https://doi.org/10.26599/AIR.2022.9150011 CAAI Artificial Intelligence Research

Meta-Semi: A Meta-Learning Approach for Semi-Supervised Learning

Yulin Wang[1], Jiayi Guo[1], Jiangshan Wang[2], Cheng Wu[1], Shiji Song[1], and Gao Huang[1] ✉

ABSTRACT

Deep learning based semi-supervised learning (SSL) algorithms have led to promising results in recent years. However, they tend to introduce multiple tunable hyper-parameters, making them less practical in real SSL scenarios where the labeled data is scarce for extensive hyper-parameter search. In this paper, we propose a novel meta-learning based SSL algorithm (Meta-Semi) that requires tuning only one additional hyper-parameter, compared with a standard supervised deep learning algorithm, to achieve competitive performance under various conditions of SSL. We start by defining a meta optimization problem that minimizes the loss on labeled data through dynamically reweighting the loss on unlabeled samples, which are associated with soft pseudo labels during training. As the meta problem is computationally intensive to solve directly, we propose an efficient algorithm to dynamically obtain the approximate solutions. We show theoretically that Meta-Semi converges to the stationary point of the loss function on labeled data under mild conditions. Empirically, Meta-Semi outperforms state-of-the-art SSL algorithms significantly on the challenging semi-supervised CIFAR-100 and STL-10 tasks, and achieves competitive performance on CIFAR-10 and SVHN.

KEYWORDS

deep learning; semi-supervised learning; computer vision

Recent success of deep learning in supervised tasks is fueled by abundant annotated training data[1–8]. However, collecting precise labels in practice is usually very time-consuming and costly. In many real-world applications, only a small subset of all available training data are associated with labels[9, 10]. Semi-supervised learning (SSL) is a learning paradigm that aims to improve the model performance by simultaneously leveraging labeled and unlabeled data[11–13].

In the context of deep learning, many successful SSL methods incorporate unlabeled data by performing unsupervised consistency regularization[10, 14–17]. In specific, they first add small perturbations to the unlabeled samples, and then enforce the consistency between the model predictions on the original data and the perturbed data. Though impressive performance has been achieved, the state-of-the-art consistency based algorithms tend to introduce multiple tunable hyper-parameters. The final performance of the algorithms is usually conditioned on setting proper values for these hyper-parameters. However, in many real SSL scenarios like medical image processing[18, 19], hyper-spectral image classification[20, 21], network traffic recognition[22], and document recognition[23], hyper-parameter searching is usually unreliable as the annotated data are scarce, leading to high variance when cross-validation is adopted[9]. This problem will become even more serious if the performance of the algorithm is sensitive to the hyper-parameter values. Furthermore, since the searching space grows exponentially with respect to the number of hyper-parameters[24], the computational cost may become unaffordable for modern deep learning algorithms.

Another challenge to develop practical and robust deep SSL algorithms is how to exploit the labeled data more efficiently, as these data, although being scarce, have the precise and reliable annotations. Consistency based SSL algorithms[10, 14–17] usually model the labeled and unlabeled data in separate terms in the loss function, where the unlabeled data receives no supervision, at least explicitly, from the former, leading to an inefficient use of the labeled data.

In this paper, we propose a meta-learning-based SSL algorithm, named Meta-Semi, to efficiently exploit the labeled data, while it requires tuning only one additional hyper-parameter to achieve impressive performance under various conditions. The proposed algorithm is derived from a simple motivation: the network can be trained effectively with the correctly "pseudo-labeled" unannotated samples. To be specific, we first generate soft pseudo labels for the unlabeled data online during the training process based on the network predictions. Then we filter out the samples whose pseudo labels are incorrect or unreliable, and train the model using the remaining data with relatively reliable pseudo labels. We demonstrate that our idea naturally yields a meta-learning formulation, i.e., the correctly "pseudo-labeled" data should have a similar distribution to the labeled data, and hence if the network is trained with the former, the final loss on the latter should be minimized as well.

Driven by the discussions above, we define a meta reweighting objective: finding the optimal weights (through out the paper, the term "weights" always refer to the coefficients that we use to reweight each individual unlabeled sample, instead of referring to the parameters of neural net works) for different pseudo-labeled samples to train a network, such that the final loss on the labeled data is minimized. We find that this problem is computationally intensive to be directly solved via optimization algorithms.

1 Department of Automation, BNRist, Tsinghua University, Beijing 100084, China
2 Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen 518131, China
Yulin Wang and Jiayi Guo contributed equally to this work.
Address correspondence to Gao Huang, gaohuang@tsinghua.edu.cn

Therefore, we propose an approximated formulation, based on which a closed form solution can be obtained. We show theoretically that one meta gradient step is sufficient to obtain the approximate solutions at each training iteration. Finally, we propose a dynamic weighting algorithm to reweight pseudo-labeled samples with 0–1 weights. Theoretical analysis shows that our method converges to the stationary point of the supervised loss function.

Our algorithm is empirically validated on widely used image classification benchmarks (CIFAR-10, CIFAR-100, SVHN, and STL-10) with modern deep networks (e.g., CNN-13 and WRN-28-2). Meta-Semi outperforms state-of-the-art SSL algorithms, including ICT[10] and MixMatch[17], on the challenging CIFAR-100 and STL-10 SSL tasks significantly, while achieves slightly better performance than them on CIFAR-10. Besides, Meta-Semi is complementary to consistency based methods, i.e., performing consistency regularization in our algorithm further improves the performance.

1 Related Work

Consistency based SSL has been extensively studied in the context of deep learning in recent years[10,14–16,25]. These methods leverage unlabeled data by adding an unsupervised regularization term to the standard supervised loss: $\mathcal{L}_{\mathrm{S}} + w\mathcal{L}_{\mathrm{US}}$, where $\mathcal{L}_{\mathrm{S}}$ is the conventional loss on labeled data, $\mathcal{L}_{\mathrm{US}}$ is the loss contributed by unlabeled data which is usually defined as a measure of discrepancy between the model predictions on the original unlabeled samples and their perturbed counterparts, and w is a pre-defined coefficient. Existing approaches have proposed different ways to generate the perturbations for $\mathcal{L}_{\mathrm{US}}$, including data augmentation[14,25,26], adversarial noise[16], Dropout[27], data interpolation[10], etc. To enhance the model stability, an exponential moving average (EMA) on parameters or predictions is often adopted[14,15]. The effectiveness of these approaches is conditioned on the proper setting of the coefficient w. As the recent methods[17,28] usually integrate multiple regularization techniques, finding the proper hyperparameter setting becomes a challenging problem in practice, especially in the SSL scenarios where few samples are available for performing cross-validation.

Other SSL algorithms. Earlier work on SSL can be categorized into cluster assumption based methods[29,30] and graph assumption based methods[11,31]. For deep learning based SSL, Refs. [32, 33] propose to train deep generators using both the labeled and unlabeled data to estimate the data distribution. Pseudo label based method[34] is also widely used in deep SSL. It progressively uses the highly confident model predictions to generate pseudo labels for unlabeled samples during training. Minimizing the entropy of the model prediction on unlabeled data is also proven effective for SSL[16,35]. SSL can also be applied to clustering[36], active learning[37], extreme learning machines[38], and bayesian regression[39].

Meta learning. Since Meta-Semi follows a meta-learning paradigm, we briefly review the existing work on this topic. The idea of meta-learning is motivated by the goal of "learning to learn better"[40,41]. Meta-learning algorithms usually define a meta optimization problem to extract information from the learning process. For example, using the loss on a small amount of trustable data as the meta-objective is widely adopted in few-shot learning[42,43]. MAML[44] proposes to minimize the meta loss directly via gradient descents. To address the challenge that naively minimizing the meta objective requires performing multiple meta update steps iteratively for every "real" update step on model parameters, Ref. [45] proposed an online approximation method to make the meta training process more tractable. The proposed algorithm is similar to that in Ref. [45], but our contributions lie in several important aspects. First, we propose to exploit the labeled data more efficiently in SSL by leveraging the meta-reweighting method, which not only reduces the required number of tunable hyper-parameters, but also effectively improves the performance. As far as we know, this idea has not been explored in the literature. Second, we propose a novel dynamical re-weighting process that is tailored for SSL. This is non-trivial since directly applying the method in Ref. [45] to SSL leads to inferior results (see Table 1). Third, we provide a theoretical convergence analysis in the context of SSL, which utilizes different proof techniques from Ref. [45].

Semi-supervised few-shot learning. Similar to Meta-Semi, few-shot learning (FSL)[44,46] also seeks to solve the problem of scarce labeled data, and some of existing works combine SSL and FSL by leveraging both labeled and unlabeled training data[43,47–50]. However, FSL considers the cases where the training data is extremely insufficient (e.g., 1 or 5 labeled samples and 30 or 50 unlabeled samples per class), and many of effective approaches focus on transferring the experience learned from similar tasks to the target task (e.g., fine-tuning from a pre-trained feature extractor[48,50–52]). In contrast, we assume that the amount of training data is relatively sufficient, while only the portion of labeled data is small, and the backbone network needs to be trained from scratch. As a matter of fact, our method is compatible with the semi-supervised FSL algorithms, which can be deployed on top of the backbones pre-trained using Meta-Semi for higher performance.

