LaSSL: Label-guided Self-training for Semi-supervised Learning

Zhen Zhao[1], **Luping Zhou**[1*], **Lei Wang**[2], **Yinghuan Shi**[3*], **Yang Gao**[3]

[1] School of Electrical and Information Engineering, University of Sydney, Australia
[2] School of Computing and Information Technology, University of Wollongong, Australia
[3] National Key Laboratory for Novel Software Technology, Nanjing University, China
{zhen.zhao, luping.zhou}@sydney.edu.au, leiw@uow.edu.au, {syh,gaoy}@nju.edu.cn

Abstract

The key to semi-supervised learning (SSL) is to explore adequate information to leverage the unlabeled data. Current dominant approaches aim to generate pseudo-labels on weakly augmented instances and train models on their corresponding strongly augmented variants with high-confidence results. However, such methods are limited in excluding samples with low-confidence pseudo-labels and under-utilization of the label information. In this paper, we emphasize the cruciality of the label information and propose a Label-guided Self-training approach to Semi-supervised Learning (LaSSL), which improves pseudo-label generations from two mutually boosted strategies. First, with the ground-truth labels and iteratively-polished pseudo-labels, we explore instance relations among all samples and then minimize a class-aware contrastive loss to learn discriminative feature representations that make same-class samples gathered and different-class samples scattered. Second, on top of improved feature representations, we propagate the label information to the unlabeled samples across the potential data manifold at the feature-embedding level, which can further improve the labelling of samples with reference to their neighbours. These two strategies are seamlessly integrated and mutually promoted across the whole training process. We evaluate LaSSL on several classification benchmarks under partially labeled settings and demonstrate its superiority over the state-of-the-art approaches. The code will be available at https://github.com/zhenzhao/lassl.

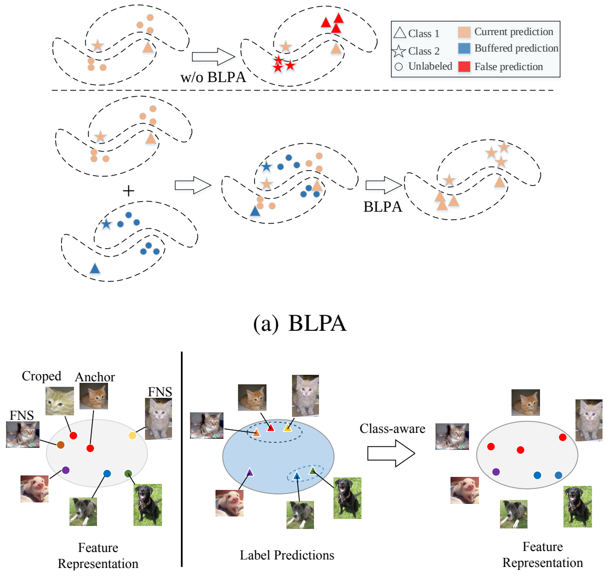
1 Introduction

In the past several years, many remarkable breakthroughs have been achieved in various computer vision tasks thanks to fast developments of deep learning (Goodfellow et al. 2016). However, such a big success is closely dependent on constructing large-scale labeled datasets which are increasingly costly and even infeasible in some professional areas (e.g., medical and astronomical fields). To mitigate the demand for labeled data, Semi-supervised learning (SSL) (Oliver et al. 2018) has been proposed as a powerful approach to leverage unlabeled data.

The principal idea of SSL is to dig guidance information for the unlabeled data and cooperate with few labeled data to train models. Current state-of-the-art (SOTA) SSL approaches, either the classic self-training-based (Lee et al. 2013; Arazo et al. 2020; Yalniz et al. 2019) or the more recent consistency-based approaches (Tarvainen and Valpola 2017; Miyato et al. 2018; Berthelot et al. 2019, 2020; Sohn et al. 2020), largely rely on the pseudo-labelling of the unlabeled data (Ouali, Hudelot, and Tami 2020). The former approaches first train the model based on the labeled data and then use the model's predictions on unlabeled data as pseudo-labels. Differently, the latter approaches usually generate two crops from a single image via data perturbations and take the prediction of one crop as the pseudo-label for the other. Such approaches commonly adopt a high-threshold mask to alleviate the confirmation bias(Arazo et al. 2020), but excluding samples with low-confidence pseudo-labels results in severe inefficiencies in exploiting unlabeled data and consumes a longer training time. More importantly, the label information in such approaches only contributes as a supervised loss, but its direct effects on pseudo-label generations are not explicitly considered.

