ENHANCING SEMI-SUPERVISED LEARNING VIA SELF-INTERESTED COALITIONAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Semi-supervised learning holds great promise for many real-world applications, due to its ability to leverage both unlabeled and expensive labeled data. However, most semi-supervised learning algorithms still heavily rely on the limited labeled data to infer and utilize the hidden information from unlabeled data. We note that any semi-supervised learning task under the self-training paradigm also hides an auxiliary task of discriminating label observability. Jointly solving these two tasks allows full utilization of information from both labeled and unlabeled data, thus alleviating the problem of over-reliance on labeled data. This naturally leads to a new learning framework, which we call Self-interested Coalitional Learning (SCL). The key idea of SCL is to construct a semi-cooperative "game", which forges cooperation between a main self-interested semi-supervised learning task and a companion task that infers label observability to facilitate main task training. We show with theoretical deduction its connection to loss reweighting on noisy labels. Through comprehensive evaluation on both classification and regression tasks, we show that SCL can consistently enhance the performance of semi-supervised learning algorithms.

1 INTRODUCTION

Despite the huge success of supervised learning with deep neural networks, they require large amounts of labeled data for model training to achieve high performance (Meng et al., 2019; Prabhu & Varma, 2014). Collecting labeled data can be very difficult and costly in practice (Meng et al., 2020). The challenge can be further exacerbated as in many cases, human or machine labeled data may contain errors or noises (Natarajan et al., 2017; Schnabel et al., 2016). Semi-supervised learning (SSL) (Chapelle et al., 2006) that leverages both labeled and unlabeled data has become an increasingly promising yet still challenging area, and is widely applicable to real-world problems.

To utilize the hidden information in the unlabeled data, a general and popular semi-supervised learning paradigm is through self-training (Yarowsky, 1995; Lee et al., 2013). Self-training uses a previously learned model to predict labels for the unlabeled data (pseudo-labeling) which are then used for subsequent model training (Yarowsky, 1995; Lee et al., 2013; Laine & Aila, 2017; Tarvainen & Valpola, 2017; Iscen et al., 2019). Despite some empirical successes, self-training methods still suffer from two core challenges, which are *over-reliance on labeled data* and *error accumulation*. Most existing semi-supervised learning methods (Yarowsky, 1995; Lee et al., 2013; Laine & Aila, 2017; Sohn et al., 2020; Xie et al., 2020; Kipf & Welling, 2017) assume the labeled and unlabeled data follow the same or similar data distribution, thus the data-label mapping mechanisms learned from the labeled data are somewhat transferable to unlabeled data. Essentially, the algorithm is still reinforcing the information in the labeled data rather than mining additional information from the unlabeled data. Second, the pseudo-labels of unlabeled data can be incorrectly predicted. Using such biased information for training in subsequent epochs could increase confidence in erroneous predictions, and eventually leading to a vicious circle of error accumulation (Cai et al., 2013; Arazo et al., 2020). The situation can be even worse when the labeled data contain errors or noises, as the model learned from labeled data will make more mistakes, resulting in more severe error accumulation.

Alleviating the issue of over-reliance on labeled data requires extracting additional sources of information from the unlabeled data. One recent approach is to introduce input consistency regularization (Sajjadi et al., 2016). Consistency regularization enforces the stability of the predictions with respect

to the transformations of the unlabeled data (e.g. data augmentation on images such as rotation, flip and shift, etc.). Although such data perturbation or augmentation are well-defined for images, they are not directly transferable to general settings, hence the success of consistency regularization methods are mostly restricted to image classification tasks (Sajjadi et al., 2016; Liu et al., 2019; Xie et al., 2020; Sohn et al., 2020). We provide a new insight by noting that, under the self-training paradigm, every semi-supervised learning task hides another auxiliary task of discriminating whether the data label is real or a pseudo-label predicted by a machine oracle. Jointly solving the main task together with the auxiliary task allows more sufficient utilization of the hidden information in unlabeled data.

To break the vicious circle of error accumulation, one needs to not always trust the machine-labeled data. A natural idea is to consider the reliability or confidence of data labels and treat them differently during training. A connected approach in supervised learning literature is to use the notion of "soft label" with confidence probability or weights, which has been verified in a number of noisy label learning problems (Natarajan et al., 2017; Schnabel et al., 2016; Shen et al., 2020; Algan & Ulusoy, 2021). This treatment can also help a semi-supervised learning algorithm gain the ability to robustly learn from noisy labels.

Combining previous insights, we develop a novel semi-supervised learning framework, called Self-interested Coalitional Learning (SCL). SCL contains three ingredients. First, it simultaneously solves a companion task by learning a discriminator as a critic to predict label observability (whether it is a true label or a pseudo-label), which is used to mine additional information in the unlabeled data while facilitating main task learning. Second, the output of the discriminator is in fact a measure of label reliability, thus can be interpreted as the confidence probabilities of labels (Qin et al., 2021). This converts the original problem into a soft label learning problem. Finally and most importantly, SCL introduces a cooperative-yet-competitive learning scheme to boost the performance of both the main task and the companion task. Specifically, SCL forges cooperation between both tasks by providing extra information to each other (the main task provides prediction loss, companion task provides label confidence measures), leading to improved performance of both tasks. Moreover, we consider the main task model to be self-interested which tries to challenge the discriminator by providing as little information as possible. This design forms a semi-cooperative "game" with a partially adversarial main task model, which can be shown theoretically equivalent to a loss reweighting mechanism on noisy soft labels. Under this design, the main task and the companion task in SCL are neither fully cooperative as in multi-objective optimization (Deb, 2014) nor fully adversarial to each other as in adversarial learning (Goodfellow et al., 2014; Matyasko & Chau, 2018), but balance the benefit of cooperation and competition. In summary, the contribution of this work include:

- We develop a novel and very general self-interested coalitional learning framework for SSL that fully utilizes the hidden information in the unlabeled data while addressing the error accumulation issue in the self-training paradigm of SSL.
- SCL can be easily incorporated into a wide range of semi-supervised algorithms (e.g. image classification, label propagation and data imputation) with limited changes on the original method. We show with comprehensive empirical experiments that SCL consistently achieves superior performance and robustness compared with the original and self-training version of different semi-supervised learning algorithms.
- SCL naturally provides the data confidence measures from the discriminator as a byproduct of the learning process, which offers additional interpretability of the semi-supervised learning task. This can be particularly useful for many practical applications, while also providing extra benefits for scenarios with noisy data labels.

