Not All Parameters Should Be Treated Equally: Deep Safe Semi-supervised Learning under Class Distribution Mismatch

Rundong He[1], **Zhongyi Han**[1*], **Yang Yang**[2], **Yilong Yin**[1*]
[1]Shandong University
[2]Nanjing University of Science and Technology
rundong_he@mail.sdu.edu.cn, hanzhongyicn@gmail.com, yyang@njust.edu.cn, ylyin@sdu.edu.cn

Abstract

Deep semi-supervised learning (SSL) aims to utilize a sizeable unlabeled set to train deep networks, thereby reducing the dependence on labeled instances. However, the unlabeled set often carries unseen classes that cause the deep SSL algorithm to lose generalization. Previous works focus on the data level that they attempt to remove unseen class data or assign lower weight to them but could not eliminate their adverse effects on the SSL algorithm. Rather than focusing on the data level, this paper turns attention to the model parameter level. We find that only partial parameters are essential for seen-class classification, termed safe parameters. In contrast, the other parameters tend to fit irrelevant data, termed harmful parameters. Driven by this insight, we propose Safe Parameter Learning (**SPL**) to discover safe parameters and make the harmful parameters inactive, such that we can mitigate the adverse effects caused by unseen-class data. Specifically, we firstly design an effective strategy to divide all parameters in the pre-trained SSL model into safe and harmful ones. Then, we introduce a bi-level optimization strategy to update the safe parameters and kill the harmful parameters. Extensive experiments show that SPL outperforms the state-of-the-art SSL methods on all the benchmarks by a large margin. Moreover, experiments demonstrate that SPL can be integrated into the most popular deep SSL networks and be easily extended to handle other cases of class distribution mismatch.

Introduction

Deep semi-supervised learning (SSL) has made breakthroughs in many applications, such as medical image analysis (Han et al. 2021b; Ren, Yeh, and Schwing 2020) , video object segmentation (Lu et al. 2020b, 2021), and object tracking (Lu et al. 2020a; Shen et al. 2021). The remarkable success of deep SSL methods attributes to a large amount of cheap unlabeled data and a static environment where we draw labeled data and unlabeled data from an identical data distribution. The research of deep SSL methods have grown into a big tree with three main branches: consistency regularization methods (Sajjadi, Javanmardi, and Tasdizen 2016; Laine and Aila 2017; Tarvainen and Valpola 2017), pseudo-labeling methods (Rizve et al. 2021; Xie et al. 2020b; Cascante-Bonilla et al. 2021) and some hybrid methods (Berthelot et al. 2019, 2020; Sohn et al. 2020).

Once the learning environment changes, like unseen-class instances emerging in the unlabeled data, previous deep SSL methods often suffer severe performance degradation due to error propagation introduced by unseen-class unlabeled data (Oliver et al. 2018; Chen et al. 2020c; Guo et al. 2020). For example, at the beginning stage of the outbreak of COVID-19, the unlabeled data inevitably contains some imperceptible COVID-19 instances in the deep SSL pneumonia classification (Han et al. 2020b, 2021a). These unseen-class instances result in the loss of safety of the pneumonia classification model. We define this case as the problem of ***Safe Deep semi-supervised learning with Unseen-class unlabeled data*** (**SDU**), which accommodates a variety of real-world applications but is rarely considered in the literature.

To solve the SDU problem defined above, several safe deep SSL methods are proposed. Since unseen-class instances contained in unlabeled data can hurt the performance of seen-class classification, these methods focus on the data level that attempt to remove unseen-class data (Chen et al. 2020c; Yu et al. 2020) or assign lower weight to unseen-class data (Guo et al. 2020). Although they have weakened the negative influence brought by unseen-class instances, the performance is restricted because they could not thoroughly eliminate the adverse effects of unseen-class data. Some hard unseen-class unlabeled instances inevitably participate in the model training, which causes partial parameters fitting to these hard unseen-class data.

Inspired by recent works (Zhang et al. 2021) and (Xia et al. 2020), we present a novel insight: *rather than focusing on data level; it is better to discover safe parameters*. We term the parameters fitting to unseen-class data as harmful parameters and the other as safe parameters. Driven by this insight, we propose Safe Parameter Learning (SPL) to reduce the adverse effects caused by unseen-class unlabeled data in the learning process and thus enhance safe parameters for seen-class classification. SPL first identifies the safe parameters in the pre-trained SSL model by exploiting the magnitude of parameter weights and the class distribution mismatch degree. In this way, SPL divides all parameters into two parts: safe parameters and harmful parameters. SPL then suggests a bi-level optimization strategy, where inner-level optimization aims to enhance the reliability of safe pa-

*Co-corresponding author

rameters by swapping the state of partial safe and harmful parameters. In contrast, outer-level optimization aims to improve the performance of seen-class classification by further updating the safe parameters and making the harmful parameters inactive.

Our contributions can be summarized as follows:

- To our knowledge, it is the first time to solve the SDU problem from the perspective of parameter level.
- We propose a novel and effective strategy to categorize the safe and harmful parameters.
- We propose a novel bi-level optimization strategy to optimize safe parameters and make harmful parameters inactive.
- Experimental results on several representative datasets show that SPL achieves remarkable improvements compared with the state-of-the-art. Moreover, extensive studies also verify the universality of SPL.

Related Work

Deep Semi-Supervised Learning. Deep SSL has made remarkable progress in various machine learning problems. This remarkable achievement is mainly owed to the exploitation of abundant unlabeled data. Deep SSL methods mainly contain three categories: consistency regularization methods, pseudo-labeling methods, and hybrid methods. Consistency regularization methods take advantage of unlabeled data by forcing the outputs of the original unlabeled data and the perturbed unlabeled data to be similar (Sajjadi, Javanmardi, and Tasdizen 2016; Laine and Aila 2017; Tarvainen and Valpola 2017). Pseudo-labeling methods leverage the model itself to obtain pseudo labels for unlabeled data (Lee et al. 2013; Xie et al. 2020b; Cascante-Bonilla et al. 2021; Pham et al. 2020; Rizve et al. 2021). Hybrid methods (Berthelot et al. 2019, 2020; Sohn et al. 2020) simultaneously combine consistency regularization, pseudo-labeling, and data augmentation (Xie et al. 2020a; Cubuk et al. 2019; Devries and Taylor 2017). However, the success of these methods is based on the assumption that all the labeled and unlabeled data are derived from the same distribution. Once this assumption is not satisfied, the performance of these SSL methods degrades, even below the performance of supervised learning methods trained with only labeled data (Oliver et al. 2018).