Inspired by the observed limitations of the existing SSL approaches as above, in this paper, we propose LaSSL, a **La**bel-guided **S**elf-training approach to **S**emi-supervised **L**earning. The term "label-guided" emphasizes the full exploitation of label information based on sample relations, which is achieved by two intrinsically connected strategies aiming at improving the generation of pseudo-labels. **Firstly**, given the potential semantic content carried by ground-truth labels and pseudo-labels, LaSSL obtains the instance relations at the prediction level and explores a better feature embedding through a proposed class-aware contrastive loss, so that the same-class samples are gathered and the different-class samples are scattered. Consequently, all the unlabeled samples are involved. At the same time, better feature representations also indirectly benefit the quality of pseudo-labels. Our approach differs from the assumption of instance discrimination in contrastive learning(Jaiswal et al. 2021), where each image instance is treated as a distinct class of its own. **Secondly**, on top of the sample relations improved by the revised contrastive learning, we propagate the labels from the labeled samples to the unlabeled ones across the underlying data manifold via the label propagation algorithm (LPA) at the feature-embedding level. In this way, we could take advantage of the correlation between the labeled and unlabeled samples to improve pseudo-label generation. Since performing LPA on all unlabeled data (i.e., at the epoch level) is computationally inefficient and even infeasible on large datasets, in LaSSL, we perform label propagation at each mini-batch (i.e., at the iteration level), with the aid from the buffered outputs of the last iteration. The buffered data with high confidence are treated as labeled data in the current LPA prediction, providing more label information, while the buffered data with low confidence are treated as unlabeled data, helping explore the potential manifold. In addition, we perform the bagging technique on the buffered data to further reduce the impact of potential noise pseudo-labels. Figure 1 shows graphic explanations of these two strategies. **In summary**, better pseudo-labels make the class-aware contrastive loss more reasonable and accurate; simultaneously, the class-aware contrastive training leads to more discriminative feature representations, which in turn can be used to polish pseudo-labels via LPA at the feature-embedding level. Therefore, unlike previous works (Li, Xiong, and Hoi 2020; Iscen et al. 2019a), our proposed two strategies are tightly coupled and mutually promoted across the whole training process. This mutually boosted design is the core of LaSSL's success.

Through extensive experiments, we demonstrate that LaSSL can propose better pseudo-labels with higher quality and quantity. In specific, the class-aware contrastive loss (CACL) can quickly increase the quantity of high-confidence pseudo-labels, while the buffer-aided label propagation algorithm (BLPA) can improve the quality of pseudo-labels effectively. Experiment results show that LaSSL can outperform the SOTA SSL methods on four benchmark classification datasets with different amounts of labeled data, including CIFAR10, CIFAR100, SVHN, and Mini-ImageNet. Especially for few-label settings, LaSSL can achieve very promising accuracy, e.g., given four labels per class, LaSSL achieves an average accuracy of 95.07% on CIFAR-10 and 62.33% on CIFAR-100.



(a) BLPA

(b) CACL

Figure 1: These figures illustrate the two proposed strategies in LaSSL. (a): Buffer-aided label propagation algorithm (BLPA) utilizes the buffered labeled data to increase label information and the unlabeled data to enhance the potential manifold. Therefore, BLPA is more accurate compared to the standard distance-based labeling. (b): Typical contrastive learning (the left part) is based on the instance discrimination. Only the anchor and its augmented crop are considered similar, while the other instances are treated to be distinct classes. Obviously, there exists many false negative samples (FNS). The class-aware contrastive loss (CACL) makes full use of the label information to explore the instance relationships and make contrastive learning more reasonable.

2 Related Work

Semi-supervised learning has been researched for decades, and the essential idea is to learn from the unlabeled data to enhance the training process. Current dominant methods tend to propose pseudo-labels on unlabeled data (Ouali, Hudelot, and Tami 2020), either for self-training-based or consistency-based SSL approaches, elaborated as follows.

