E.1. Recall and Precision Analysis

E.1.1 Detailed comparison for linear pseudo-label and semantic pseudo-label methods

We first take a closer look at the bias of pseudo-labels of each method by analyzing per-class recall and precision. We then compare the class-wise test accuracy of each model to evaluate the capability for each class, as done in the main paper. Fig. 1 provides the comparison of FixMatch w/ DASO (ours) and USADTM [9] over FixMatch [18] trained on CIFAR10-LT.



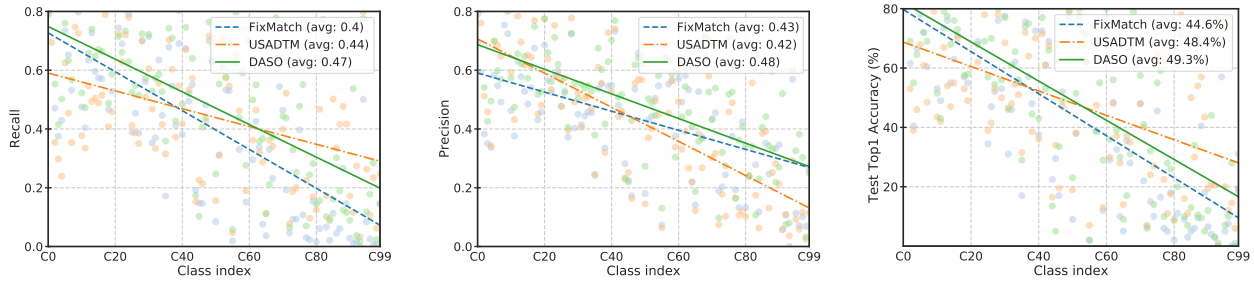
(a) Recall of pseudo-labels (b) Precision of pseudo-labels (c) Class-wise test accuracy

Figure 1. Analysis of bias in pseudo-labels and test accuracy. We consider FixMatch [18] for linear pseudo-labels, USADTM [9] for semantic pseudo-labels, and the proposed FixMatch w/ DASO trained on CIFAR10-LT with $N_1 = 500$ with $\gamma_l = \gamma_u = 100$.

Compared to the linear pseudo-labels, the recall of semantic pseudo-labels on minority classes significantly increased in Fig. 1a. However, their precision values are degraded on the minorities, which means that the semantic pseudo-labels have the bias towards the minorities, leading to performance drop on the majority classes.

In contrary, the pseudo-labels generated from our DASO maintain high precision while the recall on the minority classes increased, encouraging high performance on both of majority and minority classes. From the analyses, pseudo-labels from DASO find the trade-off between linear and semantic pseudo-labels with respect to the bias that performs well on test data. Since DASO also aims to keep the prediction of majority classes, the test accuracy drop on the head classes is well addressed.

Note that Fig. 2 shows the same analysis on the models trained on CIFAR100-LT.



(a) Recall of pseudo-labels (b) Precision of pseudo-labels (c) Class-wise test accuracy

Figure 2. Analysis of bias in pseudo-labels. We consider FixMatch [18] for linear pseudo-labels, USADTM [9] for semantic pseudo-labels, and the proposed FixMatch w/ DASO trained on CIFAR100-LT with $N_1 = 50$ with $\gamma_l = \gamma_u = 10$.

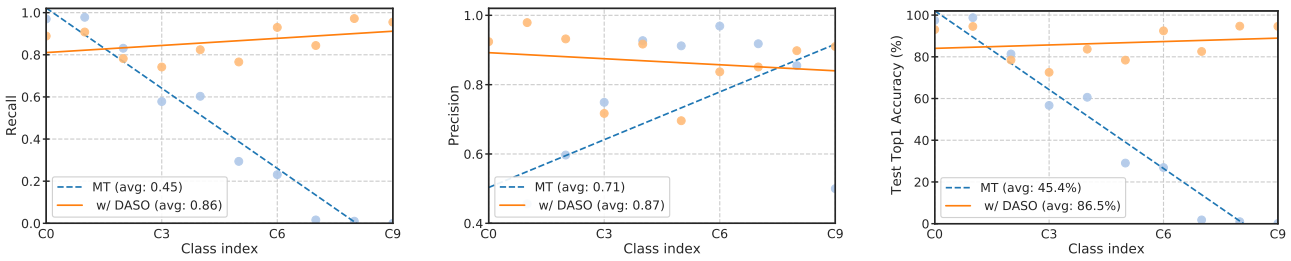
E.1.2 DASO with class distribution mismatch on traditional SSL learner

