Semi-Supervised Learning with Multi-Head Co-Training

Mingcai Chen, Yuntao Du, Yi Zhang, Shuwei Qian, Chongjun Wang*

State Key Laboratory for Novel Software Technology at Nanjing University
Nanjing University, Nanjing 210023, China
{chenmc, duyuntao, njuzhangy, QianSW}@smail.nju.edu.cn,
chjwang@nju.edu.cn

Abstract

Co-training, extended from self-training, is one of the frameworks for semi-supervised learning. Without natural split of features, single-view co-training works at the cost of training extra classifiers, where the algorithm should be delicately designed to prevent individual classifiers from collapsing into each other. To remove these obstacles which deter the adoption of single-view co-training, we present a simple and efficient algorithm Multi-Head Co-Training. By integrating base learners into a multi-head structure, the model is in a minimal amount of extra parameters. Every classification head in the unified model interacts with its peers through a "Weak and Strong Augmentation" strategy, in which the diversity is naturally brought by the strong data augmentation. Therefore, the proposed method facilitates single-view co-training by 1). promoting diversity implicitly and 2). only requiring a small extra computational overhead. The effectiveness of Multi-Head Co-Training is demonstrated in an empirical study on standard semi-supervised learning benchmarks.

Introduction

Benefiting from the rich data sources and growing computing power in the last decade, the field of machine learning has been thriving drastically. The advent of public datasets with a large amount of high-quality labels has further spawned many successful deep learning methods (Deng et al. 2009; He et al. 2016; Zagoruyko and Komodakis 2016; Krizhevsky, Sutskever, and Hinton 2012). However, there could be various difficulties for obtaining label information, such as privacy, labor costs, safety or ethic issues, and requirement of domain experts (Zhou 2018; Chapelle, Schlkopf, and Zien 2010; Mahajan et al. 2018). All of these impel us to find a way of bringing unlabeled data into full play. Semi-Supervised Learning (SSL) is a branch of machine learning which seeks to address the problem (Chapelle, Schlkopf, and Zien 2010; Chapelle, Chi, and Zien 2006; Prakash and Nithya 2014; Van Engelen and Hoos 2020). It utilizes both labeled and unlabeled data to improve performance.

As one of the earliest and most popular SSL frameworks, self-training works by iteratively retraining the model using pseudo-labels obtained from itself (Lee 2013; Berthelot et al.

2019b,a; McLachlan 1975). Despite its simplicity and alignment to the task of interest (Zoph et al. 2020), self-training underperforms due to the "confirmation bias" or "error accumulation". It means that some incorrect predictions could be selected as pseudo-labels to guide subsequent training, resulting in a loop of self-reinforcing errors (Zhang et al. 2016).

As an extension of self-training, co-training lets multiple individual models iteratively learn from each other (Zhou and Li 2010; Wang and Zhou 2017). In the early multi-view co-training setting (Blum and Mitchell 1998), there should be a natural split of features where the "sufficiency and independence" assumptions should hold, i.e., it's sufficient to make predictions based on each view, and views are conditionally independent. Later studies gradually reveal that co-training can also be successful in the single-view setting (Wang and Zhou 2017; Dasgupta, Littman, and McAllester 2002; Abney 2002; Balcan, Blum, and Yang 2005; Wang and Zhou 2007). Despite being feasible, the single-view co-training framework has received little attention recently. We attribute it to (a) the extra computational cost, which means at least twice model parameters of its self-training counterpart, and (b) the loss in simplicity, i.e., more design choices and hype-parameters are introduced for keeping individual classifiers uncorrelated.

In this paper, we aim to facilitate the adoption of single-view co-training. Inspired by recent developments of data augmentation and its applications in SSL (Berthelot et al. 2019b; Sohn et al. 2020; Cubuk et al. 2020; DeVries and Taylor 2017), we find that the enormous size of the augmentation search space naturally prevents base learners from converging to a consensus. Employing the stochastic image augmentation frees us from delicately design different network structures or training algorithms. Moreover, by replacing multiple individual models with a shared module followed by multiple classification heads, the model can achieve co-training in a minimal amount of extra parameters. Combining these, we propose Multi-Head Co-Training, a new algorithm that facilitates the usage of single-view co-training. The main contributions are as follows:

- Multi-Head Co-Training addresses two obstacle factors of standard single-view co-training, i.e., extra design and computational cost.

- Experimentally, we show that our method obtains state-of-the-art results on CIFAR, SVHN, and Mini-ImageNet. Be-

---

sides, We systematically study the components of Multi-Head Co-Training.

- We further analyze the calibration of SSL methods and provide insights regarding the link between confirmation bias and model calibration.

Related work

In this section, we concentrate on relevant studies to set the stage for Multi-Head Co-Training. More extensive surveys on SSL can be found in (Prakash and Nithya 2014; Van Engelen and Hoos 2020; Zhu 2005; Zhou and Li 2010; Zhou 2018; Subramanya and Talukdar 2014).

The basic assumptions in SSL are the smoothness assumption and low-density assumption. The smoothness assumption states that if two or more data points are close in the sample space, they should belong to the same class. Similarly, the low-density assumption states that the decision boundary for a classification model shouldn't pass the high-density region of sample space. These assumptions are intuitive in vision tasks because an image with small noise is still semantically identical to the original one. A dominant paradigm in SSL is grounded on these assumptions. From this point of view, various ways of making use of unlabeled data, including consistency regularization, entropy minimization, perturbation-based methods, self-training, and co-training, are essentially similar.