In particular, Ren et al.[43] and LST[48] solve the problem of semi-supervised FSL under a meta-learning paradigm. However, Meta-Semi differentiates itself from them in several important aspects. First, as aforementioned, the SSL problem we consider is different from them, while our method is compatible with the semi-supervised FSL algorithms like them. Second, they define meta-learning problems in the dimension of task, i.e., a model for a learning algorithm is defined and trained on the episodes representing different tasks, each with a small training set and its corresponding test set. In contrast, our meta-learning approach is based on the data dimension, i.e., only one task is considered, where we reweight different pseudo-labeled samples. Third, we propose an efficient algorithm with theoretical guarantees for solving the meta-learning problem, which is tailored for our formulation.

2 Method

In this section, we introduce the details of our Meta-Semi algorithm. Different from most existing methods that leverage unsupervised consistency regularization, we propose to solve the SSL problem in a meta-learning paradigm. As an overview, we first compute the cross-entropy loss of unlabeled samples using their corresponding pseudo labels. Then we reweight the loss on each unlabeled sample by solving a meta optimization problem that minimizes the supervised loss of labeled samples. As directly solving the meta problem is computationally intractable, we propose an approximation method to dynamically obtain the 0–1 approximate solutions, which only requires one meta gradient descent step. In addition, theoretical guarantees are provided to show that our method converges to the stationary point of the supervised loss.

2.1 Meta optimization problem

We start by presenting the weighted loss function of our method, and defining a meta optimization problem to determine the value of the weight for each unlabeled sample.

Suppose that the networks are trained with stochastic gradient descent (SGD). At each iteration, we sample a mini-batch of labeled samples $\mathcal{X}=\{(\boldsymbol{x}_i,\boldsymbol{y}_i)\}$ together with a mini-batch of unlabeled samples $\mathcal{U}=\{(\boldsymbol{u}_j,\hat{\boldsymbol{y}}_j)\}$, where $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ represent the i-th labeled sample and its associated ground truth label, respectively, and $\boldsymbol{u}_j$ and $\hat{\boldsymbol{y}}_j$ represent the j-th unlabeled sample and its pseudo label, respectively. Following earlier work[10,17], we use the MixUp augmentation[53] to generate a mixed version of the inputs to improve the generalization performance, instead of directly using $\mathcal{X}$ and $\mathcal{U}$. The augmented mini-batch of training samples are denoted by $\tilde{\mathcal{X}}=\{(\tilde{\boldsymbol{x}}_i,\tilde{\boldsymbol{y}}_i)\}$ and $\tilde{\mathcal{U}}=\{(\tilde{\boldsymbol{u}}_j,\hat{\boldsymbol{y}}_j)\}$.

We defer the details on generating pseudo labels and obtaining $\tilde{\mathcal{X}}$ and $\tilde{\mathcal{U}}$ to Section 2.3.

Consider training a deep network with parameters $\boldsymbol{\theta}$. We first feed an unlabeled sample $\tilde{\boldsymbol{u}}_j$ into the network, producing the prediction $p(\tilde{\boldsymbol{u}}_j;\boldsymbol{\theta})$. Then we calculate the cross-entropy loss $L(\hat{\boldsymbol{y}}_j,p(\tilde{\boldsymbol{u}}_j;\boldsymbol{\theta}))$ using the corresponding soft pseudo label $\hat{\boldsymbol{y}}_j$. The loss of each unlabeled sample will be reweighed by $w_j^*\in[0,1]$ to construct the final loss function

$$\mathcal{L}_{\text{meta}}=\frac{1}{\sum_{j=1}^{|\tilde{\mathcal{U}}|}w_j^*}\sum_{j=1}^{|\tilde{\mathcal{U}}|}w_j^*L(\hat{\boldsymbol{y}}_j,p(\tilde{\boldsymbol{u}}_j;\boldsymbol{\theta})) \tag{1}$$

Without loss of generality, we assume $\mathcal{L}_{\text{meta}}=0$ when $\sum_{j=1}^{|\tilde{\mathcal{U}}|}w_j^*=0$.

The value of the weight scalar w_j^* is determined by minimizing the meta loss on the labeled data. To illustrate this procedure, we assume that w_j is the variable from which w_j^* can be solved, and consider training the network with the weighted loss:

$$\boldsymbol{\theta}^*(\boldsymbol{w})=\arg\min_{\boldsymbol{\theta}}\sum_{j=1}^{|\tilde{\mathcal{U}}|}w_jL(\hat{\boldsymbol{y}}_j,p(\tilde{\boldsymbol{u}}_j;\boldsymbol{\theta})) \tag{2}$$

where $\boldsymbol{\theta}^*(\boldsymbol{w})$ is the optimal solution that minimizes the weighted loss. Obviously, it is a function of the weight vector $\boldsymbol{w}=[w_1,w_2,\ldots,w_n]^{\mathrm{T}}$. Then the weights $\boldsymbol{w}^*$ is solved by minimizing the loss with $\boldsymbol{\theta}^*(\boldsymbol{w})$ on the labeled data $\tilde{\mathcal{X}}$, namely

$$\boldsymbol{w}^*=\arg\min_{w_j\in[0,1],\,j=1,2,\ldots,|\tilde{\mathcal{U}}|}\sum_{i=1}^{|\tilde{\mathcal{X}}|}L(\tilde{\boldsymbol{y}}_i,p(\tilde{\boldsymbol{x}}_i;\boldsymbol{\theta}^*(\boldsymbol{w}))) \tag{3}$$

Intuitively, our aim is to find a subset of pseudo-labeled samples, which, if used for training, are the most beneficial in terms of the generalization performance. Our motivation is drawn from the independent and identically distributed (i.i.d). assumption, i.e., the correctly "pseudo-labeled" data should has a similar distribution to the labeled data, and hence if the network is trained with the former, the final loss on the latter should be minimized as well.

Notably, here the labeled data are leveraged to determine if each pseudo-labeled sample should be used, instead of directly being used for training, as done in most existing SSL algorithms[10,14–17]. We argue that this is a more effective approach to exploit the supervision information.

2.2 Approximating the meta solution

To solve the meta optimization problem Eqs. (2) and (3) efficiently, we introduce an algorithm to obtain an approximate solution.

At t-th step in the training process, consider estimating $\boldsymbol{\theta}^*(\boldsymbol{w})$ by performing M times of gradient descents starting from current values of network parameter $\boldsymbol{\theta}^t$:

$$\overline{\boldsymbol{\theta}}_M^t\approx\boldsymbol{\theta}^*(\boldsymbol{w}),\ \ \overline{\boldsymbol{\theta}}_0^t=\boldsymbol{\theta}^t \tag{4}$$

$$\overline{\boldsymbol{\theta}}_{m+1}^t=\overline{\boldsymbol{\theta}}_m^t-\alpha^t\left[\frac{\partial\sum_{j=1}^{|\tilde{\mathcal{U}}|}w_jL(\hat{\boldsymbol{y}}_jp(\tilde{\boldsymbol{u}}_j;\overline{\boldsymbol{\theta}}_m^t))}{\partial\overline{\boldsymbol{\theta}}_m^t}\right],m=0,1,\ldots,M-1 \tag{5}$$

where α^t is the learning rate. As SGD has proven to be effective for optimizing deep networks, $\overline{\boldsymbol{\theta}}_M^t$ is a reliable alternate of $\boldsymbol{\theta}^*(\boldsymbol{w})$ as long as M is sufficiently large.

Given that $\boldsymbol{\theta}^*(\boldsymbol{w})$ can be estimated by $\overline{\boldsymbol{\theta}}_M^t$, a naive method of approximating $\boldsymbol{w}^*$ is to further estimate the gradient $\nabla_{\boldsymbol{w}}\sum_{i=1}^{|\tilde{\mathcal{X}}|}L(\tilde{\boldsymbol{y}}_i,p(\tilde{\boldsymbol{x}}_i;\boldsymbol{\theta}^*(\boldsymbol{w})))$ with $\overline{\boldsymbol{\theta}}_M^t$, and then repeatedly update $\boldsymbol{w}$ following similar gradient based optimization algorithms. However, it is computationally intensive to do that since updating $\boldsymbol{w}$ for N times requires MN steps of gradient descents on the network parameters. To get a efficient estimate of $\boldsymbol{w}^*$, we propose a dynamic approximation approach in the following.

