

# CGMatch: A Different Perspective of Semi-supervised Learning

Bo Cheng<sup>1</sup>, Jueqing Lu<sup>2</sup>, Yuan Tian<sup>1\*</sup>, Haifeng Zhao<sup>3</sup>, Yi Chang<sup>1\*</sup>, Lan Du<sup>2\*</sup>

<sup>1</sup>Jilin University, China

<sup>2</sup>Monash University, Australia

<sup>3</sup>Jinling Institute of Technology, China

chengbo21@mails.jlu.edu.cn, jueqing.lu@monash.edu, yuantian@jlu.edu.cn

zhf@jit.edu.cn, yichang@jlu.edu.cn, lan.du@monash.edu

## Abstract

*Semi-supervised learning (SSL) has garnered significant attention due to its ability to leverage limited labeled data and a large amount of unlabeled data to improve model generalization performance. Recent approaches achieve impressive successes by combining ideas from both consistency regularization and pseudo-labeling. However, these methods tend to underperform in the more realistic situations with relatively scarce labeled data. We argue that this issue arises because existing methods rely solely on the model’s confidence, making them challenging to accurately assess the model’s state and identify unlabeled examples contributing to the training phase when supervision information is limited, especially during the early stages of model training. In this paper, we propose a novel SSL model called CGMatch, which, for the first time, incorporates a new metric known as Count-Gap (CG). We demonstrate that CG is effective in discovering unlabeled examples beneficial for model training. Along with confidence, a commonly used metric in SSL, we propose a fine-grained dynamic selection (FDS) strategy. This strategy dynamically divides the unlabeled dataset into three subsets with different characteristics: easy-to-learn set, ambiguous set, and hard-to-learn set. By selective filtering subsets, and applying corresponding regularization with selected subsets, we mitigate the negative impact of incorrect pseudo-labels on model optimization and generalization. Extensive experimental results on several common SSL benchmarks indicate the effectiveness of CGMatch especially when the labeled data are particularly limited. Source code is available at <https://github.com/BoCheng-96/CGMatch>.*

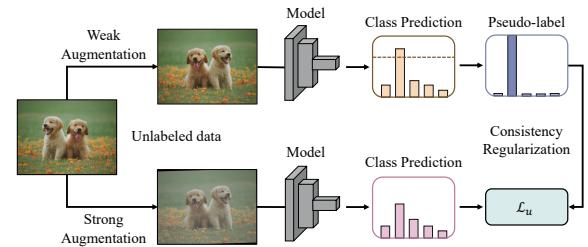


Figure 1. Paradigm of X-Match. Given an unlabeled sample, the weakly-augmented image is fed into the model, and the sample is assigned to a corresponding class (i.e., **pseudo-label**) when its highest prediction probability exceeds the confidence threshold. This threshold is determined by various strategies proposed in X-Match. **Consistency regularization** is then applied to ensure that the prediction from the strongly-augmented version remains consistent with the pseudo-label.

## 1. Introduction

Semi-supervised learning (SSL) has gained increasing attention across various fields (e.g., computer vision [17, 26, 42, 50], natural language processing [6, 13, 21], and bioinformatics [27, 30, 45]) due to its ability to utilise both labeled and unlabeled data for model training. This is beneficial in many real-world situations where large-scale labeled datasets are costly or labour-intensive to acquire.

The key of SSL is not only leveraging the informative semantic view provided by labeled data, but also enabling the model to capture the underlying structure in unlabeled data, thereby improving generalization and performance in downstream tasks, such as visual recognition [24, 41]. Numerous methods have been explored [8, 38, 43, 52], including pseudo-labeling (PL) methods [19], consistency regularization methods [1, 18, 34], deep generative methods [14, 20, 22, 35], and graph-based methods [12, 15, 44]. Recently, attention has shifted to hybrid methods that combine pseudo-labeling and consistency regularization to ensure that the predicted labels of different perturbations align

\*Joint Corresponding Author

with one another [28, 37, 47, 53]. This popular paradigm is often referred to as X-Match, as shown in Fig. 1, inspired by the seminal work FixMatch [37], where X represents the specific variation of the method. A common challenge with PL is the presence of incorrect labels, especially in the early stages of training. To address this, FixMatch introduced a fixed confidence threshold to keep high-confidence pseudo-labels, which are then treated as ground truth for consistency learning. Following that, the X-Match series of works considers: 1) dynamically adjusting the confidence threshold based on the model’s learning status; 2) developing class-wise confidence thresholds to account for different learning difficulty across different classes. However, relying solely on the model’s confidence makes them difficult to accurately assess the model’s current state and establish reliable thresholds, especially when labeled data is too scarce to provide sufficient supervision. This often results in noisy training and poor generalization.

As an effective dataset diagnostic tool, data maps [40] identifies high-quality data (i.e., easy-to-learn instances and ambiguous instances) and noisy data (i.e., hard-to-learn instances) by using two training dynamics measures, i.e., confidence and variability, defined as the mean and standard deviation of the predicted probabilities for the true label over training epochs. Unfortunately, the calculation of both variability and confidence depends on the ground truth labels, which are not accessible in the context of SSL. To expand the concept of data maps to SSL, we introduce a novel metric named **Count-Gap (CG)** for the first time and propose a novel SSL model, CGMatch. Specifically, CG quantifies the difference between the number of the most frequently predicted pseudo-labels and the second most frequent ones up to the current iteration, offering a complementary perspective for assessing the quality of the model outputs. Figure 2 presents the CG data map for the CIFAR10 dataset, constructed by running FixMatch with only 40 labeled data over 2048 iterations, followed by applying the data maps tool [40]. It can be observed that ambiguous samples tend to have higher CG values than hard-to-learn samples, suggesting that CG is well-suited for distinguishing various types of unlabeled examples. In CGMatch, we first propose the **Fine-grained Dynamic Selection (FDS)** scheme, leveraging both CG and confidence (i.e., the highest predicted probability of the model) to dynamically group the unlabeled data into easy-to-learn set, ambiguous set and hard-to-learn set as the training progresses. These three subsets, each with distinct characteristics, are then integrated into model training via different regularization strategies, mitigating the negative impact of noisy labels on model training. The overall framework of CGMatch is shown in Fig. 3. Despite its simplicity, we demonstrate that CGMatch achieves the competitive image classification results on the most widely-studied SSL benchmarks. We further analyze