2 PRELIMINARIES

2.1 PROBLEM DEFINITIONS AND FORMULATIONS

We formulate the semi-supervised learning problem as learning a model $f(\cdot)$ with input data $x \in X$ to predict the label $y \in Y$. In SSL, only a partial set of data labels Y_L with ground truth are given in the training set, the rest are unlabeled Y_U ($Y = Y_L \bigcup Y_U$). For convenience of later discussion, we denote L as the set of indices of data samples with labels and U as the set of indices for unlabeled data. In many real-world SSL problems, the size of labeled data $|L|$ is often limited. In extreme cases, Y_L may potentially contain errors or noises. We further define a mask vector M to denote the

observability of labels over each data sample. We set $m_i = 1$ if $i \in L$ and $m_j = 0$ if $j \in U$. In our proposed SCL framework, we consider following common settings of semi-supervised classification (inductive classification, label propagation) and regression (data imputation) tasks:

Inductive classification: a typical SSL task setting is to construct a classifier to predict labels $y \in Y$ for any object in the input space $x \in X$. Common examples include semi-supervised image classification tasks (Sajjadi et al., 2016; Sohn et al., 2020; Xie et al., 2020), where only a subset of data samples are given the known labels Y_L and the rest data labels Y_U are unknown.

Label propagation: another class of SSL tasks under transductive setup is to use all input samples X and observed labels Y_L to train a classifier to predict on unseen labels Y_U. Examples include classifying nodes in a graph given only small subset of node labels Y_L (Kipf & Welling, 2017).

Data imputation: one of the special case of SSL problem is that the missing state of input and output data are strongly correlated (Richardson et al., 2020; Qin et al., 2021), such as the data imputation tasks. In data imputation, the inputs X are partially filled and the labels Y are equivalent to a reconstructed version of X obtained through regression.

2.2 Auxiliary task in semi-supervised learning problems

We begin our discussion by first noting that, if we feed the unknown labels Y_U with the model f predicted labels $\tilde{Y}_U = \hat{Y}_U^{(k)}$ ($\hat{Y}_U^{(k)} = f(X)$ represents the labels from the k-th round of pseudo-labeling) or other machine-generated labels, there actually hides an auxiliary task of discriminating whether the data label is real or a pseudo-label. Denote all the data labels under pseudo-labeling as $\tilde{Y} = Y_L \bigcup \tilde{Y}_U$. We can train a discriminator $d(\cdot)$ to tell the confidence measure $p \in P$ of whether $x \to \tilde{y}$ ($x \in X, \tilde{y} \in \tilde{Y}$) is a valid mapping. This is always learnable since the ground truth of P is exactly the observability mask M. With a slight abuse of notation, we formulate the main SSL task A and the auxiliary companion task B as follows:

$$\begin{aligned} A &: f(X) = \hat{Y}, \qquad \mathcal{L}_A = loss_A(Y_L, \hat{Y}_L) \\ B &: d(X, \tilde{Y}) = P, \qquad \mathcal{L}_B = loss_B(P, M) \end{aligned} \tag{1}$$

where $loss_A$ is the original loss function of the main task and $loss_B$ can be any binary classification loss between P and M, such as the binary cross entropy (BCE) loss. Jointly solving above two tasks allows exploiting the underlying relationship between input data X and data label Y from another angle, which can potentially provide more information to facilitate main task training.

There are some other works embody the similar idea of jointly learning two tasks. For example, the multi-task style multi-objective optimization methods (Ruder, 2017; Deb, 2014) exploit the shared information and underlying commonalities between two tasks and solve the problem by minimizing an augmented loss. The generative adversarial learning (Goodfellow et al., 2014) makes a generator and a discriminator learn against each other, thus improves the performance of both tasks. However, naïvely applying these two approaches has some drawbacks. For example, multi-objective optimization cannot handle potential contradictions of the two tasks in certain settings, where jointly minimizing the augmented loss may impede both tasks from achieving the best performance (Qin et al., 2021). Whereas under the adversarial learning framework, $loss_A$ is implicitly optimized through the discriminator, which could result in potential loss of information. Moreover, solving the minimax optimization problem under a GAN-style model is much harder compared with directly minimize both loss functions explicitly in (1) in a supervised learning fashion.

3 Self-interested coalitional learning

In this section, we introduce a new learning strategy to jointly solving the two tasks in Problem (1) for semi-supervised learning, which is called self-interested coalitional learning (SCL). As its name suggests, SCL forges cooperation between a self-interested main task and an unselfish companion task for boosted performance. We show with theoretical derivation that under the pseudo-labeling paradigm, this special learning scheme will lead to a soft label learning formulation for the original SSL task. We also provide an intuitive interpretation of the SCL scheme.