First, to reduce the iterations of updating $\boldsymbol{w}$, we exploit a first order Taylor approximation of Eq. (3) at $\boldsymbol{w}=0$:

$$\boldsymbol{w}^*\approx\arg\min_{w_j\in[0,1],\,j=1,2,\ldots,|\tilde{\mathcal{U}}|}\boldsymbol{w}^T\left[\frac{\partial\sum_{i=1}^{|\tilde{\mathcal{X}}|}L(\tilde{\boldsymbol{y}}_i,p(\tilde{\boldsymbol{x}}_i;\overline{\boldsymbol{\theta}}_M^t))}{\partial\boldsymbol{w}}\right|_{\boldsymbol{w}=0}] \tag{6}$$

Notably, $\overline{\boldsymbol{\theta}}_M^t$ is obtained using the gradients of the weighted loss according to Eq. (5), and thus it is differentiable with respect to w_j. As the optimization objective in Eq. (6) is linear, it is straightforward to derive the solution:

Algorithm 1 Meta-Semi Algorithm

Initialize: $\boldsymbol{\theta}^0$
for $t=1$ **to** T **do**
  Randomly sample $\mathcal{X}$, $\mathcal{U}$
  Generate $\tilde{\mathcal{X}}$, $\tilde{\mathcal{U}}$
  Compute $p(\tilde{\boldsymbol{u}}_j;\boldsymbol{\theta}^t)$, $\tilde{\boldsymbol{u}}_j\in\tilde{\mathcal{U}}$
  $\boldsymbol{w}\leftarrow 0$, $\overline{\boldsymbol{\theta}}_0^t\leftarrow\boldsymbol{\theta}^t$
  $\nabla_{\overline{\boldsymbol{\theta}}_0^t}\leftarrow\frac{\partial\sum_{j=1}^{|\tilde{\mathcal{U}}|}w_jL(\hat{\boldsymbol{y}}_j,p(\tilde{\boldsymbol{u}}_j;\boldsymbol{\theta}^t))}{\partial\boldsymbol{\theta}^t}$
  $\overline{\boldsymbol{\theta}}_1^t\leftarrow\overline{\boldsymbol{\theta}}_0^t-\alpha^t\nabla_{\overline{\boldsymbol{\theta}}_0^t}$
  Compute $p(\tilde{\boldsymbol{x}}_i;\overline{\boldsymbol{\theta}}_1^t)$, $\tilde{\boldsymbol{x}}_i\in\tilde{\mathcal{X}}$
  Meta gradient: $\nabla_{\boldsymbol{w}}^t\leftarrow\frac{\partial\sum_{i=1}^{|\tilde{\mathcal{X}}|}L(\tilde{\boldsymbol{y}}_i,p(\tilde{\boldsymbol{x}}_i;\overline{\boldsymbol{\theta}}_1^t))}{\partial\boldsymbol{w}}$
  $\boldsymbol{w}^t\leftarrow\text{sign}(\max(-\nabla_{\boldsymbol{w}}^t,0))$ (Eq. (9))
  $\mathcal{L}_{\text{meta}}\leftarrow\frac{1}{\sum_{j=1}^{|\tilde{\mathcal{U}}|}w_j^t}\sum_{j=1}^{|\tilde{\mathcal{U}}|}w_j^tL(\hat{\boldsymbol{y}}_j,p(\tilde{\boldsymbol{u}}_j;\boldsymbol{\theta}^t))$
  $\boldsymbol{\theta}^{(t+1)}\leftarrow\boldsymbol{\theta}^t-\alpha^t\frac{\partial\mathcal{L}_{\text{meta}}}{\partial\boldsymbol{\theta}^t}$
end for

$$w_j^* \approx w_j^t = \begin{cases} 1 & \left.\dfrac{\partial \sum_{i=1}^{|\tilde{\mathcal{X}}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i; \overline{''\boldsymbol{\theta}}_M^t))}{\partial w_j}\right|_{w=0} \leqslant 0 \\ 0 & \left.\dfrac{\partial \sum_{i=1}^{|\tilde{\mathcal{X}}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i; \overline{''\boldsymbol{\theta}}_M^t))}{\partial w_j}\right|_{w=0} > 0 \end{cases} \tag{7}$$

where w_j^t denotes the approximate solution of w_j^*. The required steps of gradient descents are reduced to M from MN by leveraging Formula (7). However, the algorithm is still inefficient since a large M is necessary to get a sufficiently accurate $\overline{\boldsymbol{\theta}}_M^t$. To further reduce the computational cost, an intriguing property can be leveraged. In the following proposition, we show that the results of Formula (7) will remain the same if $\overline{\boldsymbol{\theta}}_M^t$ in the equation is replaced by $\overline{\boldsymbol{\theta}}_1^t$. In other words, Formula (7) can be precisely solved using $\overline{\boldsymbol{\theta}}_1^t$ instead of $\overline{\boldsymbol{\theta}}_M^t$, and the former only needs one gradient descent step to obtain.

Proposition 1. Suppose that $\overline{\boldsymbol{\theta}}_M^t$ is given by M steps of gradient descents starting from $\overline{\boldsymbol{\theta}}_0^t = \boldsymbol{\theta}^t$. Then we have

$$\left.\frac{\partial \sum_{i=1}^{|\tilde{\mathcal{X}}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i; \overline{\boldsymbol{\theta}}_M^t))}{\partial w_j}\right|_{w=0} = M \left[\left.\frac{\partial \sum_{i=1}^{|\tilde{\mathcal{X}}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i; \overline{\boldsymbol{\theta}}_1^t))}{\partial w_j}\right|_{w=0}\right], \tag{8}$$

$\forall 1 \leqslant j \leqslant |\tilde{\mathcal{U}}|$.

Proof can be accessed in the Electronic Supplementary Material (ESM).

With Proposition 1, we are ready to present the final form of our dynamically reweighting formula:

$$w_j^t = \begin{cases} 1, & \left.\dfrac{\partial \sum_{i=1}^{|\tilde{\mathcal{X}}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i; \overline{\boldsymbol{\theta}}_1^t))}{\partial w_j}\right|_{w=0} \leqslant 0; \\ 0, & \left.\dfrac{\partial \sum_{i=1}^{|\tilde{\mathcal{X}}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i; \overline{\boldsymbol{\theta}}_1^t))}{\partial w_j}\right|_{w=0} > 0 \end{cases} \tag{9}$$

As we leverage a meta learning approach to reweight different pseudo-labeled samples, we call our method Meta-Semi. The pseudo code of Meta-Semi is presented in Algorithm 1. In summary, after each standard forward step of the pseudo-labeled samples, we first update the parameters with the loss of all samples weighted by zero. Such a meta updating step does not change the values of parameters, but construct a differentiable computational graph. Then we calculate the supervised loss on labeled data, and exploit the computational graph to take the derivative of the supervised loss with respect to the zero weight, which is called "meta gradient". Finally, we only use the pseudo-labeled samples with negative meta gradients to train the network.

Interpretation of meta gradients. A straightforward way to interpret the meta gradients is that it can be viewed as the influence on the supervised loss when the weight of certain pseudo-labeled sample changes slightly around zero during training. In fact, there exists a more intriguing and interesting interpretation. The meta gradients given in Formula (9) can be expressed as

$$\left.\frac{\partial \sum_{i=1}^{|\tilde{\mathcal{X}}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i; \overline{\boldsymbol{\theta}}_1^t))}{\partial w_j}\right|_{w=0} = \left[\frac{\partial \sum_{i=1}^{|\tilde{\mathcal{X}}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i; \overline{\boldsymbol{\theta}}_1^t))}{\partial \overline{\boldsymbol{\theta}}_1^t}\right]^{\mathrm{T}} \left[\left.\frac{\partial (\overline{\boldsymbol{\theta}}_0^t - \alpha^t \nabla_{\overline{\boldsymbol{\theta}}_0^t})}{\partial w_j}\right|_{w=0}\right] = -\alpha^t \left[\frac{\partial \sum_{i=1}^{|\tilde{\mathcal{X}}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i; \boldsymbol{\theta}^t))}{\partial \boldsymbol{\theta}^t}\right]^{\mathrm{T}} \left[\frac{\partial L(\hat{\boldsymbol{y}}_j, p(\tilde{\boldsymbol{u}}_j; \boldsymbol{\theta}^t))}{\partial \boldsymbol{\theta}^t}\right] \tag{10}$$

which follows from $\nabla_{\overline{\boldsymbol{\theta}}_0^t} = \sum_{k=1}^{|\tilde{\mathcal{U}}|} w_k \frac{\partial L(\tilde{\boldsymbol{y}}_k, p(\tilde{\boldsymbol{u}}_k; \overline{\boldsymbol{\theta}}_0^t))}{\partial \overline{\boldsymbol{\theta}}_0^t}$ and $\overline{\boldsymbol{\theta}}_1^t = \overline{\boldsymbol{\theta}}_0^t = \boldsymbol{\theta}^t$. For the pseudo-unlabeled sample $(\tilde{\boldsymbol{u}}_j, \hat{\boldsymbol{y}}_j)$, its meta gradient is negatively proportional to the inner product of the average gradient of labeled samples and the gradient produced by itself. In other words, the sign of the meta gradient indicates whether the angle between the former and the later is larger than 90 degrees.