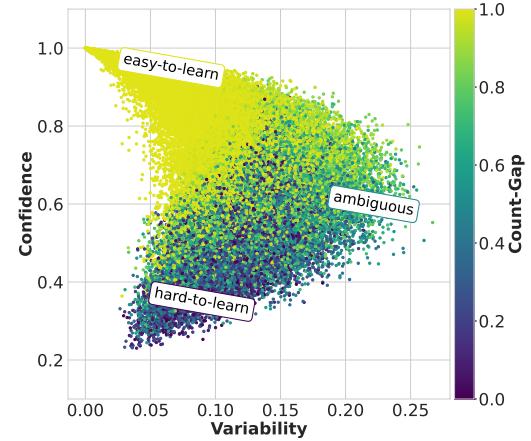


Figure 2. Data map for the CIFAR10 dataset with 40 labels regarding Count-Gap. The x-axis represents variability, and the y-axis reflects confidence. Colors indicate the **Count-Gap (CG)**. In this data map, the top-left corner (low variability, high confidence) highlights **easy-to-learn** examples, the bottom-left corner (low variability, low confidence) identifies **hard-to-learn** examples, while examples on the right (high variability) are categorized as **ambiguous**. It is evident that Count-Gap is effective in distinguishing these three types of subsets within the context of SSL.

the advantages of CGMatch and validate the effectiveness of CG from different views.

In summary, the major contributions of this paper are outlined below.

- We introduce the notion of Count-Gap into semi-supervised learning for the first time and demonstrate it’s effectiveness in identifying unlabeled data that contribute to model training.
- We propose the FDS scheme, utilizing two key metrics, Count-Gap and confidence, to effectively categorize unlabeled data into three distinct subsets following a progressively challenging curriculum. We further apply different regularization techniques to each unlabeled subset, mitigating the contamination of incorrect pseudo-labels to model learning.
- We carry out extensive experiments to demonstrate that CGMatch delivers highly competitive performance compared to existing semi-supervised learning methods on various standard SSL benchmark datasets. Notably, CGMatch significantly outperforms others when the number of labeled data is particularly limited.

## 2. CGMatch

### 2.1. Preliminaries

In the semi-supervised classification task, we aim to learn from a few labeled data  $\mathcal{D}_L = \{x_i, y_i\}_{i=1}^{N_L}$  and a large amount of unlabeled data  $\mathcal{D}_U = \{u_i\}_{i=1}^{N_U}$ , then the model

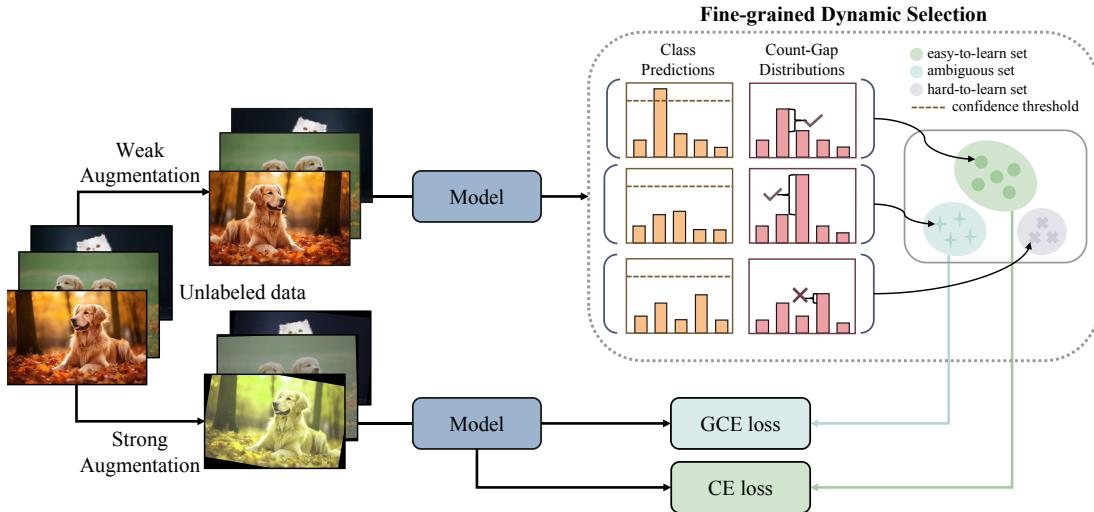


Figure 3. The framework of the proposed CGMatch. First, one network takes the unlabeled samples with diverse augmentations as input and outputs the corresponding prediction distributions, which is necessary for consistency regularization. Then, a fine-grained dynamic selection (FDS) strategy is designed by taking the class predictions and Count-Gap distributions of weakly-augmented versions into account, which is utilized to divide the unlabeled data into three subsets: easy-to-learn set, ambiguous set, and hard-to-learn set. Finally, different regularization techniques are employed to involve easy-to-learn samples and ambiguous samples into model training, aiding both model optimization and generation.

$f$  with network parameter  $\theta$  can be trained by minimizing a joint objective  $\mathcal{L} = \mathcal{L}_s + \lambda \mathcal{L}_u$ , where  $\mathcal{L}_s$  and  $\mathcal{L}_u$  are the supervised loss on labeled data and unsupervised loss on unlabeled data respectively, and  $\lambda$  is the loss weight. Specifically, in a training batch containing  $B_l$  labeled samples and  $B_u$  unlabeled samples, we denote  $\mathcal{L}_s$  as the cross-entropy loss on labeled batch:

$$\mathcal{L}_s = \frac{1}{B_l} \sum_{i=1}^{B_l} H(y_i, p_\theta(y|\omega(x_i))) \quad (1)$$

where  $\omega(\cdot)$  is a stochastic data augmentation function.

Most recent research [47, 53] has proven that a superior performance can be achieved by combining pseudo-labeling and consistency regularization. FixMatch [37], as a representative algorithm, generates pseudo-labels for unlabeled images after applying weak augmentation, such as random cropping. It then enforces consistency by aligning the predictions of strongly-augmented inputs with these generated pseudo-labels, provided they meet a certain confidence level, i.e., the confidence exceeds a predefined threshold. So  $\mathcal{L}_u$  can be formed as:

$$\mathcal{L}_u = \frac{1}{B_u} \sum_{i=1}^{B_u} \mathbb{I}(\max(p_i) > \tau) H(\tilde{y}_i, p_\theta(y|\Omega(u_i))) \quad (2)$$

where  $p_i = p_\theta(y|\omega(u_i))$  is the prediction distribution of the weakly-augmented version, while  $p_\theta(y|\Omega(u_i))$  corresponds to the prediction distribution from the same unlabeled sample with strong augmentation  $\Omega(\cdot)$ ;  $\tilde{y}_i = \text{argmax}(p_i)$  is the

hard target,  $\tau$  is the predefined confidence threshold, and  $\mathbb{I}(\cdot)$  is the indicator function to filter low-confidence unlabeled examples.