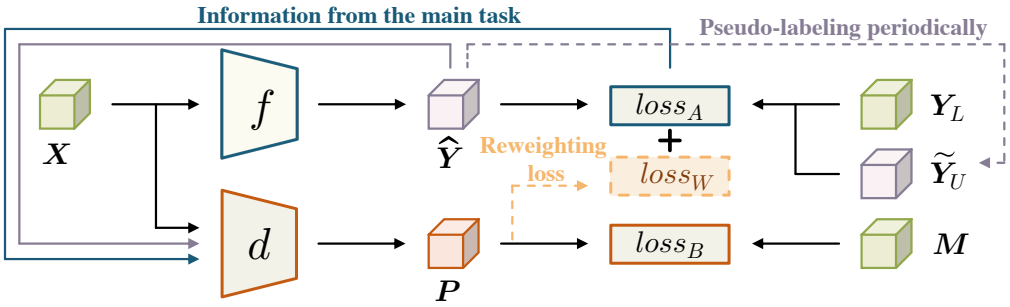


Figure 1: Illustration of self-interested coalitional learning

3.1 ALTERNATIVE FORMULATION UNDER SCL

We consider an alternative formulation of Problem (1) by modifying the loss of the main task $\mathcal{L}_A$ and the inputs of the companion task to establish cooperation and information sharing:

$$\begin{aligned} A: f(X) = \hat{Y}, \qquad & \mathcal{L}_A = loss_A(\tilde{Y}, \hat{Y}) + \alpha \cdot loss_W(\tilde{Y}, \hat{Y}, P) \\ B: d(X, \hat{Y}, g(\tilde{Y}, \hat{Y})) = P, \quad & \mathcal{L}_B = loss_B(P, M). \end{aligned} \tag{2}$$

where $g(\tilde{y}, \hat{y})$ is some form of information provided by the main task. We can simply model the information $g(\cdot)$ as the element-wise loss term of the main task, i.e. $g(\tilde{y}, \hat{y}) = loss_A(\tilde{y}, \hat{y})$, such as the cross entropy loss for classification tasks and mean square error (MSE) loss for regression tasks. Moreover, $loss_W(\tilde{Y}, \hat{Y}, P)$ is a corrective additional loss term impacted by the label confidence measure information P provided by the discriminator d, and α is its weight parameter. Due to the existence of the corrective information from $loss_W$, we now train the main task with all data labels $\tilde{Y}$ rather than only with observed labels Y_L, thus allows more sufficient utilization of information from the unlabeled data. We illustrate the strategy of SCL in Figure 1.

The design of SCL has several interesting properties. First, since additional information are provided to both tasks, they can be both learned better. Second, a less well-learned model f will produce large element-wise error $loss_A(\tilde{y}, \hat{y})$ for a predicted label $\hat{y}$, which in turn provides more information in $g(\cdot)$ for the discriminator d to distinguish whether a label is real or a pseudo-label. As f is learned better during the cooperative learning process, when g no longer provides useful information for discriminating the label observability, then we may have reason to believe f has achieved satisfactory inference performance. Hence minimizing the information in $g(\cdot)$ coincides with improving the learning of f. This suggests that if we alter the learning direction of f using $loss_W$ to encourage it providing as little information in $g(\cdot)$ as possible, then we might obtain a better learned f. This in fact forms a cooperative-yet-competitive "game" between both tasks, with the main task been self-interested and partially competitive. Finally, it also allows the discriminator to detect errors or noises in observed data labels. As the erroneous (x, y) data pairs may possess different $x \to y$ mapping patterns against normal data, thus are more likely to be identified as pseudo-labels. Such negative impact can be further corrected using $loss_W$ with label confidence measure P.

3.2 DERIVATION OF THE ADDITIONAL LOSS TERM $loss_W$

Until this point, the only mystery about SCL is the exact form of $loss_W$ that embodies the self-interested behavior for the main task model f. We resort to calculus of variations to provide theoretical derivation of $loss_W$.

Under the alternative formulation (2), both the discriminator d and its loss $\mathcal{L}_B$ are impacted the information $g(\cdot)$ provided by f, thus they are now functionals of f. We wish to inspect how the variation of model f impacts $\mathcal{L}_B$. Denote θ_f, θ_d as the model parameters of model f and d. Here, we fix the the discriminator d and its parameters θ_d to exclude its own influence, then $d(\theta_d, x, f)$ can be replaced by $d(x, f)$. Similarly, with d fixed, we can define the functional $J(f)$ for $\mathcal{L}_B$ and further eliminate the effect of changes in x on d by integrating over x, mathematically:

$$J(f) = \mathcal{L}_B = \int \frac{\partial \mathcal{L}_B}{\partial d} dx = \int F(x, f, f') dx \tag{3}$$

where $F(x, f, f') = \partial\mathcal{L}_B/\partial d$ is assumed to be a continuously differentiable function with respect to x, f and f'. Although the discriminator d aims to minimize its loss $\mathcal{L}_B$, as mentioned previously, we want the main task model f to challenge the discriminator by providing as little information in g as possible. This essentially requires changing the behavior of f such that the functional $J(f)$ attains its maxima. The extrema (maxima or minima) of functional $J(f)$ can be obtained by finding a function f where the functional derivative of $J(f)$ is equal to zero. By Taylor expansion,

$$J(f) = J(f_{\theta_f^0}) + J'(f_{\theta_f^0})\delta f + o(\delta f) \tag{4}$$

where δf is the variation of f. Since $o(\delta f)$ is a higher order infinitesimal term at $\theta_f \to \theta_f^0$, and assuming that $o(\delta f)$ is small and ignorable, we can obtain the finite difference approximation of the functional $J(f)$ by plugging it into Eq.(3),

$$\Delta J(f) = J(f) - J(f_{\theta_f^0}) = J'(f_{\theta_f^0})\delta f = \frac{d\int F(x, f, f')dx}{d\theta_f}\delta f \tag{5}$$