Intuitively, if the pseudo label is correct, the corresponding gradient should guide the model towards a similar direction to the labeled samples, or at least should not be largely different from the supervised gradient in direction. In essence, Meta-Semi trains deep networks using pseudo-labeled samples whose gradient directions are similar to labeled samples. An illustration is shown in Fig. 1.

2.3 Implementation details

Pseudo labels. To obtain high quality pseudo labels for the original unlabeled mini-batch $\mathcal{U}$, we first apply an exponential moving average (EMA) on model parameters, which has proven to be effective in providing supervision on unlabeled data[10, 15]. Then we feed every unlabeled sample $\boldsymbol{u}_j$ in $\mathcal{U}$ into the EMA model, and take the corresponding softmax prediction as the soft pseudo label $\hat{\boldsymbol{y}}_j$.

MixUp augmentation is an important regularization technique

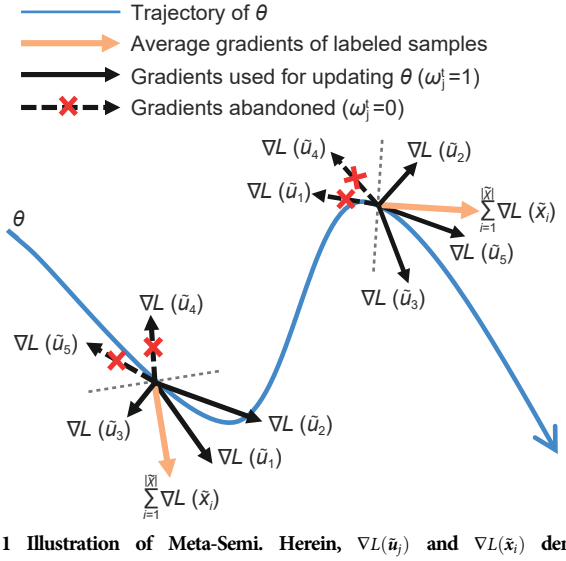

Fig. 1 Illustration of Meta-Semi. Herein, $\nabla L(\tilde{\boldsymbol{u}}_j)$ and $\nabla L(\tilde{\boldsymbol{x}}_i)$ denote $\nabla_{\boldsymbol{\theta}^t} L(\hat{\boldsymbol{y}}_j, p(\tilde{\boldsymbol{u}}_j; \boldsymbol{\theta}^t))$ and $\nabla_{\boldsymbol{\theta}^t} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i; \boldsymbol{\theta}^t))$, respectively. Our method trains the networks with pseudo-labeled samples whose gradient directions are similar to the average gradient of labeled samples.

used by state-of-the-art deep SSL algorithms[10,17]. It improves the generalization performance of models by encouraging the "convex" behavior between different samples. Given a pair of samples with corresponding annotations, $(\boldsymbol{x}_1, \boldsymbol{y}_1)$ and $(\boldsymbol{x}_2, \boldsymbol{y}_2)$, MixUp is performed to generate an augmented sample via linear interpolation:

$$\tilde{\boldsymbol{x}} = \lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2, \quad \tilde{\boldsymbol{y}} = \lambda \boldsymbol{y}_1 + (1-\lambda)\boldsymbol{y}_2 \tag{11}$$

where λ is sampled from a pre-defined Beta distribution. The addition operation in Eq. (11) is directly performed in the pixel and label space.

In Meta-Semi, we leverage MixUp to generate the mixed training data $\tilde{\mathcal{X}}$ and $\tilde{\mathcal{U}}$. Formally, $\tilde{\mathcal{X}}$ is obtained from only the labeled set $\mathcal{X}$:

$$\tilde{\mathcal{X}} = \text{MixUp}(\mathcal{X}, \text{Shuffle}(\mathcal{X}), \lambda_1), \quad \lambda_1 \sim \text{Beta}(\beta, \beta) \tag{12}$$

where β is the parameter of the Beta distribution, and it is the only tunable hyper-parameter (excluding the hyper-parameters of a supervised learning algorithm) in our algorithm. With regards to $\tilde{\mathcal{U}}$, we ideally want the unlabeled data to extract more information from the labeled samples. Therefore, we first concatenate $\mathcal{X}$ and $\mathcal{U}$ together, and then apply the MixUp procedure:

$$\begin{aligned} \tilde{\mathcal{U}} &= \text{MixUp}(\mathcal{W}, \text{Shuffle}(\mathcal{W}), \lambda_2) \\ \mathcal{W} &= \text{Concat}(\mathcal{X}, \mathcal{U}) \lambda_2 \sim \text{Beta}(\beta, \beta) \end{aligned} \tag{13}$$

where the one-hot ground truth labels are used for $\mathcal{X}$ and the soft pseudo labels are used for $\mathcal{U}$.

Compatibility with consistency based methods.

As a matter of fact, Meta-Semi is compatible with existing consistency based algorithms, and they can be integrated when necessary. To see this, the regularization term can be simply appended to the loss function with an addition coefficient γ:

$$\mathcal{L} = \mathcal{L}_{\text{meta}} + \gamma \mathcal{L}_{\text{consistency}} \tag{14}$$

Notably, Meta-Semi includes only the first term $\mathcal{L}_{\text{meta}}$, and it has only one tunable hyper-parameter β as aforementioned. In experiments, we show that vanilla Meta-Semi has already achieved state-of-the-art performance consistently. The second term in Eq. (14) may introduce additional tunable hyper-parameters that inherently exist in the combined approaches. It is able to effectively improve the accuracy on top of Meta-Semi at the cost of additional hyper-parameter searching cost.

2.4 Convergence analysis

In this section, we show theoretically that under some mild conditions, our method converges to the stationary point of the loss on labeled data. The convergence results of SGD based optimization methods with a fixed loss function has been well-known[54]. However, it is still necessary to provide the convergence analysis of Meta-Semi since the optimization objective of our method is dynamically changed. To make it clear, we first define the supervised loss on the labeled mini-batch $\tilde{\mathcal{X}}$ by

$$G(\tilde{\mathcal{X}}, \boldsymbol{\theta}^t) = \sum_{i=1}^{|\tilde{\mathcal{X}}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i; \boldsymbol{\theta}^t)) \tag{15}$$

Thus, the expected loss on all the labeled data is $\mathbb{E}_{\tilde{\mathcal{X}}} G(\tilde{\mathcal{X}}, \boldsymbol{\theta}^t)$. Then we introduce the definition of Lipschitz-smooth and a mild assumption stating that the expected norm of gradients used for updating model parameters will not get too large compared with the gradient of the overall supervised loss.

Definition 1 A function $f: \mathbb{R}^n \to \mathbb{R}$ is said to be Lipschitz-smooth with constant L if

$$\|\nabla f(x) - \nabla f(y)\| \leqslant L\|x - y\|, \quad \forall x, y \in \mathbb{R}^n$$

Assumption 1. For all $t \geqslant 0$, there exists a positive scalar σ, such that

$$\mathbb{E}_{\tilde{\mathcal{X}}, \tilde{\mathcal{U}}} \|\nabla_{\boldsymbol{\theta}^t} \mathcal{L}_{\text{meta}}\|^2 \leqslant \sigma \|\nabla_{\boldsymbol{\theta}^t} \mathbb{E}_{\tilde{\mathcal{X}}} G(\tilde{\mathcal{X}}, \boldsymbol{\theta}^t)\|^2.$$

In fact, the assumption is not very strong. Roughly, since $\mathcal{L}_{\text{meta}}$ is computed using the ground truth labels and the pseudo labels based on the prediction of the EMA model, it is usually very close to the minima of the loss function, especially when the networks tend to be stable with sufficiently large t. Empirically, we show that Assumption 1 holds in many cases of SSL, which is shown in Fig. 2. Under this condition, the following proposition shows that our method converges to the stationary point of the loss on labeled data with proper learning rate schedules.

Proposition 2. Assume that the loss function on labeled data $G(\tilde{\mathcal{X}}, \boldsymbol{\theta}^t)$ is Lipschitz-smooth with regards to $\boldsymbol{\theta}^t$ for all $\tilde{\mathcal{X}}$, and that Assumption 1 holds. Suppose also that the learning rate $\alpha^t > 0$ satisfies:

$$\lim_{t \to \infty} \alpha^t = 0, \sum_{t=0}^{\infty} \alpha^t = \infty \tag{16}$$

Then every limit point of the sequence $\{\boldsymbol{\theta}^t\}$ generated by Meta-Semi is a stationary point of $\mathbb{E}_{\tilde{\mathcal{X}}} G(\tilde{\mathcal{X}}, \boldsymbol{\theta}^t)$, namely,

$$\lim_{t \to \infty} \|\nabla_{\boldsymbol{\theta}^t} \mathbb{E}_{\tilde{\mathcal{X}}} G(\tilde{\mathcal{X}}, \boldsymbol{\theta}^t)\| = 0.$$

Proof can be accessed in the ESM.