We present the analyses of bias in pseudo-labels for the other *classic* SSL algorithms: MeanTeacher [21] and MixMatch [2] in Figs. 3 and 4, respectively, in case of uniform distribution of unlabeled data; *i.e.*, $\gamma_u = 1$. In such a case, class distribution mismatch (*i.e.*, $\gamma_l \neq \gamma_u$) can damage the accuracy of the model.

From the recall curves in Figs. 3a and 4a and the precision curves in Figs. 3b and 4b, the pseudo-labels of the baseline SSL learners are severely biased towards the head classes, since most of the minority class examples are collapsed to the majority class ones. The unlabeled data with $\gamma_u = 1$ rather significantly accelerated the bias, to the point where the precision curve is completely reversed; precision values in the majority classes significantly degraded, compared to the recall curve. Thereby, the model rarely predicts some of the minority class examples for the test dataset in Figs. 3c and 4c.

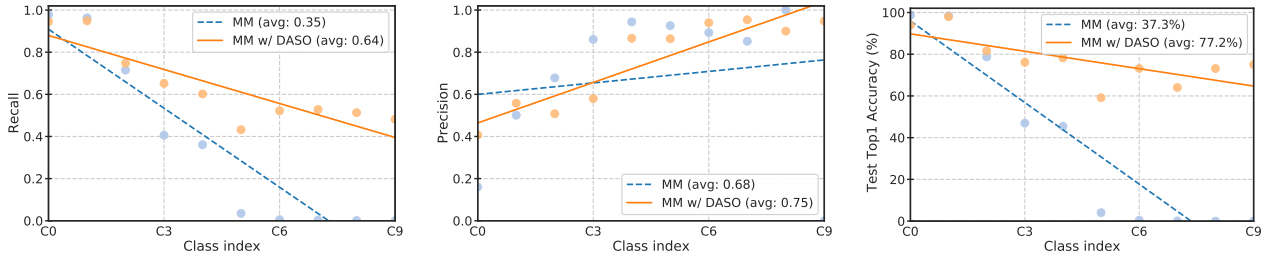
In contrast, we demonstrate that DASO can even *completely* mitigate such a devastating bias, by just coupling the linear pseudo-labels with the semantic pseudo-labels obtained from the similarity-based classifier. In this case, the semantic alignment loss $\mathcal{L}_{\text{align}}$ is not applied, due to the absence of advanced augmentation $\mathcal{A}_s$ for MeanTeacher and MixMatch. Surprisingly, in MeanTeacher (MT) with DASO, the recall and precision values become uniform, resulting in a *uniform* per-class test accuracy in Fig. 3c. When combined with MixMatch [2], DASO also recovers the minority-class pseudo-labels significantly. In final, the averaged test accuracy can be more than doubled (*i.e.*, $37.3\% \rightarrow 77.2\%$), as shown in Fig. 4c.

As such, DASO helps alleviate the bias in pseudo-labels, even when the class distributions between labeled and unlabeled data substantially differ, without accessing the knowledge about the underlying distribution of unlabeled data.



(a) Recall of pseudo-labels (b) Precision of pseudo-labels (c) Class-wise test accuracy

Figure 3. Analysis of bias in pseudo-labels and test accuracy. We consider MeanTeacher (MT) [21], and the proposed DASO applied to MT (MT w/ DASO) trained on CIFAR10-LT with $N_1 = 1500$ with $\gamma_l = 100$ and $\gamma_u = 1$.