If the uniform continuity of F and its derivative is satisfied, we can simplify the derivative of $J(f)$ with respect to θ_f as follows, while swapping the order of dx with δf:

$$\Delta J(f) = \frac{\int \partial F(x, f, f')dx}{\partial\theta_f}\delta f = \int \frac{\partial F(x, f, f')}{\partial\theta_f}\delta f dx \tag{6}$$

As f can be any function in an arbitrary function space $\mathcal{F}$. Ensuring the functional derivative of $J(f)$ equal to zero ($\Delta J(f) = 0$) requires $\partial F(x, f, f')/\partial\theta_f = 0$, which is

$$\frac{\partial F(x, f, f')}{\partial\theta_f} = \frac{\partial F}{\partial g} \cdot \frac{\partial g}{\partial f} \cdot \frac{\partial f}{\partial\theta_f} = \frac{\partial F}{\partial g} \cdot \nabla_{\theta_f} loss_A = 0 \tag{7}$$

The last step derivation in above equation is due to our design of letting $g(\cdot)$ be the element-wise loss function of f ($g(\tilde{y}, \hat{y}) = loss_A(\tilde{y}, \hat{y})$).

As an example, suppose we use BCE loss for $\mathcal{L}_B$, and write p_i as the output value of $d(X, \hat{Y}, g(\tilde{Y}, \hat{Y}))$ with index i, the Eq.(7) can be written as

$$\frac{\partial F}{\partial\theta_f} = \frac{\partial F}{\partial g} \cdot \nabla_{\theta_f} loss_A = -\sum_{i\in L} \frac{1}{p_i}\nabla_{\theta_f} loss_A(\tilde{y}_i, \hat{y}_i) + \sum_{j\in U} \frac{1}{1-p_j}\nabla_{\theta_f} loss_A(\tilde{y}_j, \hat{y}_j) \tag{8}$$

Note that the result of above equation can be equivalently perceived as the gradient of a new term $L_f(\tilde{Y}, \hat{Y}, P)$ ($\partial F/\partial\theta_f = -\partial L_f/\partial\theta_f$) with following form:

$$-L_f(\tilde{Y}, \hat{Y}, P) = \sum_{i\in L} \frac{1}{p_i} \cdot loss_A(\tilde{y}_i, \hat{y}_i) - \sum_{j\in U} \frac{1}{1-p_j} \cdot loss_A(\tilde{y}_j, \hat{y}_j) \tag{9}$$

Consequently, minimizing $-L_f$ with respect to θ_f will drive the functional derivative of $J(f)$ to zero ($-\partial L_f/\partial\theta_f = \partial F/\partial\theta_f = 0 \implies \Delta J(f) = 0$), which also lead to the maxima of $J(f)$ (update with gradient direction $-\partial F/\partial\theta_f$). We now uncover the exact form of $loss_W$, which can be set as $loss_W(\tilde{Y}, \hat{Y}, P) = -\mathcal{L}_f(\tilde{Y}, \hat{Y}, P)$ to realize the self-interested behavior of f. The resulting $loss_W$ can be considered as an additional reweighting loss that complements the original loss term $loss_A$ based on the estimated confidence measures P from the discriminator d.

3.3 Interpretation of SCL

Adding $loss_W$ to the original loss term $loss_A$, we can obtain the complete loss for the main task as:

$$\mathcal{L}_A = \sum_{i\in L} \left(1 + \frac{\alpha}{p_i}\right) \cdot loss_A(y_i, \hat{y}_i) + \sum_{j\in U} \left(1 - \frac{\alpha}{1-p_j}\right) \cdot loss_A(\tilde{y}_j, \hat{y}_j) \tag{10}$$

This essentially transforms the original main task into a cost-sensitive learning problem (Zadrozny et al., 2003) by imposing following soft-labeling weights on the errors of each data label as:

$$\text{Soft-labeling weights} = \begin{cases} 1 + \alpha/p_i, & i \in L \\ 1 - \alpha/(1-p_j), & j \in U \end{cases} \tag{11}$$

We notice that the above soft-labeling weights induce very different behaviors on the loss $loss_A(\tilde{y}, \hat{y})$ of labeled and pseudo-labeled samples. As p_i represents the judgment of the discriminator d on whether y_i is labeled or not. For labeled samples, the loss are boosted by additional weight of α/p_i, which means if the discriminator d judges an labeled samples $y_i \in Y_L$ to be unlabeled or unreliable (confidence $p_i \to 0$), the loss of this sample will get significantly boosted. This forces the main task model f to pay more attention to infer the problematic labeled samples. As unlabeled samples $y_j \in Y_U$ are periodically filled with pseudo-labels generated by f (initially filled with random values), SCL does not fully trust these labels and may even choose to enlarge the gap with the pseudo-label when $p > 1 - \alpha$ ($\alpha \leq 1$). Specifically, the more likely an label y_j is considered reliable by the discriminator ($p_j \to 1$), the stronger it forces the learning process to ignore further improvement on inferring y_j.

Note that uniform continuity needs to be satisfied in Eq.(6), however, the derivative of BCE loss will result in $1/p_i$ and $1/(1-p_i)$ which can potentially take infinite value, violating the continuity assumption. We thus clip their values as $\min(1/p_i, H)$ and $\min(1/(1-p_i), H)$ in our practical algorithm, where H is a predetermined threshold ($H = 10$ in our implementation). Algorithm 1 presents the practical algorithmic framework of SCL.