As suggested by previous research [5], some challenging examples, despite exhibiting low-confidence predictions, are still highly likely to be correctly pseudo-labeled, which are designated as “ambiguous samples” in this work. Discarding these samples not only results in the waste of unlabeled data but also reduces the learning efficiency and generalization capability of the classification model to some extent. Therefore, we propose a simple yet effective SSL model to address the aforementioned challenge.

## 2.2. Fine-grained Dynamic Selection (FDS)

We present a scheme to further explore the remaining unlabeled data filtered by confidence thresholding, referred to as FDS. First, FDS considers the unlabeled data with the highest class prediction probability exceeding the confidence threshold as **easy-to-learn samples**. Similar to the selection approach described in Eq. (2), we denote easy-to-learn set at iteration  $t$  as  $\mathcal{U}_e^t \subseteq \mathcal{D}_U$ :

$$\mathcal{U}_e^t = \{u_i, \tilde{y}_i^t | \max(p_i^t) \geq \tau_e^t\} \quad (3)$$

where  $\tau_e^t$  indicates the confidence threshold at iteration  $t$ . Then, FDS identifies **ambiguous samples** by tracking the variation in the prediction count of pseudo-labels on the unlabeled data over the iterations. Assume that  $Q_i^t = [q_i^t(C_1), q_i^t(C_2), \dots, q_i^t(C_K)]$  represents the queue recording the cumulative prediction count of each class

---

**Algorithm 1** CGMatch

---

**Require:** Labeled data  $\mathcal{D}_L = \{x_i, y_i\}_{i=1}^{N_L}$ ; Unlabeled data  $\mathcal{D}_U = \{u_i\}_{i=1}^{N_U}$ ; Number of total iterations  $T$ ;  
Number of warm-up iterations  $t_0$ ; Number of classes  $K$ ; Model  $f$  with network parameter  $\theta$ ;  
Strong augmentation function  $\Omega(\cdot)$ ; Weak augmentation function  $\omega(\cdot)$ .

- 1: Initialize model parameter  $\theta$  randomly;
- 2: Warm-up the model on only labeled data for  $t_0$  iterations and initialize the queue  $Q_i = [q_i(C_1), q_i(C_2), \dots, q_i(C_K)]$  for each unlabeled sample  $u_i$  with the cumulative number of predictions for each class obtained over the last 1000 iterations;
- 3: **for**  $t = t_0, \dots, T$  **do**
- 4:   Draw a training batch containing  $B_l$  labeled samples and  $B_u$  unlabeled samples;
- 5:   Update  $Q_i^t$  according to the generated pseudo-label  $\tilde{y}_i^t$  and calculate the value of  $\text{CG}_i^t$  for each  $u_i$  in the current batch;
- 6:   Determine the dynamic confidence threshold  $\tau_e^t$  and ambiguity threshold  $\tau_a^t$  using Eqs. (5) and (6);
- 7:   Select two types of subsets benefiting the model training  $\mathcal{U}_e^t$  and  $\mathcal{U}_a^t$  using Eqs. (3) and (4);
- 8:   Minimize the total loss  $\mathcal{L}$  via Eq. (9).
- 9: **end for**

---

with respect of the unlabeled sample  $u_i$  from the beginning of training to iteration  $t$ , where  $K$  indicates the number of unique classes and  $C_j$  is the  $j$ -th class label. The value of Count-Gap on  $u_i$  at iteration  $t$  can be computed as  $\text{CG}_i^t = k_1 - k_2$ , where  $k_1$  and  $k_2$  are the number of the most, i.e.,  $k_1 = \max(Q_i^t)$ , and the second most predicted pseudo-labels stored in the  $Q_i^t$ , i.e.,  $k_2 = \max(Q_i^t \setminus k_1)$ , respectively. Upon both metrics, we can obtain the ambiguous set at iteration  $t$ ,  $\mathcal{U}_a^t \subseteq \mathcal{D}_U - \mathcal{U}_e^t$ , as follows:

$$\mathcal{U}_a^t = \{u_i, \tilde{y}_i^t | \max(p_i^t) < \tau_e^t\} \cap \{u_i, \tilde{y}_i^t | \text{CG}_i^t \geq \tau_a^t\} \quad (4)$$

where  $\tau_a^t$  is the ambiguity threshold at iteration  $t$ . Both  $\tau_e^t$  and  $\tau_a^t$  can be dynamically adjusted throughout the training process. The choice of dynamic thresholding methods is flexible, and they can be easily integrated into our approach. In this work, we use the Exponential Moving Average (EMA) for dynamically adjusting the thresholds. Specifically, the EMA with a momentum  $m$ , which pays more attention on recent iterations, is defined as follows:

$$\begin{aligned} \tau_e^t &= m\tau_e^{t-1} + (1-m)\mu_e^t, \\ \tau_a^t &= m\tau_a^{t-1} + (1-m)\mu_a^t \end{aligned} \quad (5)$$

where  $\mu_e^t$  and  $\mu_a^t$ , estimated over the batch  $B_u$  in the current iteration  $t$ , are given by:

$$\begin{aligned} \mu_e^t &= \frac{1}{B_u} \sum_{i=1}^{B_u} \max(p_i^t), \\ \mu_a^t &= \frac{1}{B_u} \sum_{i=1}^{B_u} \text{CG}_i^t \end{aligned} \quad (6)$$

### 2.3. Learning From Selections

Based on the subset selections of FDS, we apply different training objects to learn separately from easy-to-learn samples and ambiguous samples. For the easy-to-learn set  $\mathcal{U}_e$ ,

we adopt the conventional cross-entropy (CE) loss:

$$\mathcal{L}_e = \frac{1}{|\mathcal{U}_e|} \sum_{i=1}^{|\mathcal{U}_e|} H(\tilde{y}_i, p_\theta(y|\Omega(u_i))) \quad (7)$$

where  $|\mathcal{U}_e|$  represents the number of easy-to-learn samples. For ambiguous set  $\mathcal{U}_a$ , we employ a more resilient regularization strategy named generalized cross-entropy (GCE) loss [54] that can avoid label noise from misleading the model and is easy to optimize. The associated loss function  $\mathcal{L}_a$  is described below:

$$\mathcal{L}_a = \frac{1}{|\mathcal{U}_a|} \sum_{i=1}^{|\mathcal{U}_a|} \left( \frac{1 - p_w(\tilde{y}_i; u_i)^q}{q} + \frac{1 - p_s(\tilde{y}_i; u_i)^q}{q} \right) \quad (8)$$

where  $|\mathcal{U}_a|$  is the size of the ambiguous set;  $p_w(\tilde{y}_i; u_i)$  and  $p_s(\tilde{y}_i; u_i)$  correspond to the prediction confidence of weakly-augmented and strongly-augmented views of image  $u_i$  given pseudo-label  $\tilde{y}_i$ , respectively. The hyper-parameter  $q \in (0, 1]$  is set to 0.7 as recommended in [54]. The overall loss in CGMatch can be summarized to the following form:

$$\mathcal{L} = \mathcal{L}_s + \lambda_e \cdot \mathcal{L}_e + \lambda_a \cdot \mathcal{L}_a \quad (9)$$

where  $\lambda_e$  and  $\lambda_a$  are the balancing coefficients of  $\mathcal{L}_e$  and  $\mathcal{L}_a$ , respectively. The full algorithm of CGMatch is shown in Algorithm 1.