(a) Recall of pseudo-labels (b) Precision of pseudo-labels (c) Class-wise test accuracy

Figure 4. Analysis of bias in pseudo-labels and test accuracy. We consider MixMatch (MM) [21], and the proposed DASO applied to MM (MM w/ DASO) trained on CIFAR10-LT with $N_1 = 1500$ with $\gamma_l = 100$ and $\gamma_u = 1$.

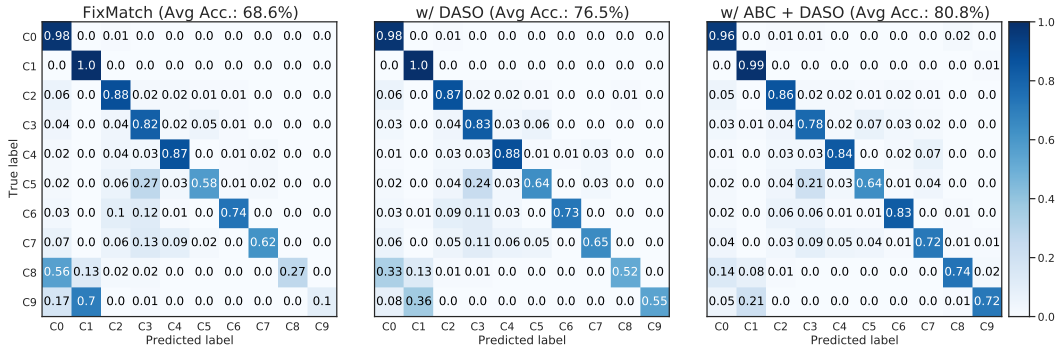


Figure 5. Analysis of predictions from test data via confusion matrix. All the methods are trained on CIFAR10-LT with $\gamma = 100$ and $N_1 = 500$ upon the same fixed random seed. DASO greatly recovers the predictions on the actual minority class examples in test data.

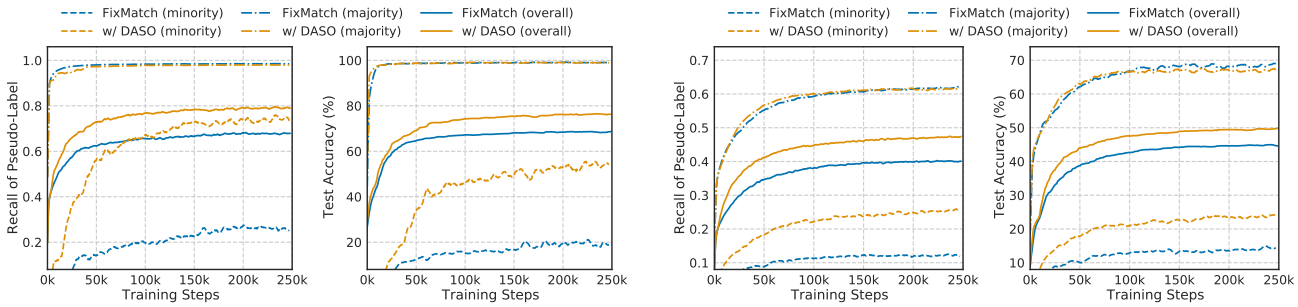
E.2. Confusion Matrix on Test Data

We compare the confusion matrices of the predictions from the test data. From the baseline FixMatch [18], we further apply our DASO on both FixMatch and FixMatch w/ ABC [15]. As shown in Fig. 5, the predictions on the tail classes (*e.g.*, C8 and C9) in FixMatch are severely biased towards the majority classes (*e.g.*, C1). This limits the overall performance, which is carried by the non-minority classes (68.6%). On the other hand, from the center of Fig. 5, DASO significantly alleviates the bias towards the head classes observing C8 and C9 classes, while the performances on the other classes are well maintained. When DASO is integrated with ABC [15] in the right figure, the accuracy values are further improved.

