A. Algorithms

We present the DC-SSL algorithms with training-free and training-based strategies in Algorithms 1 and 2. Due to their simplicity, both strategies are easy to be implemented with only minor changes to FixMatch.

**Algorithm 1 DC-SSL with Training-free strategy**

**Input:** labeled batch \( \mathcal{X} = \{(x_b, y_b) \mid (x_b, y_b) \in D_x\}_{b=1}^B \), unlabeled batch \( \mathcal{U} = \{u_b \mid u_b \in D_u\}_{b=1}^B \).

**Parameter:** unlabeled ratio \( \mu \), momentum coefficient \( \alpha \), threshold \( \tau \), loss weight \( \lambda_u \), and \( \lambda_d \).

1. Calculate supervised loss \( L_x \) on \( \mathcal{X} \) using Eq. (2).
2. for \( b = 1 \) to \( \mu B \) do
3. Obtain \( f \)'s predictions \( p_b^{w,f} \) and \( p_b^{r,f} \) for \( u_b^w \) and \( u_b^r \).
4. Obtain \( g \)'s prediction \( p_b^{w,g} \) for \( u_b^w \).
5. end for
6. Estimate the PCD \( q^f \) using Eq. (4).
7. Estimate the RCD \( q^g \) using Eq. (5).
8. Obtain revised pseudo-labels \( p_b^{w,f} \) using Eq. (6).
9. Calculate consistency loss \( L_u \) using Eq. (7).
10. return \( L_x + \lambda_u L_u + \lambda_d L_d \)

**Algorithm 2 DC-SSL with Training-based strategy**

**Input:** labeled batch \( \mathcal{X} = \{(x_b, y_b) \mid (x_b, y_b) \in D_x\}_{b=1}^B \), unlabeled batch \( \mathcal{U} = \{u_b \mid u_b \in D_u\}_{b=1}^B \).

**Parameter:** unlabeled ratio \( \mu \), momentum coefficient \( \alpha \), threshold \( \tau \), loss weight \( \lambda_u \), and \( \lambda_d \).

1. Calculate supervised loss \( L_x \) on \( \mathcal{X} \) using Eq. (2).
2. for \( b = 1 \) to \( \mu B \) do
3. Obtain \( f \)'s predictions \( p_b^{w,f} \) and \( p_b^{r,f} \) for \( u_b^w \) and \( u_b^r \).
4. Obtain \( g \)'s prediction \( p_b^{w,g} \) for \( u_b^w \).
5. end for
6. Estimate the PCD \( q^f \) using Eq. (4).
7. Estimate the RCD \( q^g \) using Eq. (5).
8. Calculate instance-consistency loss \( L_u \) using Eq. (9).
9. Calculate distribution-consistency loss \( L_d \) using Eq. (8).
10. return \( L_x + \lambda_u L_u + \lambda_d L_d \)

B. EMA model for pseudo-label generations

In this section, we investigate two different ways to use the EMA model’s predictions as pseudo-labels, as shown in Figs. 5b and 5c. The revised-V1 directly replaces the \( f \) with \( g \) to produce pseudo-labels, while the revised-V2 retains the FixMatch structure and generate pseudo-labels on \( g \). As shown in Tab. 7, the revised-V2 can achieve relatively higher test accuracy, simply because it ensures \( f \) can also view the weakly-augmented images, which can slightly benefit the Batch-normalization layers of \( f \). However, as we have discussed in Sec. 3.2.1, the accuracy in both ways decreases sharply compared to the original FixMatch.

C. Benefits of the EMA model

In our paper, based on the analysis on CIFAR10 with 40 labels, we observe that the EMA model can achieve a higher accuracy of pseudo-labels on all unlabeled data but a lower accuracy on high-confidence ones. As shown in Fig. 6, we find the same observations on MiniImageNet with 1000 labels. In addition, we investigate the performance of the EMA model in a mismatched distribution setting on CIFAR-10 with \( |D_x| = 40 \) and \( \gamma_u = 50 \). It can be seen from Fig. 7 that \( g \) can outperform \( f \) throughout the training process in terms of the accuracy on all pseudo-labels, yet with lower accuracy on high-confidence ones.
Figure 6. Following the explorations in Fig. 4, we observe similar findings on MiniImageNet with 1000 labels. (b) In terms of the accuracy difference \( Q_f - Q_g \) of the high-confidence pseudo-labels in a mini-batch, \( g \) obtains a lower accuracy than \( f \) at about 61% iterations. (c) However, the accurate difference \( A_f - A_g \) of all pseudo-labels between \( f \) and \( g \) shows that the model \( g \) can generate more accurate pseudo-labels in 88% iterations.

Figure 7. Following the explorations in Fig. 4, we investigate the EMA model’s performance on CIFAR10 in a mismatched distribution setting as in Fig. 1b. (b) In terms of the accuracy difference \( Q_f - Q_g \) of the high-confidence pseudo-labels in a mini-batch, \( g \) obtains a lower accuracy than \( f \) at about 67% iterations. (c) However, the accurate difference \( A_f - A_g \) of all pseudo-labels between \( f \) and \( g \) shows that the model \( g \) can generate more accurate pseudo-labels in 97% iterations.