Safe Semi-Supervised Learning. Safe SSL ensures the performance of SSL methods is no worse than a simple supervised learning model. The safe SSL problem under study mainly consists of three situations: data quality (Zhou et al. 2003; Han et al. 2020a; Guo et al. 2020), model uncertainty (Li and Zhou 2015), and measure diversity (Li and Liang 2019). This paper focuses on the data quality that the unseen-class instances emerge in unlabeled data. To mitigate the performance degradation of seen-class classification brought by unseen-class instances in unlabeled data, several deep safe SSL methods are proposed. Guo et al. (2020) proposes to assign soft weights to each unlabeled instance by a weighting function. Chen et al. (2020c) uses the model to identify unseen-class instances at the beginning of the training time but which is unstable. Yu et al. (2020) identifies unseen-class by considering labeled data as in-distribution data and unlabeled data as out-of-distribution data. However, the performance of unseen-class identification is limited by the small number of in-distribution data and noisy out-of-distribution data. Cascante-Bonilla et al. (2021) uses labeled data to train a supervised model and then identify unseen classes based on the model confidence. However, as the training process progresses, all unlabeled instances are involved in the training set. Although these methods have weakened the adverse effects brought by unseen-class instances, the improvement of performance is limited due to inaccurate unseen-class identification. Some unseen-class unlabeled data are selected to participate in the model training, which leads to some model parameters inevitably fitting to the irrelevant data. Different from these methods that focus on the data level, this paper focuses on the parameter level and proposes a novel and effective strategy to learn the safe parameters and make the harmful parameter inactive.

Lottery Tickets Hypothesis. The lottery ticket hypothesis (LTH) was originally proposed in (Frankle and Carbin 2018) which advocates the existence of an independently trainable sparse sub-network from a dense network. LTH has been explored widely in numerous contexts, such as image classification (Ma et al. 2021; Chen et al. 2021a), natural language processing (Gale, Elsen, and Hooker 2019; Chen et al. 2020a), reinforcement learning (Yu et al. 2019), generative adversarial networks (Chen et al. 2021c), graph neural networks (Chen et al. 2021b), adversarial robustness (Cosentino et al. 2019), lifelong learning (Chen et al. 2020b), out-of-distribution generalization (Zhang et al. 2021), and so on. Different from them, this paper focuses on the safe SSL problem and propose a new safe parameter learning strategy to improve seen-class classification.

Safe Parameters Learning

In this section, we first introduce the necessary notations. Then, we propose Safe Parameter Learning (SPL) to solve the SDU problem by answering three vital issues: How to identify the safe/harmful parameters? How to determine the ratio of safe parameters? Moreover, how to enhance the reliability of safe parameters? The overview of SPL can be seen in Fig. 1.

Learning Set-Up

Definition 1 *Distribution for Safe Semi-supervised Scenario. Given a feature space $\mathcal{X} \subset \mathbb{R}^d$ and the label space $\mathcal{Y}$, the labeled and unlabeled data have different joint distributions $P(X^l, Y^l)$ and $P(X^u, Y^u)$, where the feature space $X^l, X^u \subset \mathcal{X}$ and the label space $Y^l, Y^u \subset \mathcal{Y}$.*

Definition 2 *Safe Deep semi-supervised learning with Unseen-class unlabeled data (SDU). Let $D_L = \{(\boldsymbol{x}_i^l, y_i^l)\}_{i=1}^m$ denote the labeled data set, where m is the number of labeled data, $\boldsymbol{x}_i^l \in X^l$, $y_i^l \in Y^l$. Let $D_U = \{\boldsymbol{x}_i^u\}_{i=1}^n$ denote the unlabeled data set, where n is the number of unlabeled data, $\boldsymbol{x}_i^u \in X^u$, and $m \ll n$.*

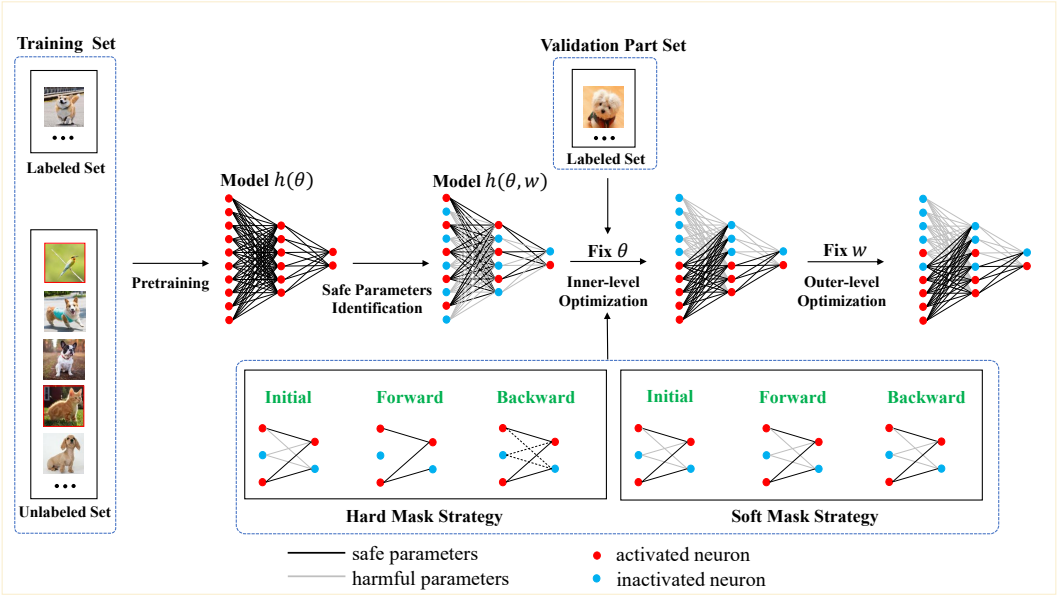


Figure 1: The overview of SPL for deep safe SSL with class distribution mismatch. Especially, SPL uses a novel and effective strategy to categorize the safe and harmful parameters, and uses a novel bi-level optimization strategy to optimize safe parameters and make harmful parameters inactive.

$Y^l \subset Y^u$ and $Y^{new} = Y^u \backslash Y^l$, where Y^{new} denotes unseen classes that only emerge in the unlabeled set D_U. Let $K = |Y^l|$ denote the number of seen classes.

Identifying Safe Parameters

The security of parameters has an active correlation with the magnitude of parameters in the pre-trained model (Han et al. 2015). Therefore, we classify safe and harmful ones based on the magnitude of the parameters. The process of categorizing parameters is to apply to mask ω on the parameter θ. The value of ω is '0' or '1'. '0' means that the parameter is harmful. '1' means that the parameter is essential for seen-class classification. Assuming that the ratio of safe parameters p is known, this process can be described formally by

$$\omega \odot \theta_0 \Leftarrow \theta_0\,, \quad s.t. \quad \frac{\|\omega\|_0}{k} = p\,, \tag{1}$$

where $\|\cdot\|_0$ means the standard ℓ_0-norm, $\odot$ denotes the element-wise multiplication, k is the number of parameters and θ_0 is the pre-trained model parameters. By Eq. (1), the process of identifying safe parameters is based on a pre-trained model. To obtain the pre-trained model, we can use common SSL methods such as VAT (Miyato et al. 2019), Pi-Model (Sajjadi, Javanmardi, and Tasdizen 2016), and so on. The objective function of the pre-training stage is as follows,

$$\theta_0 = \min_{\theta} \mathcal{L}(\theta; D_L, D_U)\,, \tag{2}$$

where θ_0 denotes the weights of the pre-trained model and $\mathcal{L}$ denotes the loss function.