Algorithm 1 Self-interested coalitional learning framework

Require: Training samples $\{X, Y\} = \{X_L \bigcup X_U, Y_L \bigcup Y_U\}$, observability mask M, weight parameter α

Randomly initialized θ_f, θ_d, Y_U, and let $\tilde{Y}_U \leftarrow Y_U$
while training **do**
  Inference the all data labels $\hat{Y}$ using the main task model f
  Calculating the element-wise loss of the main task $loss_A(\tilde{Y}, \hat{Y})$ as $g(\tilde{Y}, \hat{Y})$
  Predict the label confidence measures P using the discriminator d and $g(\tilde{Y}, \hat{Y})$
  Update θ_d^{t+1} by minimizing $\mathcal{L}_B$ in Eq.(2) **// Update d with information from main task**
  Update θ_f^{t+1} by minimizing $\mathcal{L}_A$ in Eq.(10) **// Update f with information from companion task**
  update the pseudo-label $\tilde{Y}_U \leftarrow \hat{Y}_U$
end while

4 EXPERIMENTS

The proposed SCL framework can be easily incorporated in a variety of SSL tasks. We choose three representative SSL tasks and apply SCL on their official implementations to evaluate the performance of SSL, including: image classification (He et al., 2016) for inductive classification, label propagation (Kipf & Welling, 2017) for transductive classification and data imputation (Gondara & Wang, 2018) for regression.

Image classification: We use RestNet18 for image classification on the fashion-mnist dataset (Xiao et al., 2017), with 6,000/10,000 training/testing set partition. Note that the training set only contains 10% of the original fashion-mnist dataset which aims to substantially increase the task difficulty.

Label propagation: We trained a two-layer GCN (Kipf & Welling, 2017) integrated with SCL and evaluate the prediction accuracy on Cora dataset(Sen et al., 2008). The test set consists of 1000 labeled examples extracted from a knowledge graph with 2,708 nodes and 5,429 edges. The labeling rate represents the number of labeled nodes used for training divided by the total number of nodes in the training set.

Data imputation: We use a deep denoising autoencoder (DAE) (Gondara & Wang, 2018) for data imputation task under regression setting. We use the UCI online news popularity dataset (Asuncion & Newman, 2007) for evaluation. The dataset is split the training/testing set with the ratio of 0.8/0.2.

For all tasks, we remove certain proportion (referred as *missing rate*) of data labels in the training set and treat them as unlabeled data in the semi-supervised setting. Regarding the initial pseudo-labels for unlabeled data, for the regression task (data imputation), we compute the mean μ of the labeled data and fill the missing entries with μ adding random noise; for the classification tasks, we fill the missing label Y_U by randomly selecting a data label. We present the average results over 10 random seeds for all the experiments. Implementation details are discussed in Appendix A.

Missing rate (%)	Image classification Accuracy (%)		Label propagation Accuracy (%)		Data imputation MSE	
	ResNet18	ResNet18+SCL	GCN	GCN+SCL	DAE	DAE+SCL
10	83.59	84.69	82.90	84.33	0.9021	0.8851
20	83.06	85.15	82.60	84.00	0.9516	0.9344
30	82.89	85.38	83.00	83.33	0.9658	0.9430
40	82.43	84.98	82.30	83.00	0.9872	0.9629
50	81.94	84.61	79.90	81.33	1.0024	0.9716
60	81.23	83.87	80.40	82.33	0.9895	0.9867
70	79.70	82.90	78.50	83.67	1.1593	1.1582
80	78.84	81.31	70.10	80.33	1.1429	1.1425

Table 1: Comparison of the original models and SCL-integrated version for different tasks

4.1 PERFORMANCE EVALUATION OF SCL

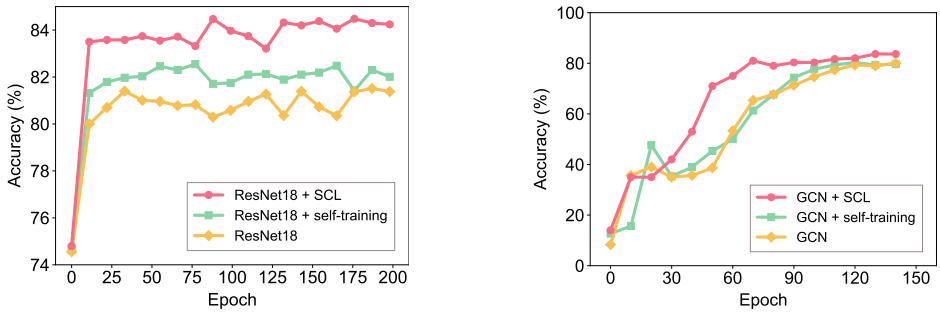
We compared our method with the original model in three representative SSL tasks to demonstrate the effectiveness and robustness of SCL under diverse problem setting. Results are summarized in Table 1. We report the classification accuracies and imputation error measured by MSE. SCL consistently achieves superior performance over all tasks, which shows the universality of SCL under diverse SSL problem setting. We also use 8 different levels of missing rates to verify the robustness of SCL. By randomly replacing the original labels with random labels, we gradually increase the missing rate of the training set from 10% to 80%. It is observed in Table 1 that most of the original models experience substantial performance deterioration with the increase of missing rate. By contrast, SCL remains high level of robustness with limited performance drop in most tasks.

For the inductive image classification task, the accuracy of the original model continuously decrease when fewer labeled data are used, as less information can be used to well learn the inductive classifier. SCL performs cost-sensitive learning for different data samples to maximally utilize the hidden information from both labeled and unlabeled data, which lead to more stable and accurate inference under different missing rates. For the transductive classification task like label propagation, all the input data X and the labeled data Y_L are used for inference on the unobserved labels Y_U. The amount of information from the labeled data plays a vital role for overall prediction accuracy. Hence as the missing rate becomes large, we observe a substantial performance drop for the original GCN model. SCL, on the other hand, maintains high accuracy in spite of high missing rate by continuously soft pseudo-labeling the data to ensure the reliable pseudo-labels are properly used to facilitate the main task training, thus exhibits high robustness under missing information.