## 3. Experiments

### 3.1. Settings

To evaluate the performance of the proposed method, we conducted experiments on several benchmark datasets: CIFAR10/100 [16], SVHN [29], and STL10 [9] with different quantities of labeled examples. In our studies, we specifically focused on settings where the number of labeled samples is relatively small, i.e., 4, 10, or 25 labeled samples

Dataset	CIFAR10			CIFAR100			SVHN			STL10		
# Labels/Class	4	10	25	4	10	25	4	10	25	4	10	25
PseudoLabel	76.29 <sub>±0.08</sub>	65.18 <sub>±0.52</sub>	48.28 <sub>±2.10</sub>	87.15 <sub>±0.87</sub>	76.06 <sub>±1.12</sub>	59.09 <sub>±0.61</sub>	75.95 <sub>±3.39</sub>	27.49 <sub>±1.17</sub>	16.60 <sub>±1.13</sub>	73.81 <sub>±0.82</sub>	68.13 <sub>±2.02</sub>	49.21 <sub>±1.54</sub>
MeanTeacher	76.93 <sub>±2.29</sub>	55.80 <sub>±4.73</sub>	56.06 <sub>±2.03</sub>	90.34 <sub>±0.65</sub>	65.48 <sub>±2.37</sub>	61.13 <sub>±0.57</sub>	81.94 <sub>±1.33</sub>	13.73 <sub>±0.18</sub>	25.10 <sub>±3.17</sub>	70.94 <sub>±0.92</sub>	64.48 <sub>±0.83</sub>	44.51 <sub>±2.02</sub>
MixMatch	70.67 <sub>±1.25</sub>	53.59 <sub>±1.81</sub>	37.28 <sub>±0.61</sub>	79.95 <sub>±0.29</sub>	64.31 <sub>±1.23</sub>	49.58 <sub>±0.62</sub>	79.63 <sub>±5.78</sub>	24.78 <sub>±3.14</sub>	3.71 <sub>±0.20</sub>	68.78 <sub>±1.54</sub>	55.75 <sub>±0.73</sub>	42.34 <sub>±0.99</sub>
ReMixMatch	14.50 <sub>±2.58</sub>	11.22 <sub>±1.20</sub>	9.21 <sub>±0.55</sub>	57.10 <sub>±0.00</sub>	42.79 <sub>±0.53</sub>	34.77 <sub>±0.32</sub>	31.27 <sub>±18.79</sub>	6.15 <sub>±0.29</sub>	6.38 <sub>±1.09</sub>	46.67 <sub>±4.27</sub>	30.26 <sub>±5.37</sub>	21.49 <sub>±1.75</sub>
FixMatch	8.33 <sub>±1.41</sub>	7.69 <sub>±2.35</sub>	4.97 <sub>±0.05</sub>	53.37 <sub>±2.15</sub>	42.75 <sub>±1.04</sub>	34.29 <sub>±0.43</sub>	3.56 <sub>±2.09</sub>	<b>2.01</b> <sub>±0.05</sub>	<b>2.04</b> <sub>±0.02</sub>	40.81 <sub>±4.64</sub>	16.40 <sub>±5.32</sub>	9.81 <sub>±1.10</sub>
FlexMatch	5.21 <sub>±0.36</sub>	5.24 <sub>±0.30</sub>	5.00 <sub>±0.05</sub>	50.15 <sub>±1.51</sub>	39.48 <sub>±0.70</sub>	33.35 <sub>±0.33</sub>	3.36 <sub>±0.37</sub>	3.58 <sub>±0.59</sub>	5.02 <sub>±1.20</sub>	24.41 <sub>±7.39</sub>	<u>12.14</u> <sub>±0.77</sub>	<u>8.71</u> <sub>±0.17</sub>
SoftMatch	5.06 <sub>±0.02</sub>	5.08 <sub>±0.03</sub>	<u>4.84</u> <sub>±0.10</sub>	49.64 <sub>±1.46</sub>	38.93 <sub>±0.40</sub>	33.05 <sub>±0.05</sub>	2.88 <sub>±0.93</sub>	2.29 <sub>±0.16</sub>	2.15 <sub>±0.05</sub>	<u>22.36</u> <sub>±5.37</sub>	<b>11.59</b> <sub>±2.60</sub>	<b>8.01</b> <sub>±0.32</sub>
FreeMatch	4.97 <sub>±0.09</sub>	5.01 <sub>±0.17</sub>	4.85 <sub>±0.10</sub>	49.24 <sub>±2.16</sub>	39.18 <sub>±0.95</sub>	<u>32.79</u> <sub>±0.21</sub>	3.79 <sub>±0.03</sub>	4.18 <sub>±1.86</sub>	4.09 <sub>±0.66</sub>	26.58 <sub>±3.23</sub>	12.85 <sub>±0.77</sub>	8.78 <sub>±0.88</sub>
FullFlex	6.21 <sub>±1.32</sub>	5.07 <sub>±0.24</sub>	4.98 <sub>±0.04</sub>	48.35 <sub>±1.18</sub>	38.86 <sub>±0.39</sub>	32.85 <sub>±0.10</sub>	<u>2.84</u> <sub>±0.23</sub>	3.71 <sub>±1.51</sub>	4.63 <sub>±1.26</sub>	33.28 <sub>±0.85</sub>	14.75 <sub>±1.99</sub>	8.83 <sub>±0.89</sub>
<b>CGMatch (Ours)</b>	<b>4.87</b> <sub>±0.18</sub>	<b>4.92</b> <sub>±0.21</sub>	<b>4.80</b> <sub>±0.07</sub>	<b>47.55</b> <sub>±2.48</sub>	<b>38.68</b> <sub>±0.51</sub>	<b>32.51</b> <sub>±0.07</sub>	<b>2.39</b> <sub>±0.06</sub>	<u>2.18</u> <sub>±0.06</sub>	<u>2.13</u> <sub>±0.05</sub>	<b>22.19</b> <sub>±1.8</sub>	16.23 <sub>±2.39</sub>	9.48 <sub>±0.60</sub>