E.3. Train Curves for Recall and Accuracy

We compare the train curves of recall and test accuracy values from FixMatch [18] and FixMatch w/ DASO (Ours) trained on CIFAR10/100-LT respectively in Figs. 6a and 6b. Here, we plot those from majority classes (*e.g.*, first 20% classes) and minority classes (*e.g.*, last 20% classes), in addition to the overall values. From both CIFAR10/100-LT benchmarks, DASO significantly improves the recall and test accuracy values on the minority classes, while relatively maintaining those from the

majority classes. This verifies the efficacy of DASO that specifically handles the biased minority classes in unlabeled data.



(a) Train curves on CIFAR10-LT (b) Train curves on CIFAR100-LT

Figure 6. Train curves for the recall and test accuracy values obtained from FixMatch and FixMatch w/ DASO (Ours). The training details are consistent from the main paper. DASO well reduces the biases on the tail classes, while preserving those from the head classes.

E.4. Further Comparison of Feature Representations

To verify the efficacy of the proposed semantic alignment loss ($\mathcal{L}_{\text{align}}$), we further visualize the t-SNE [22] of the feature encoder outputs from FixMatch w/ $\mathcal{L}_{\text{align}}$ in the center of Fig. 7. Compared to FixMatch, applying $\mathcal{L}_{\text{align}}$ without the class-adaptive pseudo-label blending can already cluster the minority classes (*e.g.*, C6, C8, and C9) in the center of the figure. However, those indicated clusters lie nearby the head-class clusters (*e.g.*, C0 and C1), where the classifier can still be confused. In that sense, the complete DASO from the right figure further improves the separability of the tail classes from the head classes. This demonstrates that while applying the semantic alignment loss $\mathcal{L}_{\text{align}}$ could be helpful for the minority classes, both class-adaptive pseudo-label blending and $\mathcal{L}_{\text{align}}$ are the essential components for our DASO framework.

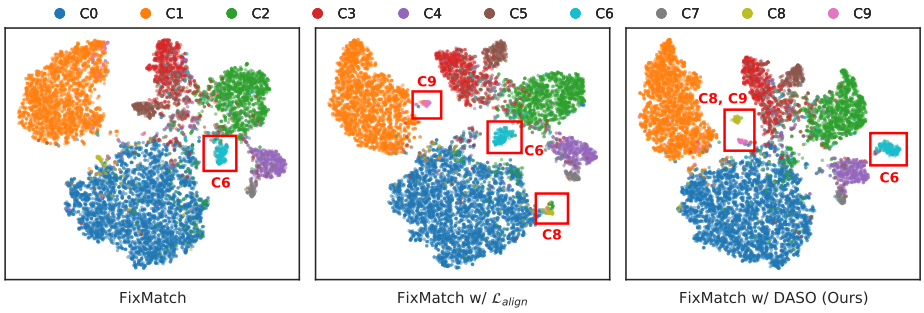
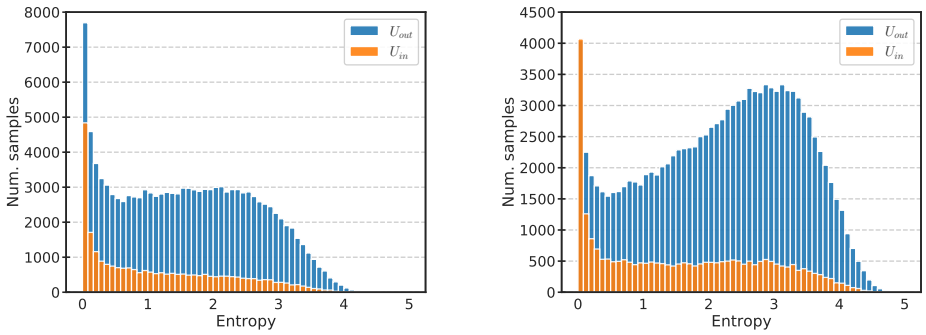


Figure 7. Comparison of t-SNE [22] visualizations of feature representations. We additionally compare the model trained with FixMatch w/ $\mathcal{L}_{\text{align}}$ between the original FixMatch [18] and FixMatch w/ DASO (Ours). Note that both of the semantic alignment loss $\mathcal{L}_{\text{align}}$ and our class-adaptive pseudo-label blending contribute to alleviating the bias in pseudo-labels in perspective of feature representation.