Consistency regularization (Park et al. 2018; Suzuki and Sato 2020) constrains the model to make consistent predictions across the same example under variants of noises. In a general form,

$$D[q(y \mid x), p(y \mid x')] \tag{1}$$

where q and p are the modeled distributions. Different notations are used here, indicating that they could come from different models. The target example is denoted as x and its noisy counterpart is denoted as x'. $D(\cdot, \cdot)$ can be any distance measurement, such as KL divergence or mean square error. SSL methods falling into this category differ in the source of noise, models for two distributions, and the distance measurement. For example, VAT (Miyato et al. 2018) generates noise in an adversarial direction. Laine & Aila (Laine and Aila 2016) propose Π-Model and Temporal Ensembling. Π-Model performs Gaussian noise, dropout, etc., to augment images. Temporal Ensembling further ensembles prior network evaluations to encourage consistent predictions. Mean Teacher (Tarvainen and Valpola 2017) instead maintains an Exponential Moving Average (EMA) of model's parameters. ICT (Verma et al. 2019) applies consistency regularization between the prediction of interpolation of unlabeled points and the interpolation of the predictions at those points. UDA (Xie et al. 2019) replaces the traditional data augmentation with unsupervised data augmentation.

Self-training[1] favors low-density separation by using model's own predictions as pseudo-labels. Pseudo-Labeling

---

[1]In the field of SSL, the terminology "self-training" overlaps with "pseudo-labeling", which refers to training the model incrementally instead of retraining in every iteration. For illustration, we use "self-training" throughout this paper, referring to a broad category of methods.

(Lee 2013) picks the most confident predictions as hard (one-hot) pseudo-labels. Apart from that, the method also uses auto-encoder and dropout as regularization. MixMatch (Berthelot et al. 2019b) uses the average of predictions on the image under multiple augmentations as the soft pseudo-label. Furthermore, Mixup (Zhang et al. 2017), as a regularizer, is used to mix labeled and unlabeled data. ReMixMatch (Berthelot et al. 2019a) improves MixMatch by introducing other regularization techniques and a modified version of AutoAugment (Cubuk et al. 2019). FixMatch (Sohn et al. 2020) finds an effective combination of image augmentation techniques and pseudo-labeling. One of the reasons for the popularity of self-training is its simplicity. It can be used with almost all supervised classifier (Van Engelen and Hoos 2020). Another important but rarely mentioned factor is its awareness of the task of interest. Although some other unsupervised constraints may help build general representations, it has been shown that self-training can align to the task of interest well and benefit model in a more secure way (Zoph et al. 2020). However, confirmation bias, where the prediction mistakes would accumulate during the training process, damages the performance of self-training.

As an extension of self-training, co-training alleviates the problem of confirmation bias. Two or more models are trained by each other's predictions. In the original form (Blum and Mitchell 1998), two individual classifiers are trained on two views. What's more, the proposed co-training algorithm requires the "sufficiency and independence" assumptions hold, i.e., two views should be sufficient to perform accurate prediction and be conditionally independent given the class. Nevertheless, later studies show that weak independence (Abney 2002; Balcan, Blum, and Yang 2005) or single-view data (Wang and Zhou 2017, 2007; Du, Ling, and Zhou 2010) is enough for a successful co-training algorithm.

Without the distinct views of the data containing complementary information, single-view co-training has to promote diversity in some other ways. It also has been shown that the more uncorrelated the individual classifiers are, the more effective the algorithm is. (Wang and Zhou 2017, 2007) Several early studies attempt to split the single-view data (Balcan, Blum, and Yang 2005; Chen, Weinberger, and Chen 2011), and some further approach a pure single-view setting by introducing diversity among the classifiers otherwise (Zhou and Goldman 2004; Goldman and Zhou 2000; Xu, He, and Man 2012). Recently, Deep Co-training (Qiao et al. 2018) maintains disagreement through a view difference constraint. Tri-Net (Dong-DongChen et al. 2018) adopts a multi-head structure. To prevent consensus, it designs different head structures and samples different sub-datasets for learners. CoMatch (Li, Xiong, and Hoi 2020) applies a graph-based smoothness regularization and also integrates supervised contrastive learning. Although have achieved a number of successes, these methods complicate the practical adoption of co-training. In Multi-Head Co-Training, employing the stochastic image augmentation frees us from delicately designing different network structures or training algorithms. Unique to other methods, no preventive measures, e.g., extra loss term or different base learners, are needed to avoid collapse. Along with the reduction of computational cost brought by the multi-
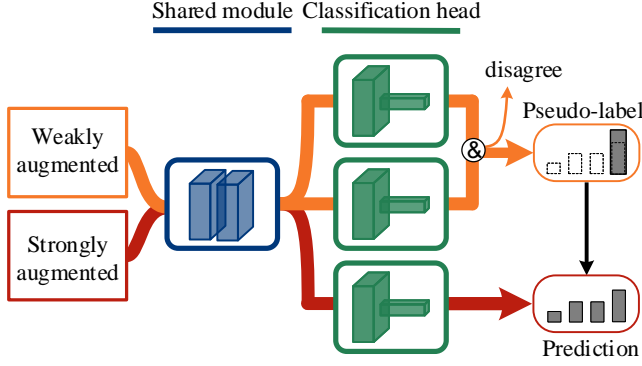
Figure 1: Diagram of Multi-Head Co-Training with three heads. Images are fed into a shared module (blue box) followed by three classification heads (green boxes). Among them, weakly augmented images (orange lines) are for pseudo-labeling. The pseudo-labels are to guide the predictions on strongly augmented examples (red line). Here, pseudo-labels for the bottom head are generated and selected according to the other two heads' predicted classes on the weakly augmentation images. *Note that only the co-training process of the bottom head is shown here.* The weakly and strongly augmented images are in fact simultaneously fed into all three heads.