3 Experiment

In this section, we empirically evaluate the effectiveness of the

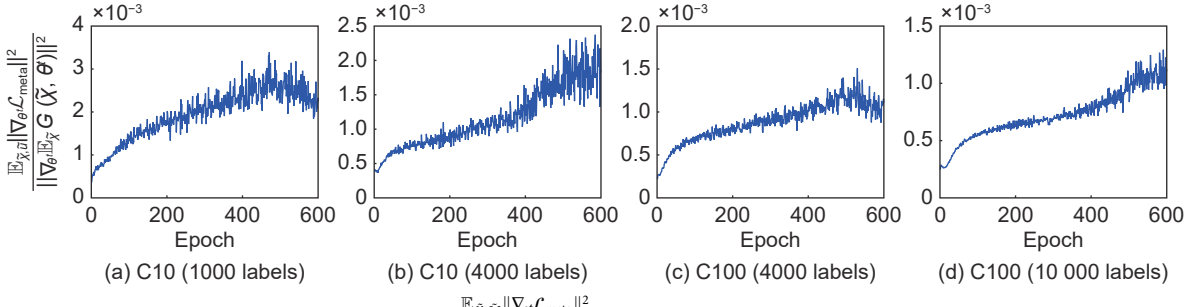

Fig. 2 The empirical validation of Assumption 1. The value of $\frac{\mathbb{E}_{\tilde{\mathcal{X}},\tilde{\mathcal{U}}}\|\nabla_{\boldsymbol{\theta}^t}\mathcal{L}_{\text{meta}}\|^2}{\|\nabla_{\boldsymbol{\theta}^t}\mathbb{E}_{\tilde{\mathcal{X}}}G(\tilde{\mathcal{X}},\boldsymbol{\theta}^t)\|^2}$ is estimated at each training epoch using Monte-Carlo sampling with a sample size 500. Results on CIFAR-10 (C10) and CIFAR-100 (C100) with varying numbers of labeled samples are presented. It can be observed that the ratio generally increases before the 500th epoch, but gradually becomes stable or even decreases in the last part of the training process when the learning rate approaches 0. Therefore, it is empirically reasonable to assume that Assumption 1 holds.

proposed Meta-Semi method, analyze its time complexity experimentally, and give sensitivity tests as well as ablation studies. All experiments are conducted using a single Nvidia Titan Xp GPU.

3.1 Experimental setup

Our experiments are based on four widely used image classification benchmarks, i.e., CIFAR-10/100[55], SVHN[56], and STL-10[57], and two modern deep networks, i.e., a 13-layer CNN (CNN-13) and the Wide-ResNet-28-2 (WRN-28-2). On CIFAR and SVHN, we randomly preserve the labels of certain numbers of samples (identical for each class), and remain all other samples unlabeled. On STL-10, we use pre-defined folds. The experimental settings on data pre-processing, training/validation splitting, training configurations and baselines follow the common practice of SSL[9,10,15,17,58]. The hyper-parameter β of Meta-Semi is selected among $[0.2, 1]$ on the validation set.

Datasets. (1) The CIFAR-10 / CIFAR-100 datasets consist of 60 000 32×32 colored images of 10/100 classes, 50 000 for training and 10 000 for test. Following the common practice of SSL[9,10,15,17,58], we hold out 5000 images from the training set as the validation set. Images are normalized with channel means and standard deviations for pre-processing. Then data augmentation is performed by 4×4 random translation followed by random horizontal flip[5,6]. On CIFAR-10, we preserve 100, 200, and 400 labels per class respectively, corresponding to 1000, 2000, 4000 labeled samples in total. All other samples are unlabeled. We randomly split the dataset for 5 times to conduct multiple experiments, and report the mean test errors associated with standard deviations. Similarly, On CIFAR-100, evaluation is performed with 40 and 100 randomly preserved labeled samples per class. (2) SVHN consists of 32×32 colored images of digits. 73 257 images for training, 26 032 images for testing and 531 131 images for additional training are provided. Following[15,62], we merely perform random 2×2 translation to augment the training set, and hold out 1000 images for validation. Similar to CIFAR, we randomly preserve 500 and 1000 labels for experiments. (3) STL-10[57] contains 5000 training examples divided into 10 predefined folds with 1000 examples each, and 100 000 unlabeled images drawn from a similar—but not identical—data distribution. All the samples are 96×96 colored images. We use the same experimental protocol as Ref. [17].

Networks. Our experiments are based on a 13-layer CNN (CNN-13) and the Wide-RestNet-28-2 (WRN-28-2) network. The CNN-13 network has been adopted as the standard model for experiments by state-of-the-art SSL algorithms[10,15,16,27,58,62]. Following Ref. [10], we remove the Gaussian noise layer and the dropout layer in the network. Other methods use these techniques if mentioned in their original papers, which provide stronger regularization. Some recent works adopt the WRN-28-2 network[9,17] in their experiments. We also implement Meta-Semi with WRN-28-2 to present comparisons with them.

Training details. The CNN-13 network uses the SGD optimizer with a Nesterov momentum of 0.9. The L2 regularization coefficient is set to 10^{-4}, and the initial learning rate is set to 0.1. For all experiments with CNN-13, we train the network for 600 epochs using the cosine learning rate annealing technique[10,63,64]. The batch size of labeled samples and unlabeled samples is set to 25 and 75 respectively. To generate pseudo labels for unlabeled samples, we use an exponential moving average on model parameters with a decay rate of 0.999. For WRN-28-2, we adopt the same training details as Ref. [17]. Particularly, the batch size is set to 32 for labeled samples and 96 for unlabeled samples, such that the total batch size is the same as Ref. [17]. The ratio of labeled/unlabeled samples in each mini-batch is always set to 1:3 in Meta-Semi, which consistently achieves excellent performance on the validation set, and does not need to be tuned for the specific SSL task.

Baselines. Our method is compared with several state-of-the-art baselines including SSL algorithms and a meta-reweighting method.

- Π-model[14] enforces the model predictions to remain the same when different augmentation and dropout modes are performed.
- Temp-ensemble[14] attaches a soft pseudo label for each unlabeled sample by performing a moving average on the historical predictions of networks.
- Mean teacher (MT)[15] establishes a teacher network by performing exponential moving average on the parameters of the model, and leverages the teacher networks to produces supervision for unlabeled data.
- Virtual adversarial training (VAT)[16] adds adversarial perturbations to the samples and enforce the model to have the same predictions on perturbed samples and the original samples.
- Smooth neighbors on teacher graphs (SNTG)[62] constructs a teacher graph to regularize the feature distribution of unlabeled samples.
- Learning to Reweight[45] proposes to reweight different training samples by solving a similar meta-learning problem to us. Since their original algorithm requires labels of all the training, we adopt a version modified for SSL in this paper. In specific, we retain our approach of generating pseudo-labeled samples, but use their reweighting strategy.
- MT + Fast SWA[58] is an improved MT algorithm using a fast stochastic weight averaging optimizer.
- Interpolation consistency training (ICT)[10] encourages the prediction on an interpolation of unlabeled samples to be consistent with the interpolation of the predictions on those points. They first use MixUp augmentation in deep SSL.
- MixMatch[17] is a holistic deep SSL approach that integrates various dominant consistency regularization techniques.

We implement these methods in the same codebase, and search for the best hyper-parameters for them on the validation set according to the recommendations provided by their original papers. Notably, for MixMatch[17], we fix the sharpening temperature $T = 0.5$ and the number of unlabeled augmentations $K = 2$, and adjust the α parameter for Beta distribution and the unsupervised loss coefficient $\lambda_{\tilde{U}}$, as suggested by the paper. We first reproduce the CIFAR-10 results of MixMatch reported by their paper, and then tune α and $\lambda_{\tilde{U}}$ on the validation set of CIFAR-100.

3.2 Main results

Results on CIFAR with various numbers of labeled samples are presented in Table 1. It can be observed that Meta-Semi consistently outperforms state-of-the-art SSL algorithms in terms of generalization performance, especially with relatively less labeled data and larger numbers of classes. For example, when using CNN-13, on CIFAR-10 with 4 000 labels, Meta-Semi outperforms the competitive baseline, ICT, by 0.13% in absolute error, while with 1000 labels on CIFAR-10 and with 4000 labels on CIFAR-100, Meta-Semi yields more significant improvements of 2.17% and 2.46%, respectively. MixMatch shows robust performance with small labeled sets as well, but Meta-Semi outperforms it in terms of test accuracy. Moreover, it is shown that the performance of Meta-Semi can be significantly improved

Table 1 Performance of Meta-Semi and state-of-the-art SSL algorithms on CIFAR with varying amount of labeled data. We report the average test errors and the standard deviations of 5 trials. † and ‡ refer to the experiments using ResNet-18 and WRN-28-2, while all the others use CNN-13. In each setting, the best two results with CNN-13 and the best result with WRN-28-2 are bold-faced. (%)