However, we assume that the ratio of safe parameters p is known when identifying safe parameters. In fact, we need to estimate p in advance. To achieve that, we design an effective estimation strategy in the next subsection.

Estimating the Ratio of Safe Parameters

We have presented how to judge the safety of parameters and then divide them into safe and harmful ones. However, how to obtain the ratio of safe parameters is also a critical issue. We exploit the class distribution mismatch proportion in unlabeled data and size proportion degree between labeled and unlabeled data to help estimate the ratio of safe parameters. Intuitively, if the class distribution mismatch proportion is high, the number of unseen-class instances is large. The number of harmful parameters for memorizing unseen-class instances is then large. Therefore, the number of harmful parameters has a positive correlation with the class distribution mismatch proportion. Similarly, if the size proportion degree between labeled and unlabeled data is low, the number of unseen-class instances is large. The number of harmful parameters for memorizing unseen-class instances is then large. The number of harmful parameters has a negative correlation with the ratio of the labeled set. In summary, we combine the class distribution mismatch proportion and the ratio of the labeled set to help determine the ratio of safe parameters. Let δ denote the class distribution mismatch proportion and let τ denote the ratio of the labeled set. If δ is not known in advance, we can consider the unseen-class instances in the unlabeled set as open set noises and then easily infer it by (Liu and Tao 2015). τ can be obtained by calculating m/n, where m denotes the number of instances in labeled set D_L, and n denotes the number of instances in unlabeled set D_U. Then, the ratio p of safe parameters is:

$$p = 1 - \delta \cdot (1 - \tau)\,. \tag{3}$$

According to the estimated ratio p of safe parameters and magnitude-based safe parameter identification criterion, we can divide the parameters into safe ones whose masks are set '1' and harmful ones whose masks are set '0'.

Last but not least, there is still an important problem that cannot be ignored: the masks obtained by magnitude-based safe parameter identification criterion are not reliable. It is not stable to identify safe parameters simply by the corresponding magnitude. Because the security of parameters cannot be measured by magnitude alone, which is also affected by other factors, such as gradient flow (Wang, Zhang, and Grosse 2020), input data (Zhang et al. 2021), and so on. To enhance the safe parameters, we propose an effective bi-level optimization strategy.

Bi-level Optimization

The objective of the bi-level optimization strategy is to optimize the parameters of the network and the masks of the network parameters. Formally, we define the bi-level optimization by

$$\begin{aligned} &\min_{\theta} \mathcal{L}^{outer}\left(\omega^* \odot \theta; D_L, D_U\right) , \\ &s.t. \quad \omega^* = \arg\min_{\omega} \mathcal{L}^{inner}\left(\omega \odot \theta; V_1\right) , \end{aligned} \tag{4}$$

where θ denotes the parameters of network, ω denotes the mask of parameters, $\mathcal{L}^{outer}$ denotes the loss function of outer-level optimization, $\mathcal{L}^{inner}$ denotes the loss function of inner-level optimization, V_1 denotes a part of the validation set V, $|V_1|/|V| = 0.7$, the remaining validation set $V_2 = V \backslash V_1$ are used to select optimal model for testing. Eq. (4) can be divided into two stages: first, inner-level optimization seeks the optimal mask ω^* by minimizing $\mathcal{L}^{inner}$, then outer-level optimization aims to further enhance seen-class classification by minimizing $\mathcal{L}^{outer}$. $\mathcal{L}^{inner}$ is

$$\mathcal{L}^{inner} = \sum_{(\boldsymbol{x}_i, y_i) \in V_1} \ell(h(\boldsymbol{x}_i; \omega, \theta), y_i) , \tag{5}$$

where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathcal{R}$ refers to certain loss function, e.g., mean squared error or cross entropy loss, and $h(\cdot)$ denotes the output of model. $\mathcal{L}^{outer}$ is defined as follows,

$$\mathcal{L}^{outer} = \sum_{\boldsymbol{x}_i^l \in D_L} \ell\left(h\left(\boldsymbol{x}_i^l; \omega, \theta\right), y_i^l\right) + \sum_{\boldsymbol{x}_i^u \in D_U} \Omega\left(\boldsymbol{x}_i^u; \omega, \theta\right) , \tag{6}$$

where $\Omega(\cdot)$ denotes the regularization term defined by

$$\Omega(\boldsymbol{x}; \omega, \theta) = \|h(\text{perturb}(\boldsymbol{x}); \omega, \theta) - h(\boldsymbol{x}; \omega, \theta)\|_2^2 , \tag{7}$$

where perturb$(\cdot)$ refers to certain stochastic perturbation.

Optimization Strategy

Eq. (4) is a bi-level optimization problem (Bard 2013), where one optimization problem is nested within another problem. The inner-level optimization is to enhance the reliability of safe parameter selection by swapping the state of the partial safe and harmful parameters given a part of the validation set, whereas the outer-level optimization is to improve the performance of seen-class classification by fine-tuning the weights of safe parameters given the learned mask, labeled data, and unlabeled data. More specifically, we adopt gradient descent methods to obtain the optimal ω^* approximately. And the training procedure can be written as:

$$\omega_{t+1} = \omega_t - \eta_\omega \nabla_\omega \mathcal{L}^{inner}\left(\omega_t \odot \theta; V_1\right) , \tag{8}$$

where η_ω is learning rate for ω, t indicates the t-th iteration. But it is worth mentioning that the initial mask $\omega \in \{0, 1\}^k$, which fails to back propagate due to discrete value.

To achieve gradient updating to ω, SPL offers two feasible strategies called *hard mask* and *soft mask*. First, SPL relaxes the mask ω from $\{0, 1\}$ to continuous values in the interval [0, 1], denoted by $\bar{\omega}$. Then, hard mask strategy tries to use the straight-through gradient estimator (Bengio, Léonard, and Courville 2013) to approximate $\nabla_\omega \mathcal{L}^{inner}(\cdot)$. During the forward propagation, we use the binarization function $h(\cdot)$ to $\bar{\omega}$ as follows,

$$\omega = h(\bar{\omega}) = \begin{cases} 1 , \text{ if } \bar{\omega} \text{ in the top- } p^l \text{ largest } , \\ 0 , \text{ otherwise } , \end{cases} \tag{9}$$

where p^l is the special ratio of safe parameters in l-th layer. When $h(\bar{\omega}) = 0$, the corresponding parameters are considered as harmful parameters and not involved in forward propagation, and vice versa. During the backpropagation, relaxed mask $\bar{\omega}$ is updated by Eq. (8).

