## Appendix A. Experiment Details

In Table 5, we show the complete set of hyperparameters in our semi-supervised learning experiments.

| Dataset | $B$ | $\mu$ | $\lambda_{cls}$ | $\alpha$ | $K$ | $t$ | $\tau$ | $T$ | $\lambda_{ctr}$ |
|---|---|---|---|---|---|---|---|---|---|
| CIFAR-10 | 64 | 7 | 1 | 0.9 | 2560 | 0.2 | 0.95 | 0.8 | 1 |
| STL-10 | | | | | | | | | 5 |
| ImageNet 1% labels | 160 | 4 | 10 | 0.9 | 30000 | 0.1 | 0.6 | 0.3 | 10 |
| ImageNet 10% labels | | | | | | | 0.5 | 0.2 | 2 |

Table 5: Hyperparameters for CoMatch in the semi-supervised learning experiments.

The strong augmentation $\text{Aug}_\text{s}$ on ImageNet unlabeled data uses color distortion in addition to the standard crop-and-flip. A pseudo-code for the color distortion in PyTorch is as follows:

```
from torchvision import transforms as T
color_jitter = T.ColorJitter(0.4,0.4,0.4,0.1)
transforms.Compose([
  T.RandomApply([color_jitter], p=0.8)
  T.RandomGrayscale(p=0.2)])
```

## Appendix B. MB and MQ in CoMatch

Figure 5 illustrates how the EMA model is utilized in CoMatch to construct the memory bank (MB) and the momentum queue (MQ). The memory bank contains the class probability and the low-dimensional embeddings for both weakly-augmented labeled samples and weakly-augmented unlabeled samples. The momentum queue contains the pseudo-labels for the unlabeled samples and their strongly-augmented embeddings.

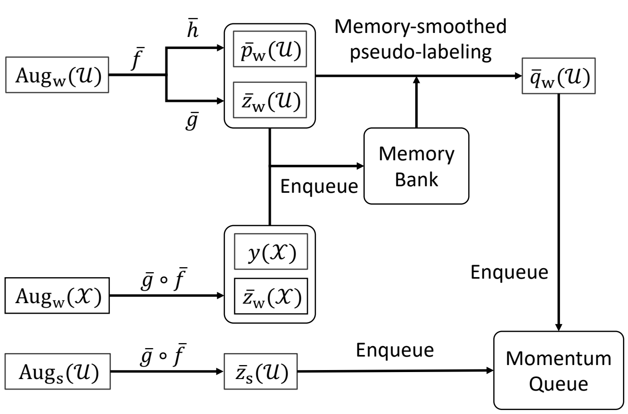


Figure 5: Illustration of the memory bank and the momentum queue. $\mathcal{U}$ is the batch of unlabeled data, $\mathcal{X}$ is the batch of labeled data. $\bar{f}$, $\bar{h}$, and $\bar{g}$ refer to the EMA version of the encoder, the classification head, and the projection head, respectively.

## Appendix C. Pseudo-code of CoMatch

Algorithm 1 presents the pseudo-code of CoMatch.

**Algorithm 1:** Pseudo-code of CoMatch (one iteration).

**Input:** labeled batch $\mathcal{X} = \{(x_b, y_b)\}_{b=1}^{B}$, unlabeled batch $\mathcal{U} = \{u_b\}_{b=1}^{\mu B}$, encoder $f$, classifier $h$, projection head $g$, memory bank $\text{MB} = \{(p_k^w, z_k^w)\}_{k=1}^{K}$.

**for** $b \in \{1, ..., \mu B\}$ **do**
  `// class probability prediction`
  $p_b^w = h \circ f(\text{Aug}_\text{w}(u_b))$
  `// distribution alignment`
  $p_b^w = \text{DA}(p_b^w)$
  `// weakly-augmented embedding`
  $z_b^w = g \circ f(\text{Aug}_\text{w}(u_b))$
  `// memory-smoothed pseudo-labeling`
  **for** $k \in \{1, ..., K\}$ **do**
    $a_k = \frac{\exp(z_b^w \cdot z_k^w / t)}{\sum_{k=1}^{K} \exp(z_b^w \cdot z_k^w / t)}$ `// affinity`
  **end**
  $q_b = \alpha p_b^w + (1-\alpha) \sum_{k=1}^{K} a_k p_k^w$
  `// strongly-augmented embeddings`
  $z_b = g \circ f(\text{Aug}_\text{s}(u_b))$
  $z_b' = g \circ f(\text{Aug}_\text{s}'(u_b))$
**end**
**for** $b \in \{1, ..., \mu B\}$ **do**
  **for** $j \in \{1, ..., \mu B\}$ **do**
    `// pseudo-label graph`
    $$W_{bj}^q = \begin{cases} 1 & \text{if } b = j \\ q_b \cdot q_j & \text{if } b \neq j \text{ and } q_b \cdot q_j \geq T \\ 0 & \text{otherwise} \end{cases}$$
    `// embedding graph`
    $$W_{bj}^z = \begin{cases} \exp(z_b \cdot z_b' / t) & \text{if } b = j \\ \exp(z_b \cdot z_j / t) & \text{if } b \neq j \end{cases}$$
  **end**
  $\hat{W}^q = \text{Normalize}(W^q)$
  $\hat{W}^z = \text{Normalize}(W^z)$
**end**
`// losses`
$\mathcal{L}_x = \frac{1}{B} \sum_{b=1}^{B} \text{H}(y_b, p(y|\text{Aug}_\text{w}(x_b)))$
$\mathcal{L}_u^{cls} = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}(\max q_b \geq \tau) \text{H}(q_b, p(y|\text{Aug}_\text{s}(u_b)))$
$\mathcal{L}_u^{ctr} = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \text{H}(\hat{W}_b^q, \hat{W}_b^z)$
$\mathcal{L} = \mathcal{L}_x + \lambda_{cls} \mathcal{L}_u^{cls} + \lambda_{ctr} \mathcal{L}_u^{ctr}$
update $f$, $h$, $g$ with SGD to minimize $\mathcal{L}$.