Table 1. Top-1 error rate (%) on CIFAR10, CIFAR100, SVHN, and STL10 of 3 different random seeds. The best results are highlighted in **bold** and the second-best results are underlined.

per class, mimicking scenarios with very limited supervision. For convenience, we use  $D$ - $L$  to denote the case where dataset  $D$  has  $L$  labels per class. To maintain fairness in the comparisons, all methods were trained and evaluated using the unified codebase USB [46] with the same backbones and hyperparameters<sup>1</sup>. We used WRN-28-2 for CIFAR10/100 and SVHN, and WRN-VAR-37-2 for STL10. We adopted SGD optimizer with the initial learning rate of 0.03 and a momentum of 0.9, then the learning rate was adjusted using a cosine learning rate decay schedule with the total iteration number  $T$  of  $2^{20}$ . Besides, in a batch, the number of labeled data is 64, and the number of unlabeled data is 7 times larger by following previous work [5, 7, 37, 47]. We ran each method with three random seeds and report the average top-1 error rate.

In CGMatch, we first warmed-up the model on labeled data for 2048 iterations (*i.e.*,  $t_0 = 2048$ ) and retained Count-Gap values from the most recent 1000 iterations to initialize the Count-Gap queue for each unlabeled sample. In the subsequent training stage, we set  $\lambda_e$  to 1 for the cross-entropy loss associated with  $\mathcal{U}_e$ ; and  $\lambda_a$  for the generalized cross-entropy loss of  $\mathcal{U}_a$  was set to increase gradually during the early training phase, when the model’s predictions are still unstable, and to become more sensitive as the model improves in the mid and late stages. This ensures a balanced adjustment of the impact of ambiguous data on model training. To achieve this, we chose a non-linear increasing function with the desired properties for our experiments:  $\lambda_a = \left(\frac{t-t_0}{T-t_0}\right)^2$ , where  $t$  is the current iteration step. For SVHN, as processed in FreeMatch [47], we restricted the confidence threshold to the range of [0.9, 0.95] during the whole training process.

<sup>1</sup>We re-implement the FullFlex algorithm in the USB framework.

### 3.2. Main Results

Table 1 presents the classification results of all compared methods. Our method performs the best on almost all cases and achieves the second-best results on SVHN-4 and SVHN-10. We make several observations in more detail, and discuss them below.

- CGMatch outperforms the competing methods in 8 out of 12 settings, demonstrating its effectiveness in selecting unlabeled samples that aid in model optimization and generalization, therefore further improves the classification performance.
- CGMatch achieves better performance than FlexMatch in 10 out of 12 settings, particularly on settings with relatively small labeled data. For example, on more realistic dataset, CIFAR100-4, CGMatch achieves significant improvement of 2.60%. This indicates that relying solely on dynamic confidence thresholding techniques makes it challenging to effectively utilize unlabeled data, especially when supervision is scarce, highlighting the necessity of Count-Gap. Additionally, this illustrates the capability and potential of CGMatch for deployment in real-world applications.
- Compared to the latest counterparts, SoftMatch and FullFlex, CGMatch delivers better classification results, except on STL10-10 and STL10-25. This implies a more effective approach should be proposed to incorporate hard-to-learn samples into training.
- CGMatch shows a gap compared to the best-performing method, SoftMatch, on STL10-10 and STL10-25. We argue that a possible reason is that STL10 is an open-set dataset, containing a large number of unlabeled samples from classes not present in the labeled data. After experiencing the warm-up phase, our model learns more from the known classes in the labeled data on STL10-10 and STL10-25 than on STL10-4, resulting in a larger calibration error. This makes it challenging for Count-Gap

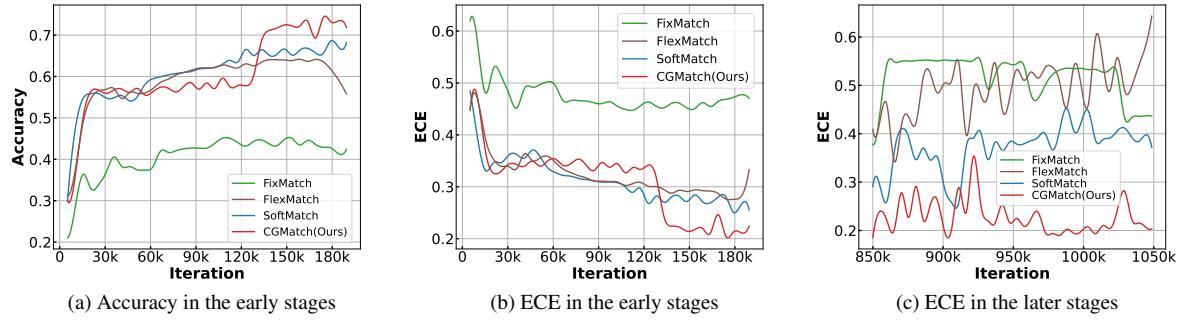


Figure 4. The learning efficiency and the model calibration error of different methods on STL10-4. (a) and (b) demonstrate the evaluation accuracy and ECE within the first 180K iterations. (c) depicts how ECE evolves over the final 200k iterations.

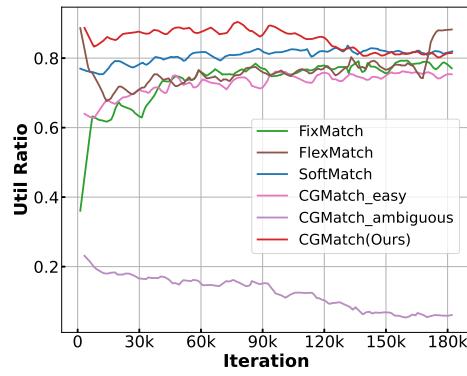


Figure 5. Comparison of the quantity of pseudo-labels on STL10-4. *CGMatch*, *CGMatch\_ambiguous*, and *CGMatch\_easy* refer to the changing trends in the total number of unlabeled samples, the number of ambiguous samples, and the number of easy-to-learn samples involved in model training, respectively.

to generalize to other unseen classes during the training phase, ultimately leading to a drop in performance. This can be one limitation of CGMatch, which is subject to our future work.