Unlike hard mask strategy using binarization function operation, soft mask strategy tries to use $\bar{\omega}$ with ReLU (Glorot, Bordes, and Bengio 2011), defined by

$$\omega = \text{ReLU}(\bar{\omega}) = \begin{cases} \bar{\omega} , \text{ if } \bar{\omega} > 0 , \\ 0 , \text{ otherwise } . \end{cases} \tag{10}$$

By comparing Eq. (9) and Eq. (10), we can know that ω in hard mask strategy and soft mask strategy during the forward propagation is discrete value and continuous value, respectively. We can also understand the two strategies in this way, where hard mask strategy can be seen as selecting safe parameters, and soft mask strategy can be seen as weighting the parameters according to parameters' security. The specific process of the two strategies can be seen in Fig. 1.

These two strategies have their advantages and deficiencies. Regarding the hard mask strategy, it strictly controls the number of safe parameters and achieves the performance improvement of seen-class classification with a smaller number of parameters, while the performance improvement is not as good as the soft mask strategy. Although the performance of the soft mask strategy is superior to the hard mask strategy, the soft mask strategy cannot strictly control the ratio of safe parameters, and all harmful parameters are still being optimized continuously. We adopt the hard mask strategy to optimize the mask in our experiments.

By continuously utilizing Eq. (8), we can obtain the optimal ω^* approximately. After that, we compute $\mathcal{L}^{outer}$ and then update the parameters θ:

$$\theta_{t+1} = \theta_t - \eta_\theta \nabla_\theta \mathcal{L}^{outer}\left(\omega^* \odot \theta; D_L, D_U\right) , \tag{11}$$

where η_θ is the learning rate for θ.

The whole bi-level optimization process needs $2 \times T$ round iterations. T is very small compared to the number of iterations required for the pre-training process. So, the time spent on the bi-level optimization is very little. The overall algorithm is summarized in Algorithm 1.

Algorithm 1: Safe Parameter Learning (SPL).

input : The parameters θ of model, the mask ω of parameters, the ratio p of safe parameters, max iterations T, max epochs E, T $\ll$ E
output: θ, ω

/*Pre-training:*/
for e = 1 **to** E **do**
  compute loss $\mathcal{L}$ based on baseline by θ
  update $\theta \leftarrow$ SGD with loss $\mathcal{L}$
obtain the parameters θ_0 of model after pre-train
/*Safe Parameters Identification:*/
compute threshold based on the ratio p
obtain initial mask ω_0 according to the threshold
$\omega \leftarrow \omega_0$
/*Bi-level Optimization:*/
while not converged **do**
  for $t = 1$ **to** T **do**
    compute loss: $\mathcal{L}^{inner}\left(\omega \odot \theta; V_1\right)$
    update $\omega \leftarrow$ SGD with loss $\mathcal{L}^{inner}$
  $\omega^* \leftarrow \omega$
  for $t = 1$ **to** T **do**
    compute loss: $\mathcal{L}^{outer}\left(\omega^* \odot \theta; D_L, D_U\right)$
    update $\theta \leftarrow$ SGD with loss $\mathcal{L}^{outer}$
end while

Convergence

We show that the update of mask in the inner-level optimization algorithm leads to the convergence of $\mathcal{L}^{inner}(\cdot)$. Once the overall ratio of safe parameters p is determined, the ratio of safe parameters in each layer is also fixed. Let p^l denote the ratio of safe parameters in the l-th layer. Then, we fix the parameters of the network but optimize the corresponding masks of parameters.

Suppose $\mathcal{L}^{inner}(\cdot)$ is Lipschitz-smooth. $\bar{\omega}(u, v)$ is the related mask of parameter $\theta(u, v)$ which connect node u and v, where node u is in $(l-1)$-th layer and node v is in l-th layer. Let $\mathcal{I}_v$ denote the input to node v and $\mathcal{O}_v$ denote the output of node v. Suppose the state of parameter $\theta(u, v)$ in t-th iteration is harmful parameter and $\theta(j, v)$ in t-th iteration is safe parameter. Let ${\mathcal{I}_v}^{t+1}$ denote the input to node v at $(t+1)$-th iteration and ${\mathcal{I}_v}^{t}$ denote the input to node v at t-th iteration.

Theorem 1 *(Convergence.) When parameter $\theta(u, v)$ replaces parameter $\theta(j, v)$ as safe parameter in $(t+1)$-th iteration and the rest of the parameters remains fixed, $\mathcal{L}^{inner}(\cdot)$ is convergent, i.e.,*

$$\mathcal{L}^{inner}({\mathcal{I}_v}^{t+1}) < \mathcal{L}^{inner}({\mathcal{I}_v}^{t}) . \tag{12}$$

Furthermore, the equality in Eq. (12) *holds only when the replacement does not occur, i.e.,*

$$\mathcal{L}^{inner}({\mathcal{I}_v}^{t+1}) = \mathcal{L}^{inner}({\mathcal{I}_v}^{t}) , \tag{13}$$

if and only if

$${\mathcal{I}_v}^{t+1} = {\mathcal{I}_v}^{t} . \tag{14}$$

The specific proof process can be found in the Appendix.

Experiments

In this section, we analyze the effectiveness of SPL on standard SSL benchmarks using deep convolutional neural networks for SSL image classification.

Experimental Setup

Datasets. We evaluate SPL on image classification datasets: CIFAR-10 (Krizhevsky, Hinton et al. 2009), CIFAR-100 (Krizhevsky, Hinton et al. 2009) and TinyImageNet (a subset of ImageNet (Deng et al. 2009)), with different ratios of class mismatch. Detailed introductions to datasets can be found in the Appendix.

Baselines. We compare **SPL** on test data that only contain seen-class instances with SSL baselines: Pseudo-Labeling (**PL**) (Lee et al. 2013), Pi-Model (**PI**) (Sajjadi, Javanmardi, and Tasdizen 2016), Temporal Ensembling (**TE**) (Laine and Aila 2017), Mean Teacher (**MT**) (Tarvainen and Valpola 2017), Virtual Adversarial Training (**VAT**) (Miyato et al. 2019), **UASD** (Chen et al. 2020c), **DS^{3}L** (Guo et al. 2020), Multi-Task Curriculum (**MTC**) (Yu et al. 2020), and Curriculum Labeling (**CL**) (Cascante-Bonilla et al. 2021). Moreover, we let **supervised method** trained on D_L as another baseline.

Implementation Details For a comprehensive and fair comparison, our experiments are built upon (Oliver et al. 2018) with Pytorch. We use the standard Wide ResNet (Zagoruyko and Komodakis 2016), i.e., WRN-28-2, as the base network for training. More details of implementation are given in the Appendix.

CIFAR-10

Evaluation protocol. To simulate a more realistic SSL with class distribution mismatch, we construct the unlabeled data with unseen classes that are not in the labeled data. Following (Oliver et al. 2018), we perform experiments on CIFAR-10 for a six-class classification task, using 400 examples per class. The labeled set contains six classes of animals: bird, cat, deer, dog, frog, horse; while the unlabeled data comes from all ten classes, with a varying class distribution mismatch proportion from 0% to 60%. For instance, when the mismatch proportion is 50%, half of the unlabeled data comes from six classes of animals, and the others come from the remaining four classes. The test accuracy is reported on the six seen classes.