4.2 COMPARISON OF SCL AND SELF-TRAINING VIA PSEUDO-LABELING

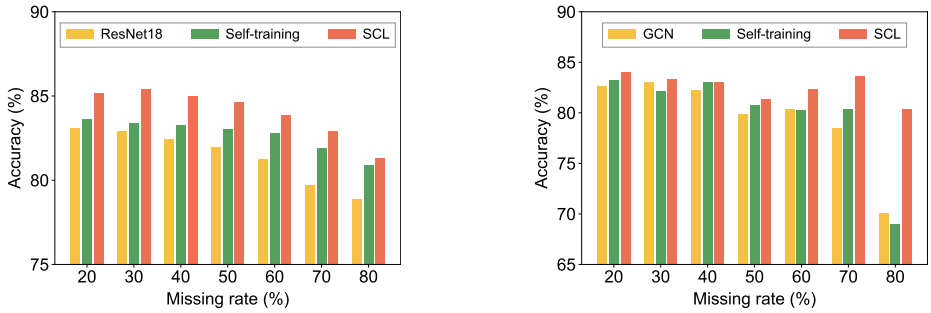
To further validate the effectiveness of SCL under semi-supervised settings, we compare the performance of SCL with the widely used pseudo-labeling approach in self-training (Yarowsky, 1995; Lee et al., 2013; Laine & Aila, 2017; Iscen et al., 2019). We compare their performances in three different aspects, i.e., convergence speed, accuracy under different missing rates and the degree of error accumulation under pseudo-labeling. In Figure 2, we can see that the learning speed of the main task model with the corrective reweighting loss term $loss_W$ in SCL is significantly faster than the original model. Meanwhile, since the hard pseudo-labels in self-training are used as true label for model learning rather than carefully reweighted as soft labels in SCL, certain degree of oscillation is observed during the learning process, which impacts the learning speed as well as the final performance of the model. This demonstrates the benefit of using soft-labeling weights to correct unreliable label information in SCL.

Figure 3 shows the accuracy of the original, self-trained and SCL versions of models for classification tasks with different missing rates. We observe that SCL maintains high accuracy under different missing rates and consistently outperforms the original and self-training versions. In the label propagation task, although self-training increases the training samples by periodically pseudo-labeling high confidence unlabeled data, the accuracy of the model could decreases due to the introduction of wrong labeling information, which results in subsequent error accumulation and performance drop.



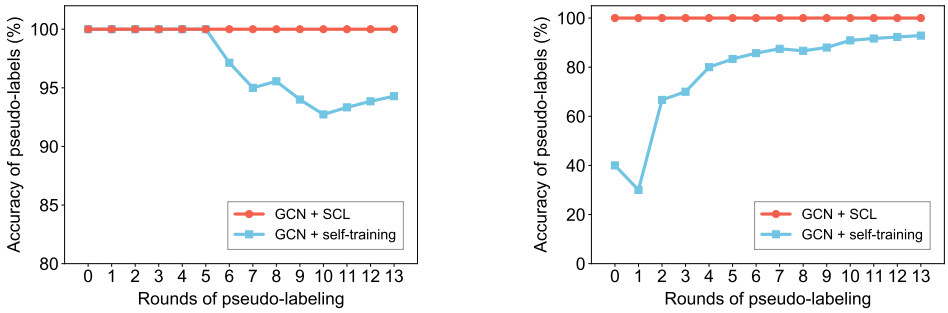
(a) Image classification (b) Label propagation

Figure 2: Test accuracy of different methods in image classification and label propagation tasks



(a) Image classification (b) Label propagation

Figure 3: Performance under different missing rate



(a) Accuracy of pseudo-labels under 20% missing rate (b) Accuracy of pseudo-labels under 70% missing rate

Figure 4: Error accumulation of different methods in the label propagation task

To further investigate the impact of error accumulation in pseudo-labels during model training, we record the accuracy of pseudo-labels after each round of pseudo-labeling in self-training. The test accuracy of these samples in the SCL version of the model are also recorded for comparison. To ensure the stability of self-training, we add pseudo-labels every 10 training epochs. Different from self-training that only label high confidence unlabeled samples, SCL introduces pseudo-labels for all unlabeled data Y_U but with soft-labeling weights. Figure 4 shows the error accumulation trends for self-training and SCL in the label propagation task. When the missing rate is low (Figure 4(a)), self-training maintains a high accuracy at the beginning, but model performance will fluctuate and decrease with the involvement of more potentially problematic pseudo-labels. On the other hand, when the missing rate is high and available information is scarce (Figure 4(b)), gradually involving pseudo-labeled data is beneficial for self-training, resulting in slowly increasing test accuracy, although the model still struggles to achieve high accuracy. Despite the occurrence of error accumulation in self-training, we barely observe such phenomenon in SCL. The test accuracy of the pseudo-labels remains high and stable across different rounds of pseudo-labeling, which clearly suggests the ability of SCL in avoiding error accumulation in pseudo-labels.