Algorithm 1: Multi-Head Co-Training with three heads.

Input: Labeled batch $\mathcal{D}_l = \{(x_b, y_b)\}; b \in 1, \ldots, B_l$, unlabeled batch $\mathcal{D}_u = \{(u_b); b \in 1, \ldots, B_u\}$, unsupervised loss weight λ.

Output: Updated model.

1: **for** $b = 1$ to B_l **do**
2:     $\hat{x}_b = Aug_w(x_b)$ ▷ weakly augment
3: **end for**
4: $\mathcal{L}_l = \frac{1}{B_l} \sum_{b=1}^{B_l} \sum_{m \in \{1,2,3\}} \mathrm{H}(y_b, p_m(y \mid \hat{x}_b))$
5: **for** $b = 1$ to B_u **do**
6:     $\hat{u}_b = Aug_w(u_b)$ ▷ weakly augment
7:     **for** $m = 1$ to 3 **do**   ▷ pseudo-labeling
8:        $q_{b,m} = \arg\max_c p_i(y \mid \hat{u}_b)$ $i \neq m$
9:     **end for**
10:     **for** $m = 1$ to 3 **do**   ▷ selected agreed examples
11:        $\mathbb{1}_{b,m} = [q_{b,i} = q_{b,j}]$ $\{i,j\} = \{1,2,3\} \setminus m$
12:     **end for**
13:     **for** $m = 1$ to 3 **do**   ▷ unsupervised loss
14:        $\tilde{u}_{b,m} = Aug_s(u_b)$ ▷ strongly augment
15:        $\ell_{b,m} = \mathrm{H}(q_{b,i}, p_m(y \mid \tilde{u}_{b,m}))$ $i \neq m$
16:     **end for**
17: **end for**
18: $\mathcal{L}_u = \frac{1}{3} \sum_{m \in \{1,2,3\}} \frac{1}{\sum_{b=1}^{B_u} \mathbb{1}_{b,m}} \sum_{b=1}^{B_u} \mathbb{1}_{b,m} \ell_{b,m}$
19: update all parameters according to $\mathcal{L}_l + \lambda \mathcal{L}_u$

---

head structure, the proposed method facilitates the adoption of single-view co-training.

Multi-Head Co-Training

In this section, we introduce the details of Multi-Head Co-Training. Formally, for a C-class classification problem, SSL aims to model a class distribution $p(y \mid x)$ for input x utilizing both labeled and unlabeled data. In Multi-Head Co-Training, the parametric model consists of a shared module f and multiple classification heads g_m ($m \in \{1, \ldots, M\}$) with the same structure. Let $p_m(y \mid x) = g_m(f(x))$ represent the predicted class distribution produced by f and g_m. All classification heads are updated using the using a consensus of predictions from other heads. Apart from that, following the principle of recent successful SSL algorithms, this method utilizes image augmentation techniques to employ a "Weak and Strong Augmentation" strategy. It uses predictions on weakly augmented images, which are relatively more accurate, to correct the predictions on strongly augmented images. The weak and strong augmentation function are denoted as $Aug_w(\cdot)$ and $Aug_s(\cdot)$ respectively and further introduced in The "Weak and Strong Augmentation" strategy. The diagram of Multi-Head Co-Training with three heads is shown in Figure 1. The overall algorithm is shown in Algorithm 1.

The multi-head structure

In every training iteration of Multi-Head Co-Training, a batch of labeled examples $\mathcal{D}_l = \{(x_b, y_b); b \in (1, \ldots, B_l)\}$ and a batch of unlabeled examples $\mathcal{D}_u = \{(u_b); b \in (1, \ldots, B_u)\}$ are randomly sampled from the labeled and unlabeled dataset respectively. For supervised training, the parameters in the shared module f and all heads g_m ($m \in \{1, \ldots, M\}$) are updated to minimize the cross-entropy loss between predictions and true labels, i.e.,

$$\mathcal{L}_l = \frac{1}{B_l} \sum_{b=1}^{B_l} \sum_{m=1}^{M} \mathrm{H}(y_b, p_m(y \mid \hat{x}_b)) \qquad (2)$$

where $\mathrm{H}(\cdot, \cdot)$ represents the cross-entropy loss function and $\hat{x}_b = Aug_w(x_b)$ is the weakly augmented labeled example.

For co-training, every head interacts with its peers through pseudo-labels on unlabeled data. To obtain reliable predictions for pseudo-labeling, weakly augmented unlabeled examples $\hat{u}_b = Aug_w(u_b)$ first pass through the shared module and all heads simultaneously,

$$q_{b,m} = \arg\max_c p_m(y = c \mid \hat{u}_b) \qquad (3)$$

where $\arg\max$ picks the class $c \in \{1, \ldots, C\}$ with the maximal probability, i.e., the most confident predicted class $q_{b,m}$. To avoid confirmation bias, pseudo-labels for each head depend on the predicted classes from other $M - 1$ heads,

$$\mathbb{Q}_{b,m} = \{q_{b,1}, \ldots, q_{b,M}\} \setminus q_{b,m} \qquad (4)$$

where $\setminus$ is the set operation of removing element. The most frequently predicted class in multiset $\mathbb{Q}_{b,m}$ is the pseudo-label $\bar{q}_{b,m}$ for m-th head, and it is selected only when more than half heads agree on the prediction,

$$\bar{q}_{b,m} = \arg\max_c \sum_{q \in \mathbb{Q}_{b,m}} [q = c]$$

$$\mathbb{1}_{b,m} = [\sum_{q \in \mathbb{Q}_{b,m}} [q = \bar{q}_{b,m}] > \frac{M}{2}] \qquad (5)$$