Comparison with baseline. Fig. 3(a) shows experimental results on CIFAR-10, including the supervised learning method, five common SSL methods, and our proposed SPL under varying class distribution mismatch proportion from 0% to 60%. It can be observed that when increasing the amount of unseen-class unlabeled data, these SSL methods degrade drastically. Especially when the class distribution mismatch proportion reaches 40%, these SSL methods are inferior to the supervised learning method, which contradicts the original purpose of SSL. However, SPL even outperforms the supervised learning method by 3.6% when the class distribution mismatch is 60%. These results demonstrate that our proposed SPL is very effective against the

Method	CIFAR-10		CIFAR-100		TinyImageNet	
	ratio=0.3	ratio=0.6	ratio=0.3	ratio=0.6	ratio=0.3	ratio=0.6
Supervised	76.3±0.4	76.3±0.4	58.6±0.5	58.6±0.5	36.5±0.5	36.5±0.5
PI	75.7±0.7	74.5±1.0	59.4±0.3	57.9±0.3	36.9±0.4	36.4±0.5
PL	75.8±0.8	74.6±0.7	60.2±0.3	57.5±0.6	36.6±0.6	35.8±0.4
VAT	76.9±0.6	75.0±0.5	63.3±0.4	61.6±0.6	36.7±0.5	36.3±0.6
DS^3L	78.1±0.4	76.9±0.5	-	-	-	-
UASD	77.6±0.4	76.0±0.4	61.8±0.4	58.4±0.5	37.1±0.7	36.9±0.6
MTC	85.5±0.6	81.7±0.5	63.1±0.6	61.1±0.3	37.0±0.5	36.6±0.4
CL	83.2±0.4	82.1±0.4	63.6±0.4	61.5±0.5	37.3±0.7	36.7±0.8
PI+SPL	78.6±0.5	77.2±0.4	60.6±0.2	59.8±0.4	37.8±0.3	37.1±0.6
PL+SPL	79.0±0.4	76.8±0.4	61.7±0.4	60.4±0.4	37.3±0.5	36.6±0.6
VAT+SPL	80.1±0.6	79.9±0.5	65.7±0.2	63.9±0.4	37.7±0.5	37.1±0.5
UASD+SPL	78.2±0.4	76.8±0.6	63.2±0.2	59.5±0.2	38.1±0.4	37.1±0.4
MTC+SPL	85.7±0.3	81.7±0.4	64.2±0.3	63.1±0.4	38.3±0.3	37.3±0.4
CL+SPL	**87.8±0.3**	**84.1±0.5**	**65.9±0.3**	**65.5±0.4**	**38.6±0.5**	**37.7±0.5**

Table 1: Accuracy (%) on the three datasets.

harms caused by class distribution mismatch. In more depth, we also compare SPL with deep safe SSL methods (such as DS^3L, UASD, MTC, CL). The results of these baseline methods on CIFAR-10 can be seen in the first column of Table 1. These baseline methods solve the SDU problem from sample selection. Different from these baseline methods, SPL is based on parameter selection. According to the first column of Table 1, we know that SPL based on CL achieves the accuracy of 87.8% and 84.1% at class distribution mismatch proportions of 30% and 60%, respectively, which improves by 2.3% and 2.0% compared to state of the art. These results show the effectiveness of SPL.

Evaluation on CIFAR-100 and TinyImageNet

Evaluation protocol. We conduct experiments on CIFAR-100 and TinyImageNet to evaluate SSL under larger class space. Following the similar setting to CIFAR-10, on CIFAR-100, we use the first 50 classes as labeled classes and the remaining 50 classes as unseen classes. And on TinyImageNet, we use the first 100 classes as labeled classes and the remaining 100 classes as unseen classes. We verify the accuracy when the class distribution mismatch proportion is 30% and 60%, respectively.

Evaluation results. The second column and third column of Table 1 reports the results on CIFAR-100 and TinyImageNet, respectively. SPL achieves the best results under different class distribution mismatch proportions. For example, on CIFAR-100, when SPL is incorporated into CL, and the class distribution mismatch proportion is 60%, SPL significantly outperforms CL by a large margin, with about 4% accuracy improvement.

In-depth Analysis

Universality analysis. As shown in Fig. 3(c) and Fig. 3(d), we incorporate SPL into two typical deep SSL methods (i.e., PI, PL) on CIFAR-10 and CIFAR-100 under different class mismatch ratios to demonstrate the universality of SPL. The results show that the two deep SSL methods

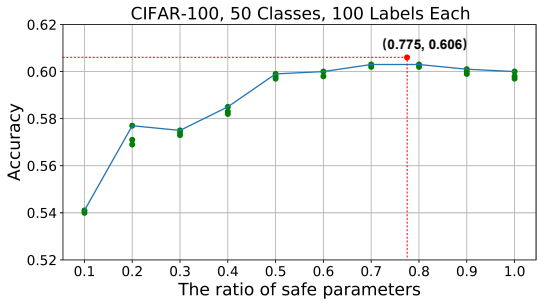
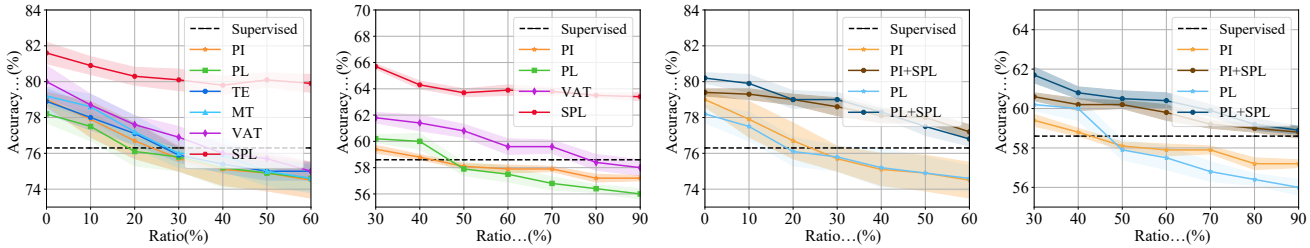


Figure 2: Classification accuracy of SPL based on PI on CIFAR-100 under different ratios of safe parameters with the class distribution mismatch proportion of 0.3.

can achieve performance improvement by combining SPL. To further demonstrate the universality of SPL, we also conduct experiments on CIFAR-10, CIFAR-100, and TinyImageNet by combining three deep SSL methods and three deep safe SSL methods with SPL. As shown in Table 1, SPL achieves state-of-the-art accuracies on all datasets. After incorporating into existing methods, our algorithm outperforms all the original methods, showing its efficacy and universality. For example, when SPL is built-in CL (CL+SPL) on CIFAR-10 under the 30% of class mismatch ratio, it significantly outperforms CL by a large margin, over 4.6% accuracy. The above results explicitly verify that safe parameter learning is an effective approach to solving the SDU problem.