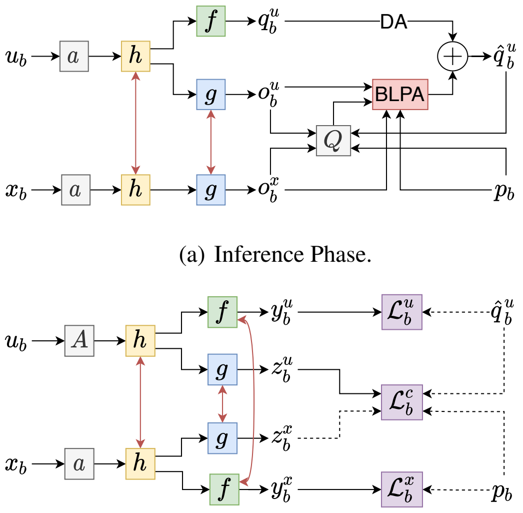
Self-training-based approaches (Lee et al. 2013; Arazo et al. 2020; McLachlan 1975; Yalniz et al. 2019) first train on the small amount of labeled data and then make predictions on unlabeled data in a form of probability distributions over the classes. Next, the unlabeled data and their corresponding pseudo-labels will be added to the labeled data if the maximal probability of the predicted pseudo-labels is higher than a predefined threshold (i.e. high confidence). After that, these approaches will train on the augmented labeled data and infer on the remaining unlabeled data, repeating the process until the model is able to make confident predictions. Some works in (Blum and Mitchell 1998; Qiao et al. 2018; Chen et al. 2018) extended the self-training from single model and single view to multiple models and multiple views, aiming to propose more confident pseudo-labels. The main weakness of such approaches is that the model cannot effectively handle wrong pseudo-labels, and the errors may quickly be accumulated, resulting in performance degradation. On the contrary, our proposed LaSSL performs self-training at the iteration level, i.e., training on both the labeled and unlabeled data within a mini-batch. Therefore the selected unlabeled data in the current iteration won't directly affect the training in the next iteration, and the potential errors won't be accumulated as before. More importantly, our proposed BLPA could further polish pseudo-labels in LaSSL.

Based on the clustering assumption (Chapelle, Scholkopf, and Zien 2009), many consistency-based SSL approaches have been proposed recently. These approaches primarily encourage invariant predictions on two perturbed inputs derived from a single image, which can also be regarded as pseudo-labelling one input for the other. Typical approaches such as Ladder Network (Rasmus et al. 2015) and

Π model (Laine and Aila 2016) applied Gaussian noise and random translation transformations to generate two different views and enforced consistency between the predictions of them. Mean Teacher (Tarvainen and Valpola 2017) highlighted the quality of pseudo-labels and introduced a weight-averaged teacher model to generate more robust targets for unlabeled data. After that, many works (Miyato et al. 2018; Verma et al. 2019; Xie et al. 2019) extensively explored various data augmentation strategies for SSL training and drew a vital conclusion that stronger and more realistic data augmentation strategies were beneficial and necessary. Holistic approaches like MixMatch (Berthelot et al. 2019), ReMixMatch (Berthelot et al. 2020) and FixMatch (Sohn et al. 2020) combined these findings and integrated other useful techniques, such as MixUp (Zhang et al. 2017), entropy minimization (Grandvalet and Bengio 2005), distribution alignment (DA) (Bridle, Heading, and MacKay 1992) into an unified framework, resulting in better performance. However, the correlation between labeled and unlabeled data and the relationship among different unlabeled data instances are ignored in these approaches.

On the other hand, recent contrastive learning studies have presented promising results to directly leverage the unlabeled data (Jaiswal et al. 2021; He et al. 2020; Chen et al. 2020a,b). Such methods exploit the similarity and dissimilarity among different data instances for representation learning, which essentially encourage similar feature representations between two random crops from the same image and distinct representations among different images. However, these approaches rely heavily on the assumption of instance discrimination, where each image instance is considered to be a distinct class. This is different from our proposed LaSSL, where we exploit the pseudo-labels and ground-truth labels to capture instance relations and construct a more reasonable class-aware contrastive loss.