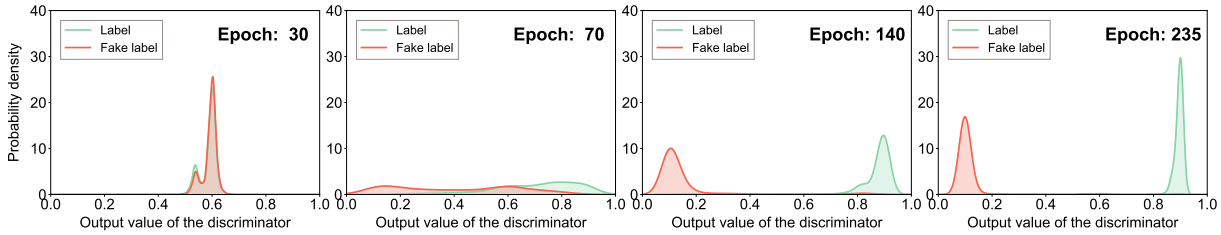


Figure 5: The probability distribution given by the discriminator during the training process

4.3 Interpretability of discriminator outputs

To evaluate the judgement of the discriminator d, we analyze its output value distribution on the labeled and intentionally introduced fake/noisy labeled data in different training stages of image classification task. The four plots in Figure 5 clearly show the transition of an improved discriminator, which gradually gains better capability to discriminate true labels ($p \to 1$) and fake/noisy labels ($p \to 0$). The interpretability of the discriminator is a byproduct of SCL, which can be particularly useful for many real-world SSL tasks or scenarios involving error or noise in labeled data.

5 Related Work

Self-training via pseudo-labeling. Pseudo-labeling (Triguero et al., 2015; Yarowsky, 1995; Lee et al., 2013) is one of the most widely used self-training method in semi-supervised learning, which uses a supervised model that is iteratively trained on both labeled and pseudo-labeled data in previous iterations of the algorithm. The basic assumption in most models (Yarowsky, 1995; Lee et al., 2013; Laine & Aila, 2017; Tarvainen & Valpola, 2017; Iscen et al., 2019) is that an unlabeled instance can be labeled for training if the underlying model has high confidence in its prediction. When the model makes inaccurate predictions, it is likely to cause error accumulation that negatively impacts training (Cai et al., 2013; Arazo et al., 2020). To alleviate this problem, many studies (Natarajan et al., 2017; Schnabel et al., 2016; Shen et al., 2020) guide the model learning based on various label confidence measures and avoid the model from being overly confident on incorrect pseudo-labels. The above works only focus on leveraging both labeled and unlabeled data, but not fully applicable to scenarios with noisy labeled data. SCL considers the contribution of both labeled and unlabeled data through soft labeling, which better addresses aforementioned drawbacks.

Learning from noisy labels. The lack of high-quality labels is common in many real-world scenarios (Natarajan et al., 2017; Schnabel et al., 2016). Erroneous or noisy labels can severely degrade the generalization performance of deep neural networks. Learning from noisy labels (Song et al., 2020) is becoming an increasinly important task in modern deep learning applications. Early studies (Cesa-Bianchi et al., 2011) require unbiased gradient estimates of the loss to provide learning guarantees. Some recent studies (Natarajan et al., 2017; Schnabel et al., 2016; Shen et al., 2020; Algan & Ulusoy, 2021) provide unbiased estimators for evaluating and training under noise data. However, these methods still require that the observed data have a relatively well-behaved (noise-free) underlying distribution, thus the noisy data are weighted according to the same distribution. Satisfying such requirements is difficult in practice and the observed distribution may be biased across the data. In contrast, SCL bypasses the dependence on the original data distribution and learn label confidence measures adaptive using an additional discriminator to facilitate model training.

6 Conclusion

We introduce a novel self-interested coalitional learning framework, which can be easily incorporated into a wide range of semi-supervised learning algorithms for enhanced performance. SCL improves semi-supervised model learning by constructing a semi-cooperative "game" that establishes cooperation between the main self-interested semi-supervised learning task and a companion task of inferring the observability of labels. Meanwhile, it also offers additional confidence measures for the original and inferred data labels, which can be particularly useful for the practical applications with noisy labels. Through comprehensive evaluations, we show that SCL can consistently improve the performance and robustness of semi-supervised classification and regression algorithms.

REFERENCES

Görkem Algan and Ilkay Ulusoy. Meta soft label generation for noisy labels. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 7142–7148. IEEE, 2021.

Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2020.

Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

Xiao Cai, Feiping Nie, Weidong Cai, and Heng Huang. Heterogeneous image features integration via multi-modal semi-supervised learning model. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1737–1744, 2013.

Nicolò Cesa-Bianchi, Shai Shalev-Shwartz, and Ohad Shamir. Online learning of noisy data. *IEEE Trans. Inf. Theory*, 57(12):7907–7931, 2011.

Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien (eds.). *Semi-Supervised Learning*. The MIT Press, 2006.

Kalyanmoy Deb. Multi-objective optimization. In *Search methodologies*, pp. 403–449. Springer, 2014.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Lovedeep Gondara and Ke Wang. MIDA: multiple imputation using denoising autoencoders. In *Advances in Knowledge Discovery and Data Mining - 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part III*, pp. 260–272, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27:2672–2680, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pp. 1026–1034, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5070–5079, 2019.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, pp. 896, 2013.

Lu Liu, Yiting Li, and Robby T Tan. Decoupled certainty-driven consistency loss for semi-supervised learning. *arXiv preprint arXiv:1901.05657*, 2019.

Alexander Matyasko and Lap-Pui Chau. Improved network robustness with adversary critic. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 10601–10610, 2018.

Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. Weakly-supervised hierarchical text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 6826–6833, 2019.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. Text classification using label names only: A language model self-training approach. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 9006–9017, 2020.

Nagarajan Natarajan, Inderjit S. Dhillon, Pradeep Ravikumar, and Ambuj Tewari. Cost-sensitive learning with noisy labels. *J. Mach. Learn. Res.*, 18:155:1–155:33, 2017.

Yashoteja Prabhu and Manik Varma. Fastxml: a fast, accurate and stable tree-classifier for extreme multi-label learning. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pp. 263–272, 2014.