where $[\cdot]$ refers to the Iverson brackets, defined to be 1 if the statement in it is true, otherwise 0. $\mathbb{1}_{b,m}$ indicates whether b-th pseudo-label is selected for m-th head. After the selection process, uncertain (less agreed) examples are filtered. In the meantime, strongly augmented unlabeled examples go through the shared module f and corresponding head g_m. The average cross-entropy is then calculated:

$$\mathcal{L}_u = \frac{1}{M} \sum_{m=1}^{M} \frac{1}{\sum_{b=1}^{B_u} \mathbb{1}_{b,m}} \sum_{b=1}^{B_u} \mathbb{1}_{b,m} \ell_{b,m} \qquad (6)$$
$$\ell_{b,m} = \mathrm{H}(\bar{q}_{b,m}, p_m(y \mid \tilde{u}_{b,m}))$$

where $\{(\tilde{u}_{b,m}); m \in (1, \ldots, M)\}$ comes from M times of strong augmentation $Aug_s(\cdot)$. Note that the transformation function generates a differently augmented image every time. The supervised loss and unsupervised loss are added together as the total loss $\mathcal{L}$ (weighted by coefficient λ), i.e.,

$$\mathcal{L} = \mathcal{L}_l + \lambda \mathcal{L}_u \qquad (7)$$

The algorithm proceeds until reaching fixed iterations (training details are illustrated in the supplementary material).

As discussed in Related work, the quality of pseudo-labels and the diversity between individual classifiers are the two important things for a co-training algorithm to succeed. In our method, the diversity between heads inherently comes from the randomness in the strong augmentation function (consequently, the unlabeled examples for each head are differently augmented, selected, and pseudo-labeled). Unique to co-training algorithms, diversity is promoted implicitly in Multi-Head Co-Training. In terms of the quality of pseudo-labels, ensemble predictions of other heads on weakly augmented examples are used for accurately pseudo-labeling and selecting.

The "Weak and Strong Augmentation" strategy

Due to the scarcity of labels, preventing overfitting, or in other words, improving the generalization ability is the core task of SSL. Data augmentation approaches such a problem via expanding the size of training set, and thus plays a vital role in SSL. In Multi-Head Co-Training, the weak and strong augmentation functions differ in the degree of image augmentation. Specifically, the weak image transformation function $Aug_w(\cdot)$ applies random horizontal flip and random crop. Two augmentation techniques, namely RandAugment (Cubuk et al. 2020) and Cutout (DeVries and Taylor 2017), constitute the strong transformation function $Aug_s(\cdot)$. In RandAugment, a given number of operations are randomly selected from a fixed set of geometric and photometric transformations, such as affine transformation, color adjustment. Then they are applied to images with a given magnitude. Cutout randomly masks out square regions of images. Both of them are applied sequentially in the strong augmentation. It has been shown that unsupervised learning benefits from stronger data augmentation (Chen et al. 2020). The same preference can also be extended to SSL. Thus, the setting of RandAugment follows the modified stronger version in FixMatch (Sohn et al. 2020) and details are reported in the supplementary material.

Exponential moving average

To enforce smoothness, Exponential Moving Average (EMA) is a widely used technique in SSL. In this paper, we maintain an EMA model for evaluation. Its parameters are updated every iteration using the training-time model's parameters:

$$\bar{\theta} \leftarrow \alpha \bar{\theta} + (1 - \alpha)\theta \qquad (8)$$

where $\bar{\theta}$ is the parameters of the EMA model. θ is the parameters of training-time model. α is the decay rate which controls how much the average model moves every time. During test-time, we simply ensemble all heads' predictions of the EMA model by adding them together.

Experiments

The number and structure of heads in our framework can be arbitrary, but we set the head number as three and the head structure as the last residual block (Zagoruyko and Komodakis 2016) in most of the experiments. The choice is the result of a trade-off between accuracy and efficiency (illustrated in Ablation study).

Results

We benchmark the proposed method on experimental settings using CIFAR-10 (Krizhevsky, Hinton et al. 2009), CIFAR-100 (Krizhevsky, Hinton et al. 2009), and SVHN (Netzer et al. 2011) as the standard practice. Different portions of labeled data ranging from 0.5% to 20% are experimented. For comparison, we consider Tri-net (Dong-DongChen et al. 2018), Π-Model (Laine and Aila 2016), Pseudo-Label (Lee 2013), Mean Teacher (Tarvainen and Valpola 2017), VAT (Miyato et al. 2018), MixMatch (Berthelot et al. 2019b), ReMixMatch (Berthelot et al. 2019a), FixMatch (Sohn et al. 2020). The results of these methods reported in this section are reproduced by (Berthelot et al. 2019b,a) using the same backbone and training protocol (except for the results of Tri-net and Fix-Match are from their papers). The main criterion is the error rate. Variance is also reported to ensure the results are statistically significant (5 runs with different seeds). We report the final performance of the EMA model. SGD with Nesterov momentum (Sutskever et al. 2013) is used, along with weight decay and cosine learning rate decay (Loshchilov and Hutter 2016). The details are in the supplementary material.

CIFAR-10, CIFAR-100 We first compare Multi-Head Co-Training to the state-of-the-art methods on CIFAR (Krizhevsky, Hinton et al. 2009), which is one of the most commonly used image recognition datasets. We randomly choose 250-4000 from 50000 training images of CIFAR-10 as labeled examples. Other images' labels are thrown away. The backbone model is WRN 28-2 (extra heads are added). As shown in Table 1, Multi-Head Co-Training performs consistently better against the state-of-the-art methods. For example, it achieves an average error rate of 3.84% on CIFAR-10 with 4000 labeled images, compared favorably to the state-of-the-art results.