Dataset	CIFAR-10				CIFAR-100	
	0 labels	1000 labels	2000 labels	4000 labels	4000 labels	10 000 labels
Supervised learning	–	39.95±0.75	27.67±0.12	20.42±0.21	58.31±0.89	44.56±0.30
Supervised learning + MixUp[53]	–	31.83±0.65	24.22±0.15	17.37±0.35	54.87±0.07	40.97±0.47
Self-supervised learning (SimCLR) †[59]	10.24±0.12	–	–	–	–	–
Self-supervised learning (MoCo) †[60]	9.88±0.12	–	–	–	–	–
Self-supervised learning (SimSiam) †[61]	9.41±0.11	–	–	–	–	–
Π-model[14]	–	28.74±0.48	17.57±0.44	12.36±0.17	55.39±0.55	38.06±0.37
Temp-ensemble[14]	–	25.15±1.46	15.78±0.44	11.90±0.25	–	38.65±0.51
Mean Teacher[15]	–	18.27±0.53	13.45±0.30	10.73±0.14	45.36±0.49	35.96±0.77
VAT[16]	–	18.12±0.82	13.93±0.33	11.10±0.24	–	–
SNTG[62]	–	18.41±0.52	13.64±0.32	10.93±0.14	–	37.97±0.29
Learning to Reweight[45]	–	11.74±0.12	–	9.44±0.17	46.62±0.29	37.31±0.47
MT + Fast SWA[58]	–	15.58	11.02	9.05	–	33.62±0.54
ICT[10]	–	12.44±0.57	8.69±0.15	7.18±0.24	40.07±0.38	32.24±0.16
Meta-Semi	–	**10.27±0.66**	**8.42±0.30**	**7.05±0.27**	**37.61±0.56**	**30.51±0.32**
Meta-Semi + ICT	–	**9.29±0.62**	**7.05±0.12**	**6.42±0.18**	**37.12±0.59**	**29.68±0.05**
Mean Teacher ‡[15]	–	17.32±4.00	12.17±0.22	10.36±0.25	–	–
MixMatch ‡[17]	–	7.75±0.32	7.03±0.15	6.24±0.06	–	30.84±0.29
Meta-Semi ‡	–	**7.34±0.22**	**6.58±0.07**	**6.10±0.10**	–	**29.69±0.18**

by combining it with consistency based methods. On CIFAR-10 with 2000 labels, Meta-Semi + ICT outperforms Meta-Semi by 1.37%.

Discussions: supervised, self-supervised, and semi-supervised learning. In Table 1, we also report the results of fully supervised learning and self-supervised learning. For the former, only the labeled data is leveraged to train the model. For the latter, the model is trained with all the data unlabeled[59–61] and evaluated with the KNN evaluation protocol[68]. We report the off-the-shelf self-supervised learning results[69] with ResNet-18 (i.e., a larger model than CNN-13/WRN-28-2). One can observe that supervised learning achieves the lowest accuracy, since only 2%–20% of the data is utilized for training, though being annotated. SSL improves the accuracy on top of it by further leveraging the unlabeled data. In addition, SSL outperforms self-supervised learning methods as well, due to its access to the labels of a small subset of the data. Nevertheless, it is totally feasible to fine-tune the self-supervised pre-trained models using SSL algorithms[70], and hence they are actually compatible with each other.

Results on STL-10 and SVHN are presented in Tables 2 and 3, respectively. The results indicate that the test accuracy of Meta-Semi outperforms MixMatch by more than 2% on STL-10, and is comparable with state-of-the-art SSL algorithms on SVHN.

Semi-supervised few-shot learning. Our method is compatible with the semi-supervised few-shot learning methods. In specific, Meta-Semi can pre-train the backbone networks effectively using both the labeled and unlabeled samples from the base classes, which have a relatively sufficient amount of training data. On its basis, the semi-supervised few-shot learning algorithms is able to be deployed by fine-tuning the pre-trained backbones, and exploiting the labeled/unlabeled data from the novel classes for accurate inference. The experimental results to demonstrate this point are shown in Table 4. Two widely used benchmark datasets, i.e., mini-ImageNet[42] and tiered-ImageNet[43] are considered. To

Table 2 Test errors on STL-10. We adopt the same experimental setups as Ref. [17]. The best result is bold-faced. (%)

Method	STL-10
	1000 labels
SWWAE[65]	25.70
CC-GAN[66]	22.20
MixMatch[17]	10.18±1.46
Meta-Semi	**8.03±0.24**

Table 3 Test errors on SVHN with varying amount of labeled data. We report the average results and the standard deviations of 5 independent experiments. All results are based on CNN-13. The best results are bold-faced. (%)

Method	SVHN	
	500 labels	1000 labels
VAT[16]	–	5.42
Π-model[14]	6.65±0.53	4.82±0.17
Temp-ensemble[14]	5.12±0.13	4.42±0.16
Mean teacher[15]	4.18±0.27	3.95±0.19
ICT[10]	4.23±0.15	3.89±0.04
SNTG[62]	3.99±0.24	3.86±0.27
Meta-Semi	4.12±0.21	3.92±0.11
Meta-Semi + ICT	**3.98±0.09**	**3.77±0.05**

ensure the fair comparisons, all the results follow the same experimental setups in Refs. [43, 47, 48]. We implement a representative recently proposed approach, EPNet[50], with their official code, and compare the results with and without Meta-Semi in the pre-training stage. The former uses only the labeled data of base classes, while the later also leverages the unannotated data for pre-training. One can observe that Meta-Semi effectively

Table 4 Test accuracy of semi-supervised 5-way few-shot learning on mini-ImageNet and tiered-ImageNet. The experimental setups in Refs. [43, 47, 48] are adopted for all the experiments. In 5-way 1/5-shot learning, 30/50 unlabeled samples are included in the unlabeled set for each novel class. The best results are bold-faced. (%)

Method	Backbone	mini-ImageNet (5-way)		tiered-ImageNet (5-way)	
		1-shot (30)	5-shot (50)	1-shot (30)	5-shot (50)
Masked Soft k-Means[43]	CONV-4	50.40	64.40	52.40	69.90
TPN[47]	CONV-4	52.78	66.42	55.74	71.01
TransMatch[49]	WRN-28-10	60.02	79.30	72.19	82.12
LEO[51]	WRN-28-10	61.76	77.59	66.33	81.44
MTL[52]	ResNet-12	61.20	75.50	65.60	78.60
Masked soft k-Means[43] + MTL[52]	ResNet-12	62.10	73.60	68.60	81.00
TPN[47] + MTL[52]	ResNet-12	62.70	74.20	72.10	83.30
MetaOpt-SVM[67]	ResNet-12	62.64	78.63	65.99	81.56
LST[48]	ResNet-12	70.10	78.70	77.70	85.20
EPNet[50]	ResNet-12	69.93	80.23	79.29	86.03
EPNet[50] + Meta-Semi	ResNet-12	**73.24**	**83.30**	**81.30**	**88.08**

improves the test accuracy (by 2%–3%).

3.3 Hyper-parameter sensitivity

The β parameter for the Beta distribution in MixUp augmentation is the only additional hyper-parameter that needs to be tuned when Meta-Semi is implemented in new SSL tasks. To study the sensitivity of our method to β, we vary the value of β, and present the test errors in Fig. 3. For comparison, we also present the results of ICT[10] when its two additional hyper-parameters (β and the unsupervised regularization coefficient w) change among the recommended candidates provided by the original paper. One can observe that the performance of Meta-Semi is relatively stable when β ranges from 0.25 to 0.75. In implementation, $\beta = 0.5$ may yield a proper setting for the preliminary experiment or a good starting point for hyper-parameter searching. In contrast, ICT is sensitive to both the two hyper-parameters. It has been shown that hyper-parameter searching is difficult for realistic SSL tasks[9]. Meta-Semi can be more easily applied as it requires less effort for tuning hyper-parameters.

Another interesting observation is that the best performance of Meta-Semi is reached at $\beta = 0.5$, while the optimal β for ICT is mainly at 0.2 and 0.3. A plausible explanation for this phenomenon is that the proposed meta-reweighting mechanism of Meta-Semi is able to filter out the samples with incorrect

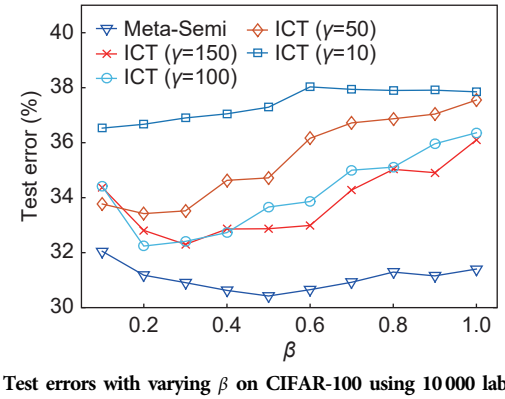

Fig. 3 Test errors with varying β on CIFAR-100 using 10 000 labels. The CNN-13 network is used. We also report the results of ICT[10] when the unsupervised consistency coefficient w changes among the recommended range.

pseudo labels. As a result, only the properly pseudo-labeled data is used for training, and thus one can leverage a stronger regularization technique (e.g., MixUp with a larger β). On the contrary, in ICT, a large β may make the model training problem overly difficult or even ill-posed, since the training data generated by MixUp incorporates many incorrect pseudo labels.