The ratio of safe parameters. While the above results have demonstrated the strengths of our proposed SPL for reducing the adverse effects of unseen-class unlabeled data, we provide a broader spectrum for more in-depth analysis for the ratio p of safe parameters. Fig. 2 shows the results of PI-based SPL on CIFAR-100 under the wider ratios of safe parameters, in which the ratios of safe parameters are set from 0.1 to 1.0 or according to Eq. (3). We can see that



(a) CIFAR-10 (b) CIFAR-100 (c) CIFAR-10 (d) CIFAR-100

Figure 3: (a): Experiment results on CIFAR-10 under varying class distribution mismatch proportion (from 0% to 60%). (b): Experiment results on CIFAR-100 with different class distribution mismatch proportion (from 30% to 90%). The shaded area indicates the standard deviation over five runs. (c): Classification accuracy of PI, PI+SPL, PL, and PL+SPL on CIFAR-10 under varying class distribution mismatch proportion. (d): Classification accuracy of PI, PI+SPL, PL, and PL+SPL on CIFAR-100 with different class mismatch proportion.

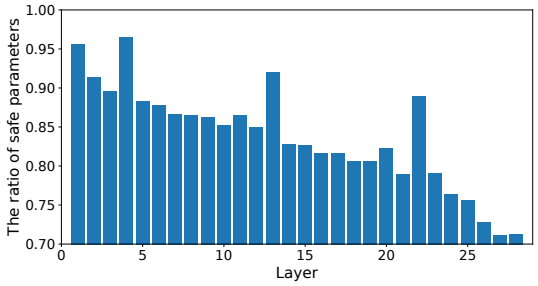


Figure 4: Each layer's ratio of safe parameters when the overall ratio of safe parameters is 0.3.

using the estimated ratio of safe parameters maintains the best accuracy. Fig. 2 also points that the accuracies with the different ratios of safe parameters are stable, indicating that the performance of our proposed SPL is insensitive to the ratio of safe parameters. Fig. 4 shows the specific ratio of safe parameters in the different layers under WRN-28-2 network with the estimated ratio of 0.775. From Fig. 4, we can find that the closer to the input layer, the higher the ratio of safe parameters, and the closer to the output layer, the lower the ratio of safe parameters. This phenomenon illustrates that the unseen-class unlabeled data have the more significant influence on high-level semantic feature learning.

Method	ratio=0.3	ratio=0.4	ratio=0.5	ratio=0.6
CL	63.6±0.4	63.2±0.2	62.4±0.3	61.5±0.5
CL+fine-tuning	62.1±0.5	62.0±0.2	61.1±0.2	61.0±0.5
CL+SPL (HMK)	65.9±0.3	65.1±0.3	65.0±0.3	65.5±0.4
CL+SPL (SMK)	66.9±0.4	67.3±0.3	66.3±0.2	66.7±0.2
Method	ratio=0.7	ratio=0.8	ratio=0.9	ratio=1.0
CL	61.5±0.4	59.3±0.2	58.9±0.5	57.9±0.6
CL+fine-tuning	59.2±0.3	57.6±0.1	56.8±0.5	56.4±0.4
CL+SPL (HMK)	63.6±0.3	62.5±0.3	60.9±0.2	60.7±0.4
CL+SPL (SMK)	65.8±0.3	63.8±0.4	63.6±0.3	63.5±0.5

Table 2: Accuracy (%) on CIFAR-100 with the class mismatch ratio from 30% to 100%.

Hard and soft mask. During the section of optimization strategy, we offer two optimization ways for ω called hard mask (HMK) and soft mask (SMK). Then, we compare these two ways on CIFAR-100, and the experimental results are shown in Table 2. Whatever the class distribution mismatch ratio is and whether to use hard mask or soft mask, the accuracy of SPL is higher than CL. The performance of soft mask is higher than hard mask, which demonstrates that weighting the parameters according to parameters' safety degree is better. Moreover, to verify that the performance improvement is due to SPL rather than introducing V_1, we fine-tune the CL-based pre-trained model directly using V_1. Results show that introducing V_1 directly for fine-tuning cannot lead to performance improvement of seen-class classification. SPL uses V_1 for safe parameter learning, which can be seen as adjusting the hyper-parameters using V_1, because we don't change the value of parameters in this step.

Ablation analysis. We validate the effectiveness of the components in SPL by ablating them and measuring the performance on CIFAR-10. Table 3 reports the results of ablation studies which contain SPL without pre-training (PT), SPL without identifying safe parameters (ISP), SPL without bi-level optimization (BLO), SPL without inner-level optimization (ILO), and SPL without outer-level optimization (OLO). We can see that all components have a significant effect as removing any of them causes a decline in performance.

Method	m=100×6	m=200×6	m=400×6
pre-training	59.7±1.0	71.2±0.7	76.9±0.6
SPL w/o PT	32.5±1.5	33.2±1.2	33.8±1.1
SPL w/o ISP	59.9±0.6	71.2±0.4	77.0±0.5
SPL w/o BLO	59.6±0.7	71.1±0.6	77.2±0.6
SPL w/o ILO	59.4±0.5	70.4±0.4	76.3±0.3
SPL w/o OLO	65.9±0.3	75.5±0.3	79.8±0.2
SPL	**66.1±0.4**	**75.7±0.2**	**80.1±0.2**

Table 3: Seen-class classification accuracy (%) of ablation studies on CIFAR-10 when the extent of labeled/unlabeled class mismatch ratio is 30% and pre-trained model is VAT.

Conclusion

We presented a new analysis of deep safe SSL with unseen-class unlabeled data, an under-explored but more realistic scenario. We also proposed a practical method called SPL guaranteed by solving three critical issues, that are how to identify the safe/harmful parameters, how to determine the ratio of safe parameters, and how to enhance the reliability of safe parameters. Empirical studies show that, unlike the compared deep SSL methods, SPL still achieves stable performance gain no matter the class distribution mismatch proportion and exceeds the existing deep safe SSL techniques by a large margin on different datasets. SPL can incorporate into the most typical deep SSL methods and deep safe SSL methods when emerging class distribution mismatch. Beyond this work, it is also worthwhile to build a unified theoretical analysis work for this new problem. One can also extend our work into other safe SSL environments.