Huiling Qin, Xianyuan Zhan, Yuanxun Li, Xiaodu Yang, and Yu Zheng. Network-wide traffic states imputation using self-interested coalitional learning. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pp. 1370–1378, 2021.

Trevor W. Richardson, Wencheng Wu, Lei Lin, Beilei Xu, and Edgar A. Bernal. Mcflow: Monte carlo flow models for data imputation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 14193–14202, 2020.

Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, 29:1163–1171, 2016.

Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. Recommendations as treatments: Debiasing learning and evaluation. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pp. 1670–1679, 2016.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

Zheyan Shen, Peng Cui, Tong Zhang, and Kun Kunag. Stable learning via sample reweighting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5692–5699, 2020.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 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*, 2020.

Hwanjun Song, Minseok Kim, Dongmin Park, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *CoRR*, abs/2007.08199, 2020.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 1195–1204, 2017.

Isaac Triguero, Salvador García, and Francisco Herrera. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information systems*, 42(2): 245–284, 2015.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

Qizhe Xie, Minh-Thang Luong, Eduard H. Hovy, and Quoc V. Le. 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*, pp. 10684–10695, 2020.

David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pp. 189–196, 1995.

Bianca Zadrozny, John Langford, and Naoki Abe. Cost-sensitive learning by cost-proportionate example weighting. In *Third IEEE international conference on data mining*, pp. 435–442. IEEE, 2003.

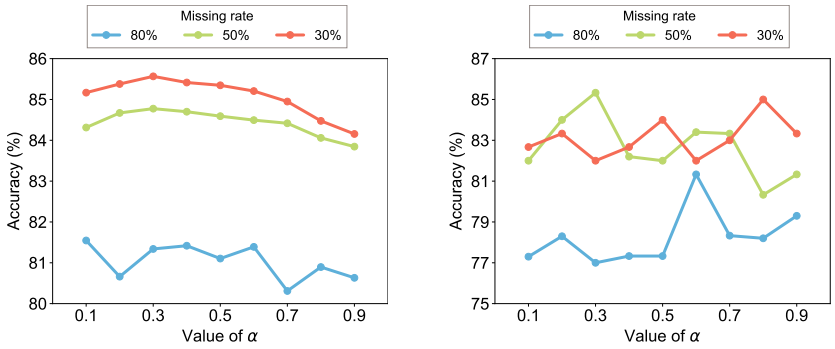
A EXPERIMENT SETTINGS

- **Image classification.**The base learner (f) is a ResNet18 (He et al., 2016) where the input channels of the first convolutional layer are replaced with 1. We use the Kaiming Normalization (He et al., 2015) to initialize the parameters of f. All input samples of training set and test set are transformed with a mean and a standard deviation for normalization. The discriminator consists of two feature extractors for X and $hatY$, respectively, and a fusion layer whose inputs are the extracted X and $hatY$ features and the main task information g. The feature extractor of X is also a ResNet18 with the number of input channels of the first convolutional layer as 1, and it outputs features with dimension 1. The feature extractor for $hatY$ is a network with three-layer FC, where the first two layers are each followed by a ReLU layer. We fuse the extracted features of X and $hatY$ by the Hadamard product. And concatenate the loss g to the fused embeddings of X, $hatY$ and fuse them through a one-layer FC. We feed the output into a sigmoid layer to output the probabilities. We use the Adam optimizer with a learning rate of 0.001 to train the base learner f and discriminator d simultaneously for a total of 300 epochs.

- **Label propagation.** We train a two-layer GCN as described in (Kipf & Welling, 2017) and evaluate prediction accuracy on a test set of 1,000 labeled examples. We initialize weights using the initialization described in (Glorot & Bengio, 2010) and accordingly (row-)normalize input feature vectors. For discriminator d, we use a five-layer and two-layer FC to encode the input X and $\hat{Y}$, and we fuse the hidden embedding of X and $\hat{Y}$ by hadamard product. We connect g to the fusion embedding of $X, \hat{Y}$ and fuse it through one-layer FC and then fed to a sigmoid layer to give the final probability. In all experiments for the task, we set $\alpha = 1$. All FC layers is followed by a ReLU operations except the output layer. We train all models for a maximum of 150 epochs (training iterations) using Adam (Kingma & Ba, 2015) with a learning rate of 0.01.

- **Data imputation.** The denoise autoencoder we used in the data imputation task as described in (Gondara & Wang, 2018) consists of a dropout layer and an encoder composed of four-layer FC and a decoder composed of four-layer FC, where the dropout layer has $p = 0.1$. For discriminator d, we use three two-layer FC to encode the input $X, \hat{Y}$ and g, and we fuse the hidden embeddings by two one-layer FC and then fed to a sigmoid layer to give the final probability. In all experiments for the task, we set $\alpha = 1$. All FC layers in the discriminator are followed by ReLU operations, while all FC layers in the DAE are followed by tanh operations, with the exception of the output layer. We train all models for a maximum of 100 epochs using Adam (Kingma & Ba, 2015) with a learning rate of 10^{-5} and early stopping with a window size of 10, i.e. we stop training if the validation loss does not decrease for 10 consecutive epochs.

B ADDITIONAL ABLATION STUDY



(a) Image classification (b) Label propagation

Figure 6: Test accuracy under different weight parameter α

For the weight parameter α, We selected three different degrees of missing rates - low (30%), medium(50%) and high(80%) - under classification tasks for validation of the sensitivity of α. Figure 6 shows the classification accuracy obtained by SCL using different α values for different missing rates. We observe that the magnitude of the effect of different values of α on the accuracy is smaller for different missing rates, and even the α that leads to the worst classification accuracy is able to obtain higher accuracy than the original model.