### 3.3. Qualitative Analysis

In this section, we provide a qualitative comparison on STL10-4 of FixMatch [37], FlexMatch [53], SoftMatch [5], and CGMatch from different aspects, and analyze why and how CGMatch works, as detailed in Figs. 4 and 5. We employed two valuable metrics: the evaluation accuracy and the Expected Calibration Error (ECE) [11], and plotted their curves over iterations in Fig. 4 to qualify the learning status and performance of the above models. Notably, ECE measures how well a model’s estimated “probabilities” match the true (observed) probabilities by taking a weighted average over the absolute difference between accuracy and confidence, and a lower value indicates a better-calibrated model. Additionally, we track the utilization ratio of unlabeled data across iterations and present their curves in

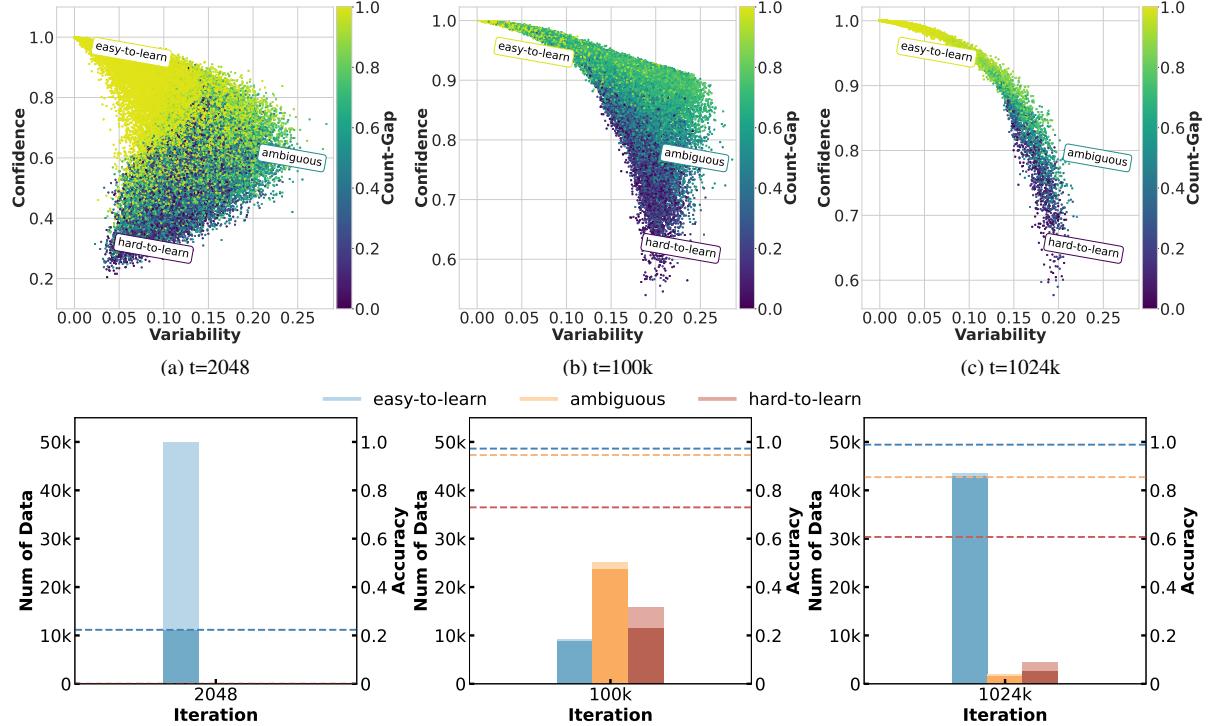
Fig. 5. The terms *CGMatch*, *CGMatch\_ambiguous*, and *CGMatch\_easy* denote the changing trends in the total number of unlabeled samples, the number of ambiguous samples, and the number of easy-to-learn samples involved in model training, respectively.

As illustrated in Fig. 5, in contrast to these SOTA methods, CGMatch integrates not only high-confidence samples with the highest predicted probability surpassing the current confidence threshold, but also ambiguous samples into the training process, resulting in a higher utilization ratio of unlabeled data at the early stage of training. Consequently, this accelerates the training efficiency and achieves better calibration, as demonstrated in Figs. 4a to 4c, further highlighting the effectiveness of FDS scheme. For example, to reach an evaluation accuracy of 0.7, CGMatch requires only 135k iterations, significantly fewer than other methods. Then, as the model becomes more proficient and confident, ambiguous samples progressively transition into either easy-to-learn or hard-to-learn categories, leading to a reduction in the number of ambiguous samples.

### 3.4. Effectiveness Analysis of Count-Gap

In this section, we conducted experiments to validate the effectiveness of Count-Gap. Specifically, following the data maps tool proposed in [40], we visualized the data maps regarding Count-Gap at the 2048, 100k and 1024k training iterations of the proposed CGMatch, conducted on the CIFAR10-4. Figures 6a to 6c show that as training progresses, the boundaries among easy-to-learn data, ambiguous data, and hard-to-learn data become clear, demonstrating that Count-Gap has the ability to partition unlabeled data in the context of SSL.

Correspondingly, based on the dynamic confidence threshold and ambiguity threshold, we further plotted the distribution of these three types of data, along with the number of correctly predicted instances within them, across different training iterations. As illustrated in Fig. 6d, Count-Gap performs well in selecting high-quality unlabeled data, highlighting its effectiveness once again. At the beginning



(d) The number and prediction accuracy of easy-to-learn data, ambiguous data, and hard-to-learn data at different training iterations.

Figure 6. Data maps and the transformation among three subsets at different training iterations of CGMatch, conducted on CIFAR10-4. At each iteration in (d), light-colored and dark-colored bars represent the total number of data and the number of correctly predicted data within each category, respectively. Additionally, the dashed lines in different colors depict the prediction accuracy for each category.

of the training stage (i.e.  $t = 2048$ ), the low confidence threshold groups all unlabeled data into an easy-to-learn set, and the model’s prediction accuracy is relatively low. As the model progressively leans from the data (i.e.  $t = 100k$  and  $t = 1024k$ ), Count-Gap enables increasingly precise data partitioning. This ensures that high-accuracy easy-to-learn and ambiguous samples are incorporated into the training, while hard-to-learn samples, often linked to incorrect predictions, are excluded, thereby reducing the negative impact of noisy labels on model performance.

### 3.5. Ablation Study

**Dynamic Thresholding Techniques.** To verify the effect of different thresholding techniques, we consider generating the confidence threshold  $\tau_e^t$  defined in our method in an adaptive or global manner, and report the mean and variance of top-1 error rate of four datasets in Tab. 2.