E.5. Confidence Analysis from Out-of-class Examples

To investigate the efficacy of DASO pseudo-label, we analyze the confidence of predictions of unlabeled data after training model with $\mathcal{U} = \mathcal{U}_{\text{in}} + \mathcal{U}_{\text{out}}$ under Semi-Aves benchmark [20]. Fig. 8 visualizes the histograms of entropy values obtained from either FixMatch [18] or FixMatch w/ DASO, respectively. Note that since both models do not explicitly learn how to distinguish *in-class* and *out-of-class* categories at all, those samples cannot be completely separated in confidence plot.

FixMatch w/ DASO, which learned the blending of linear and semantic pseudo-labels can be effective in that the *out-of-class* examples in $\mathcal{U}_{\text{out}}$ are further pushed towards the low-confidence region (*i.e.*, higher entropy) compared to the *in-class* unlabeled examples in $\mathcal{U}_{\text{in}}$. For example, about $8k$ out-of-class examples correspond to the most confident samples in Fig. 8a, while they reduced to $4k$ with DASO in Fig. 8b. We suppose DASO has the *implicit* ability to push more examples corresponding to *out-of-class* that can cause degradation, towards the low-confident area. This point implies the potential application of DASO towards an open-set SSL scenario, where SSL algorithms also observe unlabeled data in a broader class distribution compared to the labels, and learning without *harmful* out-of-class examples would be important.



(a) FixMatch [18] (b) FixMatch w/ DASO (Ours)

Figure 8. Comparisons of DASO and FixMatch [18] on the distribution of entropy values from the predictions of samples in $\mathcal{U}_{\text{in}}$ and $\mathcal{U}_{\text{out}}$ of *Semi-Aves* benchmark [20], respectively. We observe that examples $\mathcal{U}_{\text{in}}$ relatively remain in low-entropy (*e.g.*, high-confidence) area, while those in $\mathcal{U}_{\text{out}}$ are well pushed towards the high-entropy (*e.g.*, low-confidence) area from DASO (ours).