We randomly choose 10000 from 50000 training images of CIFAR-100 as labeled examples and throw other's label information. In Table 1, we present the results of CIFAR-100

	SVHN			CIFAR-10			CIFAR-100
Method	250 labels	500 labels	1000 labels	250 labels	1000 labels	4000 labels	10000 labels
Tri-net	-	-	3.71±0.14	-	-	8.45±0.22	-
Π-Model	17.65±0.27	11.44±0.39	8.60±0.18	53.02±2.05	31.53±0.98	17.41±0.37	37.88±0.11
Pseudo-Label	21.16±0.88	14.35±0.37	10.19±0.41	49.98±1.17	30.91±1.73	16.21±0.11	36.21±0.19
VAT	8.41±1.01	7.44±0.79	5.98±0.21	36.03±2.82	18.68±0.40	11.05±0.31	-
Mean Teacher	6.45±2.43	3.82±0.17	3.75±0.10	47.32±4.71	17.32±4.00	10.36±0.25	35.83±0.24
MixMatch	3.78±0.26	3.27±0.31	3.27±0.31	11.08±0.87	7.75±0.32	6.24±0.06	28.31±0.33
ReMixMatch	3.10±0.50	-	2.83±0.30	6.27±0.34	5.73±0.16	5.14±0.04	23.03±0.56
FixMatch (RA)	2.48±0.38	-	2.28±0.11	5.07±0.65	-	4.26±0.05	22.60±0.12
FixMatch (CTA)	2.64±0.64	-	2.36±0.19	5.07±0.33	-	4.31±0.15	23.18±0.11
Ours	**2.21±0.18**	**2.18±0.08**	**2.16±0.05**	**4.98±0.30**	**4.74±0.16**	**3.84±0.09**	**21.68±0.16**

Table 1: Error rates for SVHN, CIFAR-10, and CIFAR-100. The best results are in bold.

	4000 labels	10000 labels
Mean Teacher	72.51±0.22	57.55±1.11
Pseudo-Label	56.49±0.51	46.08±0.11
LaplaceNet	**46.32±0.27**	**39.43±0.09**
Ours	46.53±0.15	39.74±0.12

Table 2: Error rates for Mini-ImageNet.

	WRN 28-2	WRN 28-8
Original	1.4 M (30 min)	23.4 M (136 min)
Three model	4.2 M (66 min)	70.2 M (344 min)
Ours	3.7 M (39 min)	19.9 M (168 min)

Table 3: Number of parameters (Million) and average training time for 10000 iterations (minutes).

with 10000 labels. As the common practice in recent methods, WRN 28-8 is used to accommodate the more challenging task (more classes and each with fewer examples). We reduce the number of channels of the final block in WRN 28-8 from 512 to 256. By doing so, the model has a much smaller size. Combining the results in Table 1, Multi-Head Co-Training achieves 21.68% error rate, having an improvement of 1.5% compared to the best results of previous methods with even a smaller model.

SVHN Similarly, we evaluate the accuracy of our method with a varying number of labels from 250 to 1000 on the SVHN dataset (Netzer et al. 2011) (the extra training set is not used). The image augmentation for SVHN is different because some operations are not suitable for digit images (e.g., horizontal flip for asymmetrical digit images). Its details are in the supplementary material. The results of Multi-Head Co-Training and other methods are shown in Table 1. Multi-Head Co-Training outperforms other methods by a small margin.

Mini-ImageNet We further evaluate our model on the more complex dataset Mini-ImageNet (Vinyals et al. 2016), which is a subset ImageNet (Deng et al. 2009). The training set of Mini-ImageNet consists of 50000 images with a size of 84 × 84 in 10 object classes. We randomly choose 4000 and 10000 images as labeled examples and throw other's label information. The backbone model is ResNet-18 (Wang et al. 2017) and early stopped using the ImageNet validation set. Other methods' results are from (Sellars, Aviles-Rivero, and Schönlieb 2021). Our method achieves an error rate of 47.88% and 39.74% for 4k and 10k labeled images, respectively. The results are competitive to a recent method

LaplaceNet (Sellars, Aviles-Rivero, and Schönlieb 2021), which uses graph-based constrain and multiple strong augmentation. Besides, our co-training method, which is simple and efficient, is orthogonal to other SSL constraints.

Computational cost analysis We report the training cost of the original backbone, standard co-training with three models, and our method in Table 3. The reduction of the number of parameters and training time is significant. For example, the number of parameters in the WRN 28-8 backbone is instead fewer than the original one. Only 23.5% extra time, compared to self-training, is needed to train our method, while the standard co-training needs 152.9% extra time.

Ablation study

This section presents an ablation study to measure the contribution of different components of Multi-Head Co-Training. Specifically, we measure the effect of

1). Multi-Head Co-Training with only one head. Pseudo-labels are generated from their own predictions and selected by a confidence threshold of 0.95.

2). Multi-Head Co-Training with one strong augmentation. Strong augmentation is only performed once and forwarded to all three heads.

3). Multi-Head Co-Training without weak augmentation. The original images are for pseudo-labeling.

4). Multi-Head Co-Training with three heads with the same initialization.

5). Multi-Head Co-Training without EMA.