- **Self-Adaptive Thresholding (SAT):** Proposed in FreeMatch [47], which can automatically adjust the confidence threshold for each class by leveraging the model predictions during training.
- **Global Thresholding (GT):** Used in SoftMatch [5], only estimated as the exponential moving average (EMA) of the confidence from unlabeled batch at each training step. As illustrated in Tab. 2, GT demonstrates superior perfor-

mance across the majority of benchmark datasets when compared to SAT. This is because accurate estimation for class-wise thresholds requires many predictions for each class, which is not available for SAT in our settings.

## 4. Related Work

### 4.1. Self-training and Pseudo-labeling

Self-training is a commonly used technique in semi-supervised learning, centered around the iterative use of model predictions to enlarge the training dataset [23, 32, 36, 49]. A variation of this approach, known as pseudo-labeling [19], involves converting model predictions into hard artificial labels for further training. Although they offer an advantage of simplicity, both of them are prone to generating noisy pseudo-labels, particularly in the early stages of training, leading to poor performance. In recent years, a variety of works mainly focus on improving pseudo-labeling by enhancing the accuracy and robustness of generated pseudo-labels [31, 37, 48] or further taking the utilization ratio of unlabeled data into account [5, 7, 39, 47, 51, 53].

- **Quality-aware Pseudo-labeling** Confidence thresholding techniques [37, 48] ensure the quality of pseudo-labels by only retaining unlabeled data with confidence

Dataset	CIFAR10			CIFAR100			SVHN			STL10		
# Labels/Class	4	10	25	40	10	25	4	10	25	4	10	25
<i>Self-Adaptive Thresholding</i>	5.19 $\pm$ 0.22	4.98 $\pm$ 0.24	4.83 $\pm$ 0.06	48.01 $\pm$ 3.06	38.83 $\pm$ 0.86	33.16 $\pm$ 0.31	3.59 $\pm$ 0.00	4.03 $\pm$ 0.74	5.57 $\pm$ 1.72	26.75 $\pm$ 4.12	16.38 $\pm$ 2.19	10.68 $\pm$ 1.00
<i>Global Thresholding</i>	<b>4.87<math>\pm</math>0.18</b>	<b>4.92<math>\pm</math>0.21</b>	<b>4.80<math>\pm</math>0.07</b>	<b>47.55<math>\pm</math>2.48</b>	<b>38.68<math>\pm</math>0.51</b>	<b>32.51<math>\pm</math>0.07</b>	<b>2.39<math>\pm</math>0.06</b>	<b>2.18<math>\pm</math>0.06</b>	<b>2.13<math>\pm</math>0.05</b>	<b>22.19<math>\pm</math>1.81</b>	<b>16.23<math>\pm</math>2.39</b>	<b>9.48<math>\pm</math>0.60</b>

Table 2. Top-1 error rate (%) with different dynamic thresholding techniques on CIFAR10, CIFAR100, SVHN, and STL10. The best results are highlighted in **bold**.

levels exceeding the confidence threshold. Both FixMatch [37] and UDA [48] employ a fixed predefined threshold to select high-quality pseudo-labels. The difference between FixMatch and UDA is that FixMatch adopts one-hot ‘hard’ labels instead of sharpened ‘soft’ pseudo-labels with a temperature. Rather than solely relying on the model’s confidence at the current iteration to decide if an unlabeled example should be used for training or not, some works involve additional selection criteria. For example, UPS [31] combines both the confidence and uncertainty of a network prediction, with the uncertainty measured through MC-Dropout [10], significantly reducing noise in the training process. However, the aforementioned methods inevitably face the issue of low data utilization.

- **Quantity-aware Pseudo-labeling** Nowadays, the quantity-quality trade-off of pseudo-labeling become a trending topic in semi-supervised learning, with dynamic thresholding techniques [4, 39, 47, 51, 53] or exploration strategies of the low-confidence unlabeled data [5, 7] being proposed. Dash [51] proposes to gradually grow the fixed global threshold as the training progresses with an ad-hoc threshold adjusting scheme to improve the utilization of unlabeled data. FlexMatch [53] considers the varying learning difficulties across different classes and defines class-wise thresholds derived from a pre-defined fixed global threshold; FreeMatch [47] designs a self-adaptive threshold-adjusting scheme which can reflect the model’s learning status, leading to a significant performance improvement. MarginMatch [39] utilizes the model’s training dynamics on unlabeled data to improve pseudo-label data quality, while enabling the use of a large set of unlabelled data for learning. Additionally, SoftMatch [5] and FullFlex [7] leverage low-confidence unlabeled samples by weighting all the unlabeled samples or assigning the additional negative pseudo-label for all unlabeled data. Compared to these existing methods, our method achieves a better balance between quantity and quality, particularly in the challenging scenarios with only a few labeled data.

## 4.2. Consistency Regularization

Consistency regularization draws inspiration from the idea that a classifier should output the same class distribution

for an unlabeled example even when it is subject to random perturbations such as data augmentation [2, 3, 37], dropout [33] or adversarial perturbations [25], which serves as a valuable component in most successful SSL approaches [2, 3, 7, 37, 48]. Current state-of-the-art methods always (e.g., [2, 3, 5, 37]) adhere to a paradigm where training targets for unlabeled data are first generated using weakly-augmented strategies, followed by enforcing prediction consistency against the strongly-augmented versions. Similar to these approaches, our approach uses the same combination of weak and strong data augmentations.

## 5. Conclusion

In this paper, we concentrated on the SSL settings with relatively small labels, which more accurately depict the tough yet common scenarios encountered in real-world applications. We have presented a novel semi-supervised model named CGMatch, designed to tackle this challenge by introducing a unique metric called Count-Gap (CG), which assesses the quality of the generated pseudo-labels from an alternative perspective. Based on the Count-Gap and confidence, we proposed a fine-grained dynamic selection (FDS) strategy to categorize the unlabeled data into three subsets with pronounced characteristics during the model training with the help of dynamic thresholding techniques. Subsequently, we adopted various regularization techniques tailored to each subset to effectively manage noise in the learning process. CGMatch achieves competitive performance in most scenarios. Moreover, it is a versatile approach applicable to most SSL frameworks and adds nearly no extra hyperparameters. In the future, we will aim to apply Count-Gap more effectively in open-set SSL to overcome the limitation of current method. Also, we will explore how to leverage hard-to-learn data to further boost the performance of SSL, rather than discarding them.

## 6. Acknowledgments

We would like to thank the anonymous reviewers for their valuable and helpful comments. This work is supported by the National Key Research and Development Program of China (No.2023YFF0905400), the National Natural Science Foundation of China (No.U2341229) and the Reform Commission Foundation of Jilin Province (No.2024C003).