There are also some recent works with similar considerations to LaSSL. S^4L (Zhai et al. 2019) integrated two pretext-based self-supervised approaches in SSL and showed that unsupervised representation learning complements existing SSL methods. SelfMatch (Kim et al. 2021) pre-trained the model on unlabeled data with SOTA self-supervised contrastive learning techniques and re-trained on the whole dataset with SSL approaches. In SIMPLE (Hu et al. 2021), a revised pair-loss was introduced to explore the relations among unlabeled samples. In contrast to these methods, our proposed LaSSL enjoys benefits from exploring wider sample relations and more label information, through injecting class-aware contrastive learning and label propagation into the standard self-training. As discussed in the literature, the quality of pseudo-labels is the key to SSL. Along this line, CoMatch (Li, Xiong, and Hoi 2020) trained two contrastive representations on unlabeled data and smoothed the pseudo-labels under the help of a large memory bank. Works in (Rizve et al. 2021; Cascante-Bonilla et al. 2020) also aimed at generating more accurate pseudo-labels by introducing uncertainty-aware selections and curriculum learning (Bengio et al. 2009), respectively. In LaSSL, on top of better feature-embedding improved by CACL, we perform BLPA to polish the pseudo-labels while leveraging the most recent outputs from the last iteration.



(a) Inference Phase.

(b) Training Phase.

Figure 2: (a): Infer on unlabeled samples and polish the pseudo-labels by BLPA under the help of labeled samples. The red two-way arrows represent "sharing weights". (b): Train model on both labeled and unlabeled data by minimizing three losses. The dash line indicates "stop gradient".

3 Method

In this section, we first introduce our proposed LaSSL at a high level and then present its components in detail. The full algorithm is shown in algorithm 1.

Overview

Unlike typical SSL approaches, in addition to the encoder $h(\cdot)$ and predictor $f(\cdot)$, LaSSL also integrates a projector $g(\cdot)$ to learn feature representations. For simplicity, we use $F(\cdot) = f \circ h(\cdot)$ for the final prediction output and $G(\cdot) = g \circ h(\cdot)$ for the final projection output. Following the standard framework of self-training, LaSSL consists of two phases, the inference phase and training phase at each iteration, as illustrated in Figure 2.

Labeled data $\mathcal{X}$ and unlabeled data $\mathcal{U}$ are given in an N-class classification task. Let (x_b, p_b) be a batch of B labeled samples and u_b be a batch of μB unlabeled samples where μ denotes the size ratio of x_b to u_b. Referring to (Sohn et al. 2020), we also introduce the weak and strong augmentations in LaSSL, denoted as $a(\cdot)$ and $A(\cdot)$, respectively, and use $H(p, q)$ to represent the cross-entropy (CE) between two distributions p and q.

Inference Phase In the inference phase, as shown in Fig. 2(a), the main task is to generate pseudo-labels on unlabeled data and the model is not updated. Different from the standard self-training, we also infer on the labeled data. Given the unlabeled u_b and labeled x_b, we can have the projection outputs $o_b^u = G(a(u_b))$ and $o_b^x = G(a(x_b))$, respectively, and the prediction output $q_b^u = F(a(u_b))$, i.e. the pseudo-label. In addition, we maintain a First-in-First-out queue, denoted by Q, which only stores the outputs from the last iteration. This is simply because the most recent predictions are more convincing during the training. To be specific, at the i-th iteration, we have $Q_i = \{(o_b, q_b)\}$ where $o_b \in \{o_b^u\} \cup \{o_b^x\}$, $q_b \in \{\hat{q}_b^u\} \cup \{p_b\}$. Correspondingly, the dequeue data at the i-th iteration will be Q_{i-1}.

At the projection head, we perform the proposed buffer-aided label propagation algorithm to jointly utilize the buffered information (Q_{i-1}), current outputs (o_b^u and o_b^x), and ground-truth labels (p_b), to generate another prediction $\widetilde{q}_b^u$, which is detailed at the following section. At the prediction head, referring to (Berthelot et al. 2020), we perform distribution alignment (DA) on the predictions of unlabeled data, $\bar{q}_b^u = \mathrm{DA}(q_b^u)$. In the operation of DA, we simply replace the uniformly moving-averaging by the exponentially moving-averaging with a decay factor of 0.99 over the historical predictions. In this way, we can not only prevent q_b^u from collapsing to certain classes but also prioritize the most current predictions. Consequently, the well-polished pseudo-labels $\hat{q}_b^u$ is obtained for the unlabeled u_b.