We first set Multi-Head Co-Training's self-training counterpart as described at 1) as a baseline. It has the same back-

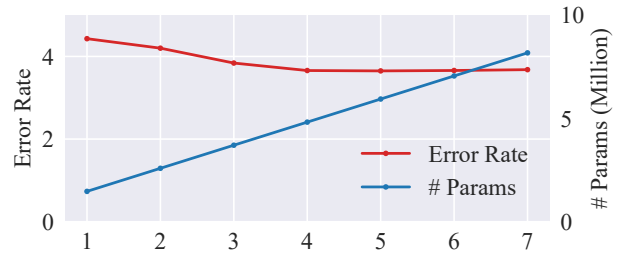| Ablation	One head	Ensemble
Multi-Head Co-Training	4.22	3.84
1). One head	4.43	4.23
2). One strong augmentation	4.45	4.03
3). W/o weak augmentation	4.86	4.55
4). Same heads' initialization	4.28	3.86
5). W/o EMA	6.23	5.30

Table 4: Ablation experiments. The models are trained on CIFAR-10 with 4000 labels. The average error rate of individual heads and their ensemble both reported.

bone and hype-parameters but with only one head. The self-training baseline obtains sub-optimal performance. To further verify the main improvement of our method is not coming from ensembling, the self-training model is retrained three times with different initialization to produce an ensemble result ("Ensemble" in 1) row). It can be observed from Table 4 that Multi-Head Co-Training, as a co-training algorithm, first shows its effectiveness by outperforming it.
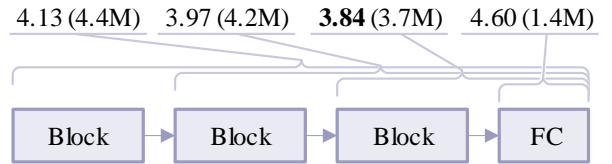
Promoting diversity between individual models is critical to the success of the co-training framework. Otherwise, they would produce too similar predictions, and thus co-training degenerates into self-training. Other single-view co-training methods create diversity mainly in several ways, including automatic view splitting (Balcan, Blum, and Yang 2005; Chen, Weinberger, and Chen 2011), using different individual classifiers or individual classifiers with different structures (Zhou and Goldman 2004; Goldman and Zhou 2000; Xu, He, and Man 2012; Dong-DongChen et al. 2018). Unique to a co-training algorithm, Multi-Head Co-Training doesn't promote diversity explicitly. The diversity between heads inherently comes from the randomness in parameter initialization and augmentation (consequently, examples selected for each head are different). To study the impact of them, we remove each source of diversity at 2) and 4) respectively. As shown in Table 4, the accuracy drops more when differently augmented images are missing. Moreover, the individual heads' error rate is almost the same as the self-training baseline 1). This shows the important role strong augmentation plays in Multi-Head Co-Training. As a regularizer, data augmentation is considered to confine learning to a subset of the hypothesis space (Zhang et al. 2016). We believe multiple strong augmentations confine classification heads of Multi-Head Co-Training in different subsets of the hypothesis space and thus, keeping them uncorrelated.

According to the observation in 3), replacing the weakly augmented images with the original ones leads to a worse final performance. Note that pseudo-labels obtained from original images are more accurate. It means that current pseudo-labels from weakly augmented images, even with lower accuracy, would lead to a better model in later training. An interesting fact implied by the phenomenon is that accuracy isn't the only important factor of pseudo-labels.

The number and structure of heads One main difference between Multi-Head Co-Training and other SSL methods is the multi-head structure. It brings many benefits. Firstly,

(a) Different number of heads.

4.13 (4.4M) 3.97 (4.2M) **3.84** (3.7M) 4.60 (1.4M)

(b) Different structure of heads.

Figure 2: The error rate and the number of parameters brought by different heads.

it naturally produces multiple uncorrelated predictions for each example that regularizes the feature representation the shared module learns. Secondly, pseudo-labels coming from the ensemble predictions on weakly augmented examples are more reliable. Thirdly, the number of parameters is much smaller because base learners share a module. Based on WRN (Zagoruyko and Komodakis 2016), we empirically find a both accurate and efficient structure. Specifically, we experiment Multi-Head Co-Training with different number of heads and different head structures in this section. Considering that it's impractical to search all combinations, WRN 28-2 backbone on CIFAR-10 with 4000 labels is studied. The head structure is fixed when we attempt to find the optimal number of heads. Similarly, the number of heads is fixed when we attempt to find the optimal head structure.

We first test Multi-Head Co-Training with 1-7 heads while fixing the structure of head as the last block in WRN 28-2. For consistency, when the number of heads is 1 or 2, i.e., the pseudo-labels come from only one head's predictions, a threshold of 0.95 is used for selecting. As shown in Figure 2a, with more heads, better performance can be obtained, but the accuracy growth slows down. The structures with more heads are not considered because the gain becomes insignificant with the increasing of the number of heads.