The Proof of Theorem 1

Proof. Let $\bar{\omega}_{t+1}(u,v)$ denote the soft mask of parameter $\theta(u,v)$ after the gradient is updated, which is obtained as follows,

$$\bar{\omega}_{t+1}(u,v) = \bar{\omega}_t(u,v) - \eta_\omega \frac{\partial \mathcal{L}^{inner}}{\partial \mathcal{I}_v} \theta(u,v)\mathcal{O}_u \,. \quad (15)$$

Because parameter $\theta(u,v)$ replaces parameter $\theta(j,v)$ as safe parameter, we can know $\bar{\omega}_t(u,v) < \bar{\omega}_t(j,v)$ but $\bar{\omega}_{t+1}(u,v) > \bar{\omega}_{t+1}(j,v)$. Accordingly,

$$\bar{\omega}_{t+1}(u,v) - \bar{\omega}_t(u,v) > \bar{\omega}_{t+1}(j,v) - \bar{\omega}_t(j,v) \quad (16)$$

which implies that

$$-\eta_\omega \frac{\partial \mathcal{L}^{inner}}{\partial \mathcal{I}_v} \theta(u,v)\mathcal{O}_u > -\eta_\omega \frac{\partial \mathcal{L}^{inner}}{\partial \mathcal{I}_v} \theta(j,v)\mathcal{O}_j \quad (17)$$

Let ${\mathcal{I}_v}^{t+1}$ denote the input to node v at $(t+1)$-th iteration and let ${\mathcal{I}_v}^{t}$ denote the input to node v at t-th iteration. Note that ${\mathcal{I}_v}^{t+1} - {\mathcal{I}_v}^{t} = \theta(u,v)\mathcal{O}_u - \theta(j,v)\mathcal{O}_j$.

Now, we need to verify $\mathcal{L}^{inner}({\mathcal{I}_v}^{t+1}) < \mathcal{L}^{inner}({\mathcal{I}_v}^{t})$. Owing to $\mathcal{L}^{inner}(\cdot)$ is Lipschitz-smooth, by Taylor expansion, we obtain the approximation of $\mathcal{L}^{inner}({\mathcal{I}_v}^{t+1})$ as follows,

$$\begin{aligned}
&\mathcal{L}^{inner}\left({\mathcal{I}_v}^{t+1}\right) \\
&= \mathcal{L}^{inner}\left({\mathcal{I}_v}^{t} + \left({\mathcal{I}_v}^{t+1} - {\mathcal{I}_v}^{t}\right)\right) \\
&\approx \mathcal{L}^{inner}\left({\mathcal{I}_v}^{t}\right) + \frac{\partial \mathcal{L}^{inner}}{\partial \mathcal{I}_v}\left({\mathcal{I}_v}^{t+1} - {\mathcal{I}_v}^{t}\right) \\
&= \mathcal{L}^{inner}\left({\mathcal{I}_v}^{t}\right) + \frac{\partial \mathcal{L}^{inner}}{\partial \mathcal{I}_v}\left(\theta(u,v)\mathcal{O}_u - \theta(j,v)\mathcal{O}_j\right)
\end{aligned} \quad (18)$$

By Eq. (17), we know that the second item of Eq. (18) less than zero. Accordingly, $\mathcal{L}^{inner}({\mathcal{I}_v}^{t+1}) < \mathcal{L}^{inner}({\mathcal{I}_v}^{t})$.

Experiments

Datasets

We evaluate SPL on image classification datasets: CIFAR-10, CIFAR-100 and TinyImageNet, with different ratios of class mismatch.

- **CIFAR-10** includes 60,000 training images and 10,000 testing images of size 32×32 which contains ten categories: "airline", "automobile", "bird", "cat", "deer", "dog", "frog", "horse", "ship", and "trunk". Our experiment carries out six-class classification tasks. We consider animal categories (birds, cats, deer, dogs, frogs, and horses) as seen classes and the rest as unseen classes. We select 400 images from each seen category to construct the labeled data set D_L, i.e., 2400 labeled instances. Meanwhile, 20,000 images in total are randomly selected as the unlabeled data set D_U from all the ten categories. We adjust the ratio of unseen-class images in the unlabeled data to modulate class distribution mismatch.
- **CIFAR-100** includes 50,000 training images and 10,000 testing images of size 32×32 which contains 100 categories. We use the first half categories (1-50) as seen classes, and the remaining classes as unseen classes. We select 100 images from each seen category to construct the labeled data set D_L, i.e., 5000 labeled instances. Meanwhile, 20,000 images in total are randomly selected as the unlabeled data set D_U from all the 100 categories with different ratios of unseen classes.
- **TinyImageNet** contains 200 categories which includes 500 training images and 50 testing images in each category. We resize all images to 32×32. We use the first 100 categories as seen classes, and the remaining classes as unseen classes. We select 100 images from each seen category to construct the labeled data set D_L, i.e., 10000 labeled instances. Meanwhile, 40,000 images in total are randomly selected as the unlabeled data set D_U from all the 200 categories with different ratios of unseen classes.

Hyperparameters

parameter	value
max iteration T	300
initial learning rate	0.1
learning decay factor	0.1
learning decay at iteration	100,200

Table 4: Hyperparameter settings of our proposed SPL.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (62176139, 61876098), the Major Basic Research Project of Natural Science Foundation of Shandong Province (ZR2021ZD15), the Natural Science Foundation of Jiangsu Province of China under Grant (BK20200460), the CAAI-Huawei MindSpore Open Fund (CAAIXSJLJJ-2021-014B).

References

Bard, J. F. 2013. *Practical bilevel optimization: algorithms and applications*, volume 30. Springer Science & Business Media.

Bengio, Y.; Léonard, N.; and Courville, A. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Berthelot, D.; Carlini, N.; Cubuk, E. D.; Kurakin, A.; Sohn, K.; Zhang, H.; and Raffel, C. 2020. ReMixMatch: Semi-Supervised Learning with Distribution Matching and Augmentation Anchoring. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Berthelot, D.; Carlini, N.; Goodfellow, I. J.; Papernot, N.; Oliver, A.; and Raffel, C. 2019. MixMatch: A Holistic Approach to Semi-Supervised Learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 5050–5060.

Cascante-Bonilla, P.; Tan, F.; Qi, Y.; and Ordonez, V. 2021. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In *AAAI*.

Chen, T.; Frankle, J.; Chang, S.; Liu, S.; Zhang, Y.; Carbin, M.; and Wang, Z. 2021a. The lottery tickets hypothesis for supervised and self-supervised pre-training in computer vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16306–16316.

Chen, T.; Frankle, J.; Chang, S.; Liu, S.; Zhang, Y.; Wang, Z.; and Carbin, M. 2020a. The lottery ticket hypothesis for pre-trained bert networks. *arXiv preprint arXiv:2007.12223*.

Chen, T.; Sui, Y.; Chen, X.; Zhang, A.; and Wang, Z. 2021b. A unified lottery ticket hypothesis for graph neural networks. In *International Conference on Machine Learning*, 1695–1706. PMLR.

Chen, T.; Zhang, Z.; Liu, S.; Chang, S.; and Wang, Z. 2020b. Long live the lottery: The existence of winning tickets in lifelong learning. In *International Conference on Learning Representations*.

Chen, X.; Zhang, Z.; Sui, Y.; and Chen, T. 2021c. Gans can play lottery tickets too. *arXiv preprint arXiv:2106.00134*.

Chen, Y.; Zhu, X.; Li, W.; and Gong, S. 2020c. Semi-Supervised Learning under Class Distribution Mismatch. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 3569–3576.