## References

- [1] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. *Advances in neural information processing systems*, 27, 2014. 1
- [2] David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019. 8
- [3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019. 8
- [4] David Berthelot, Rebecca Roelofs, Kihyuk Sohn, Nicholas Carlini, and Alexey Kurakin. Adamatch: A unified approach to semi-supervised learning and domain adaptation. In *International Conference on Learning Representations*, 2022. 8
- [5] Hao Chen, Ran Tao, Yue Fan, Yidong Wang, Jindong Wang, Bernt Schiele, Xing Xie, Bhiksha Raj, and Marios Savvides. Softmatch: Addressing the quantity-quality tradeoff in semi-supervised learning. In *The Eleventh International Conference on Learning Representations*, 2023. 3, 5, 6, 7, 8
- [6] Jiaao Chen, Zichao Yang, and Diyi Yang. MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157. Association for Computational Linguistics, 2020. 1
- [7] Yuhao Chen, Xin Tan, Borui Zhao, Zhaowei Chen, Renjie Song, Jiajun Liang, and Xuequan Lu. Boosting semi-supervised learning by exploiting all unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7548–7557, 2023. 5, 7, 8
- [8] Yanwen Chong, Yun Ding, Qing Yan, and Shaoming Pan. Graph-based semi-supervised learning: A review. *Neurocomputing*, 408:216–230, 2020. 1
- [9] 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, 2011. 4
- [10] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016. 8
- [11] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017. 6
- [12] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017. 1
- [13] Wanwei He, Yinpei Dai, Yinhe Zheng, Yuchuan Wu, Zheng Cao, Dermot Liu, Peng Jiang, Min Yang, Fei Huang, Luo Si, et al. Galaxy: A generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection. In *Proceedings of the AAAI conference on artificial intelligence*, pages 10749–10757, 2022. 1
- [14] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 27, 2014. 1
- [15] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 1
- [16] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009. 4
- [17] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016. 1
- [18] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*, 2017. 1
- [19] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, page 896, 2013. 1, 7
- [20] Chongxuan Li, Taufik Xu, Jun Zhu, and Bo Zhang. Triple generative adversarial nets. *Advances in neural information processing systems*, 30, 2017. 1
- [21] Changchun Li, Ximing Li, and Jihong Ouyang. Semi-supervised text classification with balanced deep representation distributions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5044–5053, 2021. 1
- [22] Yang Li, Quan Pan, Suhang Wang, Haiyun Peng, Tao Yang, and Erik Cambria. Disentangled variational auto-encoder for semi-supervised learning. *Inf. Sci.*, 482(C):73–85, 2019. 1
- [23] David McClosky, Eugene Charniak, and Mark Johnson. Effective self-training for parsing. In *Proceedings of the human language technology conference of the NAACL, main conference*, pages 152–159, 2006. 7
- [24] Rahul Mitra, Nitesh Bharadwaj Gundavarapu, Abhishek Sharma, and Arjun Jain. Multiview-consistent semi-supervised learning for 3d human pose estimation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6906–6915, 2019. 1
- [25] Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. Adversarial training methods for semi-supervised text classification. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. 8
- [26] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018. 1
- [27] Lewis Moffat and David T Jones. Increasing the accuracy of single sequence prediction methods using a deep semi-supervised learning framework. *Bioinformatics*, 37(21):3744–3751, 2021. 1

- [28] Islam Nassar, Samitha Herath, Ehsan Abbasnejad, Wray Buntine, and Gholamreza Haffari. All labels are not created equal: Enhancing semi-supervision via label grouping and co-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7241–7250, 2021. 2
- [29] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop*, page 4, 2011. 4
- [30] Shaojun Pan, Xing-Ming Zhao, and Luis Pedro Coelho. Semibin2: self-supervised contrastive learning leads to better mags for short-and long-read sequencing. *Bioinformatics*, 39(Supplement\_1):i21–i29, 2023. 1
- [31] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *International Conference on Learning Representations*, 2021. 7, 8
- [32] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. In *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1*, pages 29–36, 2005. 7
- [33] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, 29, 2016. 8
- [34] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, 29, 2016. 1
- [35] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 1
- [36] H. Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965. 7
- [37] 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. 2, 3, 5, 6, 7, 8
- [38] Zixing Song, Xiangli Yang, Zenglin Xu, and Irwin King. Graph-based semi-supervised learning: A comprehensive review. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):8174–8194, 2023. 1
- [39] T. Sosea and C. Caragea. Marginmatch: Improving semi-supervised learning with pseudo-margins. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15773–15782, 2023. 7, 8
- [40] Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. *arXiv preprint arXiv:2009.10795*, 2020. 2, 6
- [41] Yihe Tang, Weifeng Chen, Yijun Luo, and Yuting Zhang. Humble teachers teach better students for semi-supervised object detection. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3131–3140, 2021. 1
- [42] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017. 1
- [43] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine learning*, 109(2):373–440, 2020. 1
- [44] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 1
- [45] Lei Wang, Xin Yan, Zhu-Hong You, Xi Zhou, Hao-Yuan Li, and Yu-An Huang. Sganrda: semi-supervised generative adversarial networks for predicting circrna–disease associations. *Briefings in Bioinformatics*, 22(5):bbab028, 2021. 1
- [46] Yidong Wang, Hao Chen, Yue Fan, Wang Sun, Ran Tao, Wenxin Hou, Renjie Wang, Linyi Yang, Zhi Zhou, Lan-Zhe Guo, Heli Qi, Zhen Wu, Yu-Feng Li, Satoshi Nakamura, Wei Ye, Marios Savvides, Bhiksha Raj, Takahiro Shinozaki, Bernt Schiele, Jindong Wang, Xing Xie, and Yue Zhang. Usb: A unified semi-supervised learning benchmark for classification. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. 5
- [47] Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, Bernt Schiele, and Xing Xie. Freematch: Self-adaptive thresholding for semi-supervised learning. In *The Eleventh International Conference on Learning Representations*, 2023. 2, 3, 5, 7, 8
- [48] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems*, 33:6256–6268, 2020. 7, 8
- [49] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020. 7
- [50] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020. 1
- [51] Yi Xu, Lei Shang, Jinxing Ye, Qi Qian, Yu-Feng Li, Baigui Sun, Hao Li, and Rong Jin. Dash: Semi-supervised learning with dynamic thresholding. In *International conference on machine learning*, pages 11525–11536. PMLR, 2021. 7, 8
- [52] Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):8934–8954, 2023. 1

- [53] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34:18408–18419, 2021. [2](#), [3](#), [6](#), [7](#), [8](#)
- [54] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018. [4](#)