In terms of the head structure, the most important thing is finding the balance point between the size of the shared module and the size of the head. As shown in Figure 2b, the best performance is observed when the heads have the structure of one block and one fully connected layer and the shared module has the structure of one convolutional layer and two blocks. Whether increase or decrease the size of the heads would damage performance. Our explanation is that if there are too many shared parameters, there could be little room for the heads to make diverse predictions. If there are too many independent parameters, the heads would easily fit the pseudo-labels, and again, fail to make diverse predictions.
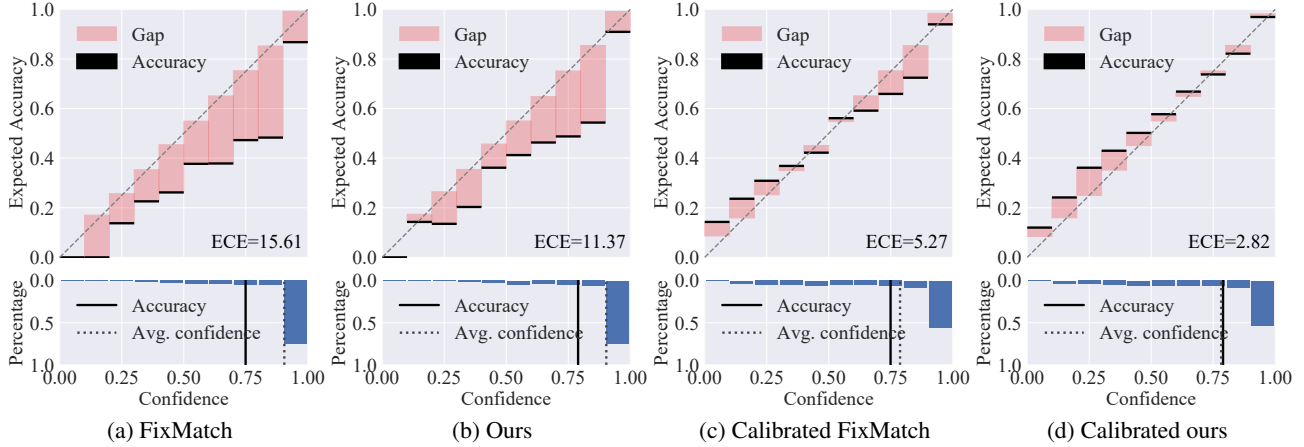
Figure 3: Reliability diagrams (top) and confidence histograms (bottom).[2] The models are trained on CIFAR-100 with 10000 labels, and their predictions on the test set are grouped into 10 interval bins (horizontal axis). Reliability diagram presents the true accuracy, the expected accuracy, and the gap between them of each bin. confidence histogram presents the percentage of examples that falls into each bin. The accuracy and average confidence are indicated by the solid and dashed lines, respectively.

Considering our main purpose of developing an effective co-training algorithm, we set the number of heads as three and the head as one block in our experiments.

Calibration of SSL

Confirmation bias comes up frequently in SSL literature. However, it is hard to formulate or observe the problem. We notice that most self-training or co-training methods select pseudo-labels by some criteria, such as confidence threshold. In these cases, confirmation bias is closely related to the over-confidence of the network: Wrong predictions with high confidence are likely to be selected and then used as pseudo-labels. Thus, we link confirmation bias to model calibration, i.e., the problem of predicted probability represents the true correctness likelihood. We envision the calibration measurements be used to evaluate confirmation bias and help the design of self-training and co-training algorithms. Apart from this, the challenges of SSL and calibration could appear simultaneously in real-world. For example, in one of the applications of SSL, medical diagnosis, control should be passed on to human experts when the confidence of automatic diagnosis is low. In such scenarios, a well-calibrated SSL model is needed. To the best of our knowledge, these two problems have hitherto been studied independently.

According to our observation, SSL models have poor performance in terms of calibration due to entropy minimization and other similar constraints. We analyze FixMatch (implemented using the same code-base), as one of the typical SSL methods, and Multi-Head Co-Training on CIFAR-100 with 10000 labels from the perspective of model calibration. Several common calibration indicators are used, namely Expected Calibration Error (ECE), confidence histogram, and reliability diagram (illustrated in the supplementary mate-

rial). As shown in Figure 3a, FixMatch has an average confidence of 78.75% but only 74.93% accuracy, producing over-confident results with an ECE value of 15.61. In Figure 3b, we show the results of Multi-Head Co-Training. Although our method produces over-confident predictions, the ECE value is smaller, indicating better probability estimation.

To further investigate, we apply a simple calibration technique called "temperature scaling" (Guo et al. 2017) (see the supplementary material). From the calibrated results in Figure 3c and Figure 3d, it can be observed that the miscalibration is remedied. The ECE value of calibrated FixMatch is improved to 5.27% Multi-Head Co-Training's reliability diagrams closely recovers the desired diagonal function with a low ECE value of 2.82. It can be concluded that Multi-Head Co-Training produces good probability estimates naturally and can be better calibrated with simple techniques. Therefore, we suggest that confirmation bias is better addressed in our method from the perspective of calibration.

Conclusion

The field of SSL encompasses a broad spectrum of algorithms. However, co-training framework has received little attention recently because of the diversity criterion, extra computational cost. Multi-Head Co-Training pointedly addresses these. It achieves single-view co-training by integrating the individual models into one multi-head structure and utilizing the data augmentation techniques. As a result, the proposed method is 1). in a minimal amount of both parameters and hype-parameters and 2). doesn't need extra effort to promote diversity. We present systematic experiments to show that Multi-Head Co-Training is a successful co-training method and outperforms state-of-the-art methods. The solid empirical results suggest that it is possible to scale co-training to more realistic SSL settings. In future work, we are interested in combining modality-agnostic data augmentation to make Multi-Head Co-Training ready to be applied to other tasks.

---

[2]Draw with the code in https://github.com/hollance/reliability-diagrams.

Acknowledgements

References

Abney, S. 2002. Bootstrapping. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 360–367.

Balcan, M.-F.; Blum, A.; and Yang, K. 2005. Co-training and expansion: Towards bridging theory and practice. *Advances in neural information processing systems*, 17: 89–96.

Berthelot, D.; Carlini, N.; Cubuk, E. D.; Kurakin, A.; Sohn, K.; Zhang, H.; and Raffel, C. 2019a. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*.

Berthelot, D.; Carlini, N.; Goodfellow, I.; Papernot, N.; Oliver, A.; and Raffel, C. 2019b. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*.

Blum, A.; and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, 92–100.

Chapelle, O.; Chi, M.; and Zien, A. 2006. A continuation method for semi-supervised SVMs. In *Proceedings of the 23rd international conference on Machine learning*, 185–192.