Cosentino, J.; Zaiter, F.; Pei, D.; and Zhu, J. 2019. The search for sparse, robust neural networks. *arXiv preprint arXiv:1912.02386*.

Cubuk, E. D.; Zoph, B.; Mané, D.; Vasudevan, V.; and Le, Q. V. 2019. AutoAugment: Learning Augmentation Strategies From Data. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 113–123.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

Devries, T.; and Taylor, G. W. 2017. Improved Regularization of Convolutional Neural Networks with Cutout. *CoRR*, abs/1708.04552.

Frankle, J.; and Carbin, M. 2018. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*.

Gale, T.; Elsen, E.; and Hooker, S. 2019. The state of sparsity in deep neural networks. *arXiv preprint arXiv:1902.09574*.

Glorot, X.; Bordes, A.; and Bengio, Y. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 315–323. JMLR Workshop and Conference Proceedings.

Guo, L.; Zhang, Z.; Jiang, Y.; Li, Y.; and Zhou, Z. 2020. Safe Deep Semi-Supervised Learning for Unseen-Class Unlabeled Data. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119, 3897–3906.

Han, S.; Pool, J.; Tran, J.; and Dally, W. J. 2015. Learning both weights and connections for efficient neural networks. *arXiv preprint arXiv:1506.02626*.

Han, Z.; Gui, X.; Cui, C.; and Yin, Y. 2020a. Towards Accurate and Robust Domain Adaptation under Noisy Environments. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 2269–2276.

Han, Z.; He, R.; Li, T.; Wei, B.; Wang, J.; and Yin, Y. 2021a. Semi-Supervised Screening of COVID-19 from Positive and Unlabeled Data with Constraint Non-Negative Risk Estimator. In Feragen, A.; Sommer, S.; Schnabel, J. A.; and Nielsen, M., eds., *Information Processing in Medical Imaging - 27th International Conference, IPMI 2021, Virtual Event, June 28-June 30, 2021, Proceedings*, volume 12729 of *Lecture Notes in Computer Science*, 611–623. Springer.

Han, Z.; Wei, B.; Hong, Y.; Li, T.; Cong, J.; Zhu, X.; Wei, H.; and Zhang, W. 2020b. Accurate Screening of COVID-19 Using Attention-Based Deep 3D Multiple Instance Learning. *IEEE Trans. Medical Imaging*, 39(8): 2584–2594.

Han, Z.; Wei, B.; Xi, X.; Chen, B.; Yin, Y.; and Li, S. 2021b. Unifying neural learning and symbolic reasoning for spinal medical report generation. *Medical Image Anal.*, 67: 101872.

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

Laine, S.; and Aila, T. 2017. Temporal Ensembling for Semi-Supervised Learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Lee, D.-H.; et al. 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, Y.; and Liang, D. 2019. Safe semi-supervised learning: a brief introduction. *Frontiers Comput. Sci.*, 13(4): 669–676.

Li, Y.; and Zhou, Z. 2015. Towards Making Unlabeled Data Never Hurt. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(1): 175–188.

Liu, T.; and Tao, D. 2015. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3): 447–461.

Lu, X.; Ma, C.; Shen, J.; Yang, X.; Reid, I.; and Yang, M.-H. 2020a. Deep Object Tracking with Shrinkage Loss. *IEEE transactions on pattern analysis and machine intelligence*.

Lu, X.; Wang, W.; Shen, J.; Crandall, D.; and Luo, J. 2020b. Zero-shot video object segmentation with co-attention siamese networks. *IEEE transactions on pattern analysis and machine intelligence*.

Lu, X.; Wang, W.; Shen, J.; Crandall, D.; and Van Gool, L. 2021. Segmenting Objects from Relational Visual Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Ma, H.; Chen, T.; Hu, T.-K.; You, C.; Xie, X.; and Wang, Z. 2021. Good Students Play Big Lottery Better. *arXiv preprint arXiv:2101.03255*.

Miyato, T.; Maeda, S.; Koyama, M.; and Ishii, S. 2019. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(8): 1979–1993.

Oliver, A.; Odena, A.; Raffel, C.; Cubuk, E. D.; and Goodfellow, I. J. 2018. Realistic Evaluation of Deep Semi-Supervised Learning Algorithms. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 3239–3250.

Pham, H.; Xie, Q.; Dai, Z.; and Le, Q. V. 2020. Meta Pseudo Labels. *CoRR*, abs/2003.10580.

Ren, Z.; Yeh, R. A.; and Schwing, A. G. 2020. Not All Unlabeled Data are Equal: Learning to Weight Data in Semi-supervised Learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Rizve, M. N.; Duarte, K.; Rawat, Y. S.; and Shah, M. 2021. In Defense of Pseudo-Labeling: An Uncertainty-Aware Pseudo-label Selection Framework for Semi-Supervised Learning. *CoRR*, abs/2101.06329.

Sajjadi, M.; Javanmardi, M.; and Tasdizen, T. 2016. Regularization With Stochastic Transformations and Perturbations for Deep Semi-Supervised Learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 1163–1171.

Shen, J.; Liu, Y.; Dong, X.; Lu, X.; Khan, F. S.; and Hoi, S. C. 2021. Distilled Siamese Networks for Visual Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C.; Cubuk, E. D.; Kurakin, A.; and Li, C. 2020. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Tarvainen, A.; and Valpola, H. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 1195–1204.

Wang, C.; Zhang, G.; and Grosse, R. 2020. Picking winning tickets before training by preserving gradient flow. *arXiv preprint arXiv:2002.07376*.

Xia, X.; Liu, T.; Han, B.; Gong, C.; Wang, N.; Ge, Z.; and Chang, Y. 2020. Robust early-learning: Hindering the memorization of noisy labels. In *International Conference on Learning Representations*.

Xie, Q.; Dai, Z.; Hovy, E. H.; Luong, T.; and Le, Q. 2020a. Unsupervised Data Augmentation for Consistency Training. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Xie, Q.; Luong, M.; Hovy, E. H.; and Le, Q. V. 2020b. Self-Training With Noisy Student Improves ImageNet Classification. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 10684–10695.

Yu, H.; Edunov, S.; Tian, Y.; and Morcos, A. S. 2019. Playing the lottery with rewards and multiple languages: lottery tickets in rl and nlp. *arXiv preprint arXiv:1906.02768*.

Yu, Q.; Ikami, D.; Irie, G.; and Aizawa, K. 2020. Multi-task curriculum framework for open-set semi-supervised learning. In *European Conference on Computer Vision*, 438–454. Springer.

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

Zhang, D.; Ahuja, K.; Xu, Y.; Wang, Y.; and Courville, A. 2021. Can Subnetwork Structure be the Key to Out-of-Distribution Generalization? *arXiv preprint arXiv:2106.02890*.

Zhou, D.; Bousquet, O.; Lal, T. N.; Weston, J.; and Schölkopf, B. 2003. Learning with Local and Global Consistency. In *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*, 321–328.