F. Overall Framework

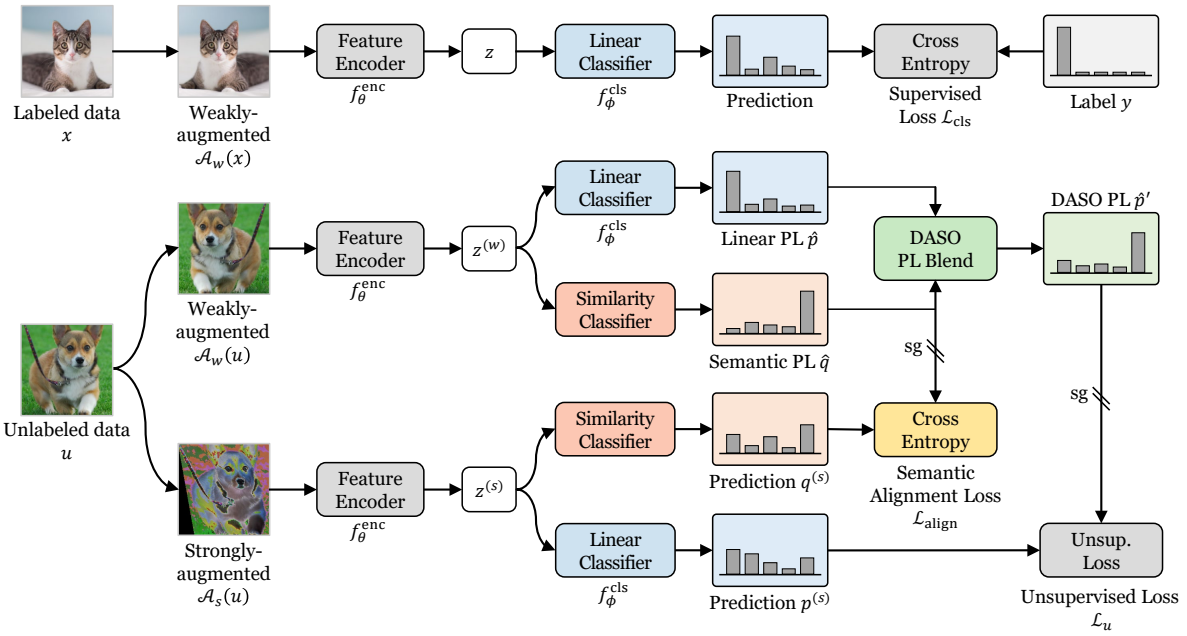


Figure 9. Overall framework of DASO including the blending of pseudo-labels (DASO PL Blend) and the semantic alignment loss ($\mathcal{L}_{\text{align}}$). As explained in Sec. 3.3 of the main paper, 'balanced prototypes' for executing the similarity-based classifier are generated from EMA features of labeled data, which is omitted in this figure. Two main components of DASO framework (blending of pseudo-labels and semantic alignment loss) can easily integrate with typical semi-supervised learning algorithms such as FixMatch [18] and ReMixMatch [1] for debiasing pseudo-labels. Note that 'sg' means stop-gradient operation.

References

[1] David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *International Conference on Learning Representations (ICLR)*, 2020. 6, 8, 13

[2] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems (NIPS)*, volume 32, pages 5049–5059, 2019. 6, 10

[3] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems (NIPS)*, volume 32, pages 1567–1578, 2019. 4, 5, 7

[4] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 15, pages 215–223, 2011. 4

[5] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Advances in Neural Information Processing Systems (NIPS)*, 2020. 2, 6

[6] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9268–9277, 2019. 4, 5, 7

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 4

[8] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 2, 6

[9] Tao Han, Junyu Gao, Yuan Yuan, and Qi Wang. Unsupervised semantic aggregation and deformable template matching for semi-supervised learning. In *Advances in Neural Information Processing Systems (NIPS)*, volume 33, pages 9972–9982, 2020. 6, 7, 9, 10

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 4

[11] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations (ICLR)*, 2020. 5, 7

[12] 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. In *Advances in Neural Information Processing Systems (NIPS)*, 2020. 4, 6, 7, 8

[13] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical report*, 2009. 4

[14] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, 2013. 5, 6, 7

[15] Hyuck Lee, Seungjae Shin, and Heeyoung Kim. Abc: Auxiliary balanced classifier for class-imbalanced semi-supervised learning. *arXiv preprint arXiv:2110.10368*, 2021. 7, 8, 11

[16] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations (ICLR)*, 2021. 5, 7, 8, 9

[17] Leslie N Smith and Adam Conovaloff. Building one-shot semi-supervised (boss) learning up to fully supervised performance. *arXiv preprint arXiv:2006.09363*, 2020. 6, 7

[18] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems (NIPS)*, 2020. 6, 7, 8, 9, 10, 11, 12, 13

[19] Jong-Chyi Su, Zezhou Cheng, and Subhransu Maji. A realistic evaluation of semi-supervised learning for fine-grained classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 4

[20] Jong-Chyi Su and Subhransu Maji. The semi-supervised inaturalist-aves challenge at fgvc7 workshop, 2021. 4, 12, 13

[21] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems (NIPS)*, volume 30, pages 1195–1204, 2017. 6, 10, 11

[22] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 12

[23] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8769–8778, 2018. 4

[24] Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Yuille, and Fan Yang. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 7, 8, 9

[25] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference (BMVC)*, pages 87.1–87.12, 2016. 4

[26] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*, 2018. 6