Chapelle, O.; Schlkopf, B.; and Zien, A. 2010. *Semi-Supervised Learning*. The MIT Press, 1st edition. ISBN 0262514125.

Chen, M.; Weinberger, K. Q.; and Chen, Y. 2011. Automatic feature decomposition for single view co-training. In *ICML*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.

Cubuk, E. D.; Zoph, B.; Mane, D.; Vasudevan, V.; and Le, Q. V. 2019. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 113–123.

Cubuk, E. D.; Zoph, B.; Shlens, J.; and Le, Q. V. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 702–703.

Dasgupta, S.; Littman, M. L.; and McAllester, D. 2002. PAC generalization bounds for co-training. *Advances in neural information processing systems*, 1: 375–382.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

DeVries, T.; and Taylor, G. W. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

Dong-DongChen; WeiWang; WeiGao; and Zhou, Z.-H. 2018. Tri-net for semi-supervised deep learning. In *International Joint Conferences on Artificial Intelligence*.

Du, J.; Ling, C. X.; and Zhou, Z.-H. 2010. When does co-training work in real data? *IEEE Transactions on Knowledge and Data Engineering*, 23(5): 788–799.

Goldman, S.; and Zhou, Y. 2000. Enhancing supervised learning with unlabeled data. In *ICML*, 327–334. Citeseer.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*, 1321–1330. PMLR.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25: 1097–1105.

Laine, S.; and Aila, T. 2016. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*.

Lee, D.-H. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3.

Li, J.; Xiong, C.; and Hoi, S. 2020. CoMatch: Semi-supervised Learning with Contrastive Graph Regularization. *arXiv preprint arXiv:2011.11183*.

Loshchilov, I.; and Hutter, F. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Mahajan, D.; Girshick, R.; Ramanathan, V.; He, K.; Paluri, M.; Li, Y.; Bharambe, A.; and Van Der Maaten, L. 2018. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 181–196.

McLachlan, G. J. 1975. Iterative reclassification procedure for constructing an asymptotically optimal rule of allocation in discriminant analysis. *Journal of the American Statistical Association*, 70(350): 365–369.

Miyato, T.; Maeda, S.-i.; Koyama, M.; and Ishii, S. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8): 1979–1993.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning.

Park, S.; Park, J.; Shin, S.-J.; and Moon, I.-C. 2018. Adversarial dropout for supervised and semi-supervised learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Prakash, V. J.; and Nithya, D. L. 2014. A survey on semi-supervised learning techniques. *arXiv preprint arXiv:1402.4645*.

Qiao, S.; Shen, W.; Zhang, Z.; Wang, B.; and Yuille, A. 2018. Deep co-training for semi-supervised image recognition. In *Proceedings of the european conference on computer vision (eccv)*, 135–152.

Sellars, P.; Aviles-Rivero, A. I.; and Schönlieb, C.-B. 2021. LaplaceNet: A Hybrid Energy-Neural Model for Deep Semi-Supervised Classification. *arXiv preprint arXiv:2106.04527*.

Sohn, K.; Berthelot, D.; Li, C.-L.; Zhang, Z.; Carlini, N.; Cubuk, E. D.; Kurakin, A.; Zhang, H.; and Raffel, C. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*.

Subramanya, A.; and Talukdar, P. P. 2014. Graph-based semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 8(4): 1–125.

Sutskever, I.; Martens, J.; Dahl, G.; and Hinton, G. 2013. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, 1139–1147. PMLR.

Suzuki, T.; and Sato, I. 2020. Adversarial Transformations for Semi-Supervised Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 5916–5923.

Tarvainen, A.; and Valpola, H. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *arXiv preprint arXiv:1703.01780*.

Van Engelen, J. E.; and Hoos, H. H. 2020. A survey on semi-supervised learning. *Machine Learning*, 109(2): 373–440.

Verma, V.; Kawaguchi, K.; Lamb, A.; Kannala, J.; Bengio, Y.; and Lopez-Paz, D. 2019. Interpolation consistency training for semi-supervised learning. *arXiv preprint arXiv:1903.03825*.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. *Advances in neural information processing systems*, 29: 3630–3638.

Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; and Tang, X. 2017. Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3156–3164.

Wang, W.; and Zhou, Z.-H. 2007. Analyzing co-training style algorithms. In *European conference on machine learning*, 454–465. Springer.

Wang, W.; and Zhou, Z.-H. 2017. Theoretical foundation of co-training and disagreement-based algorithms. *arXiv preprint arXiv:1708.04403*.

Xie, Q.; Dai, Z.; Hovy, E.; Luong, M.-T.; and Le, Q. V. 2019. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*.

Xu, J.; He, H.; and Man, H. 2012. DCPE co-training for classification. *Neurocomputing*, 86: 75–85.

Zagoruyko, S.; and Komodakis, N. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2016. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhou, Y.; and Goldman, S. 2004. Democratic co-learning. In *16th IEEE International Conference on Tools with Artificial Intelligence*, 594–602. IEEE.

Zhou, Z.-H. 2018. A brief introduction to weakly supervised learning. *National science review*, 5(1): 44–53.

Zhou, Z.-H.; and Li, M. 2010. Semi-supervised learning by disagreement. *Knowledge and Information Systems*, 24(3): 415–439.

Zhu, X. J. 2005. Semi-supervised learning literature survey.

Zoph, B.; Ghiasi, G.; Lin, T.-Y.; Cui, Y.; Liu, H.; Cubuk, E. D.; and Le, Q. V. 2020. Rethinking pre-training and self-training. *arXiv preprint arXiv:2006.06882*.