3.4 Efficiency of Meta-Semi

Our method generally requires more training time for each iteration as it includes bi-level optimization. However, we find that our algorithm converges fast and if we consider a fixed amount of training time, it still outperforms the others, as shown in Table 5.

3.5 Ablation study

To provide additional insights into our method, we further conduct the ablation experiments by removing or altering the components of Meta-Semi. The results are shown in Table 6. It can be seen that parameter EMA and performing MixUp on unlabeled data are both important techniques to achieve high generalization performance. The observation is consistent with Ref. [10]. In addition, if all pseudo-labeled samples are weighted by the constant 1, Meta-Semi is equivalent to a consistency based algorithm, which also shows effective performance.

4 Conclusion and Future Work

In this paper, we have presented a novel semi-supervised classification algorithm under the meta-learning paradigm. The proposed Meta-Semi algorithm is capable of adapting to various SSL tasks with impressive performance via tuning only one additional hyper-parameter, and empirically we have observed that the model performance is robust to different settings of this hyper-parameter. Compared to existing deep SSL algorithms, Meta-Semi requires much less effort for tuning hyper-parameters, but achieves state-of-the-art performance on four competitive datasets. Theoretically, we have provided the convergence analysis to show that Meta-Semi always converges to a stationary point under mild conditions. A limitation of Meta-Semi is that it slightly increases the training time. Our future work may focus on addressing this issue.

Table 5 Performance of Meta-Semi vs. baselines with fixed amount of training time. We report the mean test errors of both networks on CIFAR-100 with 10 000 labels. The best results are bold-faced. (%)

Method	CNN-13				WRN-28-2			
	5.0 h	7.5 h	10.0 h	12.6 h	13.7 h	18.3 h	22.8 h	29.2 h
ICT[10]	33.43	32.84	32.61	32.24	–	–	–	–
MixMatch[17]	–	–	–	–	32.94	31.91	31.26	30.84
Meta-Semi	**32.73**	**31.81**	**31.06**	**30.84**	**31.74**	**30.85**	**30.50**	**30.13**

Table 6 Ablation study results. We report the test errors on CIFAR-100 with 4000 and 10 000 labels. The CNN-13 network is used. (%)

Ablation	CIFAR-100	
	4000 labels	10 000 labels
Without parameter EMA	47.68±0.27	37.15±1.02
One-hot pseudo labels	41.52±0.51	32.78±0.41
MixUp on unlabeled data only	37.69±0.50	30.56±0.39
MixUp on labeled data only	45.90±0.15	36.11±0.21
Without MixUp	46.71±0.05	35.98±0.69
Reweighting with the constant 1	40.26±0.64	32.17±0.14
Reweighting with −1 and 1	45.41±0.38	36.39±0.44
Meta-Semi	37.61±0.56	30.51±0.32
Meta-Semi + ICT	37.12±0.59	29.68±0.05

Acknowledgment

This work was supported by the National Key R&D Program of China (No. 2019YFC1408703), the National Natural Science Foundation of China (No. 62022048), THU-Bosch JCML, and Beijing Academy of Artificial Intelligence. In particular, we appreciate the valuable discussion with Yitong Xia and Hong Zhang.

Electronic Supplementary Material

Supplementary materials including proof of Proposition 1 and Proposition 2 are available in the online version of this article at https://doi.org/10.26599/AIR.2022.9150011.

Article History

Received: 7 December 2022; Revised: 8 January 2023; Accepted: 17 January 2023

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ImageNet classification with deep convolutional neural networks, in *Proc. 25th Int. Conf. Neural Information Processing Systems*, Lake Tahoe, NE, USA, 2012, pp. 1097–1105.

[2] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, in *Proc. 3rd Int. Conf. Learning Representations*, San Diego, CA, USA, 2015.

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, Going deeper with convolutions, in *2015 IEEE Conf. Computer Vision and Pattern Recognition* (*CVPR*), Boston, MA, USA, 2015, pp. 1–9.

[4] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[5] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, in *2016 IEEE Conf. Computer Vision and Pattern Recognition* (*CVPR*), Las Vegas, NV, USA, 2016, pp. 770–778.

[6] G. Huang, Z. Liu, G. Pleiss, L. Van Der Maaten, and K. Q. Weinberger, Convolutional networks with dense connectivity, *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 44, no. 12, pp. 8704–8716, 2019.

[7] M. Wang, H. Li, X. Chen, and Y. Chen, Deep learning-based model reduction for distributed parameter systems, *IEEE Trans. Syst. Man Cybernet. Syst.*, vol. 46, no,12, pp. 1664–1674, 2016.

[8] A. I. Károly, P. Galambos, J. Kuti, and I. J. Rudas, Deep learning in robotics: Survey on model structures and training strategies, *IEEE Trans. Syst. Man Cybernet. Syst.*, vol. 51, no. 1, pp. 266–279, 2021.

[9] A. Oliver, A. Odena, C. Raffel, E. D. Cubuk, and I. J. Goodfellow, Realistic evaluation of deep semi-supervised learning algorithms, in *Proc. 32nd Int. Conf. Neural Information Processing Systems*, Montréal, Canada, 2018, pp. 3239–3250.

[10] V. Verma, A. Lamb, J. Kannala, Y. Bengio, and D. Lopez-Paz, Interpolation consistency training for semi-supervised learning, in *Proc. 28th Int. Joint Conf. Artificial Intelligence*, Macao, China, 2019, pp. 3635–3641.

[11] X. Zhu, Z. Ghahramani, and J. Lafferty, Semi-supervised learning using Gaussian fields and harmonic functions, in *Proc. Twentieth Int. Conf. Int. Conf. Machine Learning*, Washington, DC, USA, 2003, pp. 912–919.

[12] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning (Adaptive Computation and Machine Learning)*, Cambridge, MA, USA: MIT Press, 2006.

[13] J. Turian, L. A. Ratinov, and Y. Bengio, Word representations: A simple and general method for semi-supervised learning, in *Proc. 48th Ann. Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, 2010, pp. 384–394.

[14] S. Laine and T. Aila, Temporal ensembling for semi-supervised learning, arXiv preprint arXiv: 1610.02242, 2016.

[15] A. Tarvainen and H. Valpola, Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, in *Proc. 31st Int. Conf. Neural Information Processing Systems*, Long Beach, CA USA, 2017, pp. 1195–1204.

[16] T. Miyato, S. I. Maeda, M. Koyama, and S. Ishii, Virtual adversarial training: a regularization method for supervised and semi-supervised learning, *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1979–1993, 2019.

[17] D. Berthelot, N. Carlini, I. Goodfellow, A. Oliver, N. Papernot, and C. Raffel, MixMatch: A holistic approach to semi-supervised learning, in *Proc. 33rd Int. Conf. Neural Information Processing Systems*, Vancouver, Canada, 2019, p. 454.

[18] S. Y. Shin, S. Lee, I. D. Yun, S. M. Kim, and K. M. Lee, Joint weakly and semi-supervised deep learning for localization and classification of masses in breast ultrasound images, *IEEE Trans. Med. Imag.*, vol. 38, no. 3, pp. 762–774, 2019.

[19] Q. Liu, L. Yu, L. Luo, Q. Dou, and P. A. Heng, Semi-supervised medical image classification with relationd-riven self-ensembling model, *IEEE Trans. Med. Imag.*, vol. 39, no. 11, pp. 3429–3440, 2020.

[20] L. Yang, S. Yang, P. Jin, and R. Zhang, Semi-supervised hyperspectral image classification using spatio-spectral Laplacian support vector machine, *IEEE Geosc. Remote Sens. Lett.*, vol. 11, no. 3, pp. 651–655, 2014.

[21] Y. Wu, G. Mu, C. Qin, Q. Miao, W. Ma, and X. Zhang, Semi-supervised hyperspectral image classification via spatial-regulated self-training, *Remote Sensing*, vol. 12, no. 1, p. 159, 2020.
[22] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson, Offline/realtime traffic classification using semi-supervised learning, *Performance Evaluation*, vol. 64, no. 9-12, pp. 1194–1213, 2007.
[23] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha, Deep learning for classical Japanese literature, arXiv preprint arXiv: 1812.01718, 2018.
[24] J. Bergstra and Y. Bengio, Random search for hyper-parameter optimization, *J Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012.
[25] M. Sajjadi, M. Javanmardi, and T. Tasdizen, Regularization with stochastic transformations and perturbations for deep semi-supervised learning, in *Proc. 30th Int. Conf. Neural Information Processing Systems*, Barcelona, Spain, 2016, pp. 1171–1179.
[26] P. Bachman, O. Alsharif, and D. Precup, Learning with pseudo-ensembles, in *Proc. 27th Int. Conf. Neural Information Processing Systems*, Montreal, Canada, 2014, pp. 3365–3373.
[27] S. Park, J. Park, S. J. Shin, and I. C. Moon, Adversarial dropout for supervised and semi-supervised learning, in *Proc. 32nd AAAI Conf. Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conf. and Eighth AAAI Symp. Educational Advances in Artificial Intelligence*, New Orleans, LA, USA, 2018, pp. 480.
[28] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel, ReMixMatch: Semi-supervised learning with distribution alignment and augmentation anchoring, arXiv preprint arXiv: 1911.09785, 2020.
[29] T. Joachims, Transductive learning via spectral graph partitioning, in *Proc. Twentieth Int. Conf. Int. Conf. Machine Learning*, Washington, DC, USA, 2003, pp. 290–297.
[30] T. Joachims, Transductive inference for text classification using support vector machines, in *Proc. 16th Int. Conf. Machine Learning*, San Francisco, CA, USA, 1999, pp. 200–209.
[31] B. Yoshua, D. Olivier, and R. N. Le, Label propagation and quadratic criterion, in *Semi-Supervised Learning*, O. Chapelle, B. Scholkopf, A. Zien, eds. Cambridge, MA, USA: MIT Press, 2006, pp. 192–216.
[32] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, Semi-supervised learning with deep generative models, in *Proc. 27th Int. Conf. Neural Information Processing Systems*, Montreal, Canada, 2014, pp. 3581–3589.
[33] A. Odena, Semi-supervised learning with generative adversarial networks, arXiv preprint arXiv: 1606.01583, 2016.
[34] D. H Lee, Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks, in *Proc. 30th Int. Conf. Machine Learning*, Atlanta, GA, USA, 2013, p. 2.
[35] Y. Grandvalet and Y. Bengio, Semi-supervised learning by entropy minimization, in *Proc. 17th Int. Conf. Neural Information Processing Systems*, Vancouver, British, 2005, pp. 529–536.
[36] G. He, Y. Pan, X. Xia, J. He, R. Peng, and N. N. Xiong, A fast semi-supervised clustering framework for large-scale time series data, *IEEE Trans. Syst. Man Cybernet. Syst.*, vol. 51, no. 7, pp. 4201–4206, 2019.
[37] G. He, B. Li, H. Wang, and W. Jiang, Cost-effective active semi-supervised learning on multivariate time series data with crowds, *IEEE Trans. Syst. Man Cybernet. Syst.*, vol. 52, no. 3, pp. 1437–1450, 2020.
[38] G. Wang, K. W. Wong, and J. Lu, AUC-based extreme learning machines for supervised and semi-supervised imbalanced classification, *IEEE Trans. Syst. Man Cybernet. Syst.*, vol. 51, no. 12, pp. 7919–7930, 2020.
[39] J. Zhao, L. Chen, W. Pedrycz, and W. Wang, A novel semi-supervised sparse Bayesian regression based on variational inference for industrial datasets with incomplete outputs, *IEEE Trans. Syst. Man. Cybernet. Syst.*, vol. 50, no. 11, pp. 4773–4786, 2020.
[40] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, Building machines that learn and think like people, *Behav. Brain Sci.*, vol. 40, p. e253, 2017.
[41] M. Andrychowicz, M. Denil, S. G. Colmenarejo, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, Learning to learn by gradient descent by gradient descent, in *Proc. 30th Int. Conf. Neural Information Processing Systems*, Barcelona, Spain, 2016, pp. 3981–3989.
[42] S. Ravi and H. Larochelle, Optimization as a model for few-shot learning, in *5th Int. Conf. Learning Representations*, Toulon, France, 2017, pp. 1–11.
[43] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel, Meta-learning for semi-supervised few-shot classification, in *Proc. 6th Int. Conf. Learning Representations*, Vancouver, Canada, 2018.
[44] C. Finn, P. Abbeel, and S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in *Proc. 34th Int. Conf. Machine Learning*, Sydney, Australia, 2017, pp. 1126–1135.
[45] M. Ren, W. Zeng, B. Yang, and R. Urtasun, Learning to reweight examples for robust deep learning, in *Proc. 35th Int. Conf. Machine Learning*, Stockholmsmässan, Sweden, 2018, pp. 4334–4343.
[46] F. F. Li, R. Fergus, and P, Perona, One-shot learning of object categories, *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, 2006.
[47] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S. Hwang, and Y. Yang, Learning to propagate labels: Transductive propagation network for few-shot learning, in *Proc. 7th Int. Conf. Learning Representations*, New Orleans, LA, USA, 2019.
[48] X. Li, Q. Sun, Y. Liu, S. Zheng, Q. Zhou, T. S. Chua, and B. Schiele, Learning to self-train for semi-supervised few-shot classification, in *Proc. 33rd Int. Conf. Neural Information Processing Systems*, Vancouver, Canada, 2019, p. 922.
[49] Z. Yu, L. Chen, Z. Cheng, and J. Luo, TransMatch: A transfer-learning scheme for semi-supervised few-shot learning, in *2020 IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), Seattle, WA, USA, 2020, pp. 12853–12861.
[50] P. Rodríguez, I. Laradji, A. Drouin, and A. Lacoste, Embedding propagation: Smoother manifold for few-shot classification, in *16th European Conf. Computer Vision*, Glasgow, UK, 2020, pp. 121–138.
[51] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell, Meta-learning with latent embedding optimization, in *Proc. 7th Int. Conf. Learning Representations*, New Orleans, LA, USA, 2019.
[52] Q. Sun, Y. Liu, T. S. Chua, and B. Schiele, Meta-transfer learning for few-shot learning, in *2019 IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), Long Beach, USA, 2019, pp. 403–412.
[53] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. Mixup: Beyond empirical risk minimization. in *Proc. 6th Int. Conf. Learning Representations*, Vancouver, Canada, 2018.
[54] S. J. Reddi, A. Hefny, S. Sra, B. Póczós, and A. Smola, Stochastic variance reduction for nonconvex optimization, in *Proc. 33rd Int. Conf. Int. Conf. Machine Learning*, New York, NY, USA, 2016, pp. 314–323.
[55] A. Krizhevsky, *Learning Multiple Layers of Features from Tiny Images*,https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf, 2009.
[56] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, Reading digits in natural images with unsupervised feature learning, presented on 25th Conf. Neural Information Processing Systems Workshop on Deep Learning and Unsupervised Feature Learning, Granada, Spain, 2011.
[57] A. Coates, A. Ng, and H. Lee, An analysis of single-layer networks in unsupervised feature learning, in *Proc. Fourteenth Int. Conf. Artificial Intelligence and Statistics*, Fort Lauderdale, FL, USA, 2011, pp. 215–223.
[58] B. Athiwaratkun, M. Finzi, P. Izmailov, and A. G. Wilson, There are many consistent explanations of unlabeled data: Why you should average, in *Proc. 7th Int. Conf. Learning Representations*, New

Orleans, LA, USA, 2019.

[59] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, A simple framework for contrastive learning of visual representations, in *Proc. 37th Int. Conf. Machine Learning*, virtual, 2020, p. 149.

[60] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, Momentum contrast for unsupervised visual representation learning, in *2020 IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), Seattle, WA, USA, 2020, pp. 9726–9735.

[61] X. Chen and K. He, Exploring simple Siamese representation learning, in *2021 IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), Nashville, TN, USA, 2021, pp. 15745–15753.

[62] Y. Luo, J. Zhu, M. Li, Y. Ren, and B. Zhang, Smooth neighbors on teacher graphs for semi-supervised learning, in *2018 IEEE Conf. Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 8896–8905.

[63] I. Loshchilov and F. Hutter, SGDR: Stochastic gradient descent with warm restarts, in *Proc. 5th Int. Conf. Learning Representations*, Toulon, France, 2016.

[64] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, Snapshot ensembles: Train 1, get m for free, in *Proc. 5th Int. Conf. Learning Representations*, Toulon, France, 2017.

[65] J. Zhao, M. Mathieu, R. Goroshin, and Y. LeCun, Stacked what-where auto-encoders, arXiv preprint arXiv: 1506.02351, 2015.

[66] E. Denton, S. Gross, and R. Fergus, Semi-supervised learning with context-conditional generative adversarial networks, arXiv preprint arXiv: 1611.06430, 2016.

[67] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, Meta-learning with differentiable convex optimization, in *2019 IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), Long Beach, CA, USA, 2019, pp. 10649–10657.

[68] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, Unsupervised feature learning via non-parametric instance discrimination, in *2018 IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 3733–3742.

[69] W. Huang, M. Yi, and X. Zhao, Towards the generalization of contrastive self-supervised learning, arXiv preprint arXiv: 2111.00743, 2021.

[70] J. Li, C. Xiong, and S. C. H. Hoi, CoMatch: Semi-supervised learning with contrastive graph regularization, in *2021 IEEE/CVF Int. Conf. Computer Vision* (*ICCV*), Montreal, Canada, 2021, pp. 9455–9464.

[71] D. P. Bertsekas, *Nonlinear Programming* 2nd ed., Belmont, WY, USA: Athena Scientific, 1999.