Training Phase The training phase is the core to update the model with three losses, a supervised CE loss $\mathcal{L}_b^x$, an unsupervised CE loss $\mathcal{L}_b^u$, and a class-aware contrastive loss (CACL) $\mathcal{L}_b^c$. As shown in Fig. 2(b), similar to the inference phase, we can obtain the prediction output y_b^x and projection output z_b^x for labeled samples, y_b^u and z_b^u for unlabeled samples. The ground-truth labels p_b and generated pseudo-labels $\hat{q}_b^u$ are used to calculated the loss $\mathcal{L}_b^x$ and $\mathcal{L}_b^u$, respectively.

$$\mathcal{L}_b^x = H(p_b, y_b^x) \tag{1}$$

$$\mathcal{L}_b^u = \mathbf{1}(\max(\hat{q}_b^u) \geq \tau)\, H(\hat{q}_b^u, y_b^u) \tag{2}$$

where $\mathbf{1}(\cdot)$ retains the pseudo-labels whose maximum probability is higher than a predefined threshold τ, i.e. high-confidence threshold. As to CACL, we first explore the instance relationship $\omega_{i,j}$ by computing the cosine similarity between their corresponding labels y_i and y_j. Specifically, we regard the different image instances as the same class if they have a high-confidence similarity, as distinct class otherwise. After that, we can minimize a class-aware contrastive loss to obtain better feature representations, so that same-class samples are gathered and the different-class samples are scattered.

Though CACL in LaSSL can help the model to make better feature representations, it has no direct effect on downstream tasks. Thus we re-weight the CACL with a ramp-down function, starting from λ_c^0 along a decreasing exponential curve. i.e., as the training progresses, we will pay more attention to classification tasks, and less attention to contrastive representation learning.

Algorithm 1: LaSSL algorithm at each iteration

```
Input: labeled data $(x_b, p_b)$, unlabeled data $u_b$, weight $\lambda_c$
Parameter: pseudo-label threshold $\tau$, similarity threshold $\varepsilon$, prediction ratio $\eta$, sampling $K$ times, weight $\lambda_u$.
Output: updated $h, f, g$.
// I. Inference Phase
obtain predictions (pseudo-labels) $q_b^u$ for $a(u_b)$
obtain smoothed predictions $\bar{q}_b^u$ via DA
obtain projections $o_b^x$ for $a(x_b)$ and $o_b^u$ for $a(u_b)$
obtain the other pseudo-labels $\widetilde{q}_b^u$ via BLPA
obtain final pseudo-labels $\hat{q}_b^u$ using Eqn. (10)
// II. Training Phase
obtain prediction $y_b^u$ and projection $z_b^u$ for $A(u_b)$
obtain prediction $y_b^x$ and projection $z_b^x$ for $a(x_b)$
calculate three losses using Eqns.( 1), (2), (12)
combine three losses with $\lambda_u$ and $\lambda_c$
back-propagate the loss and update $h, g, f$
update the EMA model
```

Buffer-aided Label Propagation Algorithm

At the i-th iteration, the dequeue data Q_{i-1} contains the feature embedding, o_{b-1}, and corresponding labels, q_{b-1}, from the last iteration. To exploit these most recent historical outputs, we regard the dequeue samples with high confidence as labeled data in the current iteration, providing more label information, while treat the dequeue samples with low confidence as unlabeled data, effectively helping explore the potential manifold. However, the samples with high-confidence labels can inevitably include errors. In order to decrease the noise, we do K random sampling with replacement on the dequeue data (i.e. bagging), and denote each sampling result as $o_{b-1}(k)$ and $q_{b-1}(k)$, where $k = 1, 2, ...K$. After that, we can split the sampling data with a predefined confidence threshold τ into two groups, the high-confidence portion ($o_{b-1}^{high}(k), q_{b-1}^{high}(k)$) and the low-confidence one ($o_{b-1}^{low}(k)$). i.e., we have

$$q_{b-1}^{high}(k) = \mathbf{1}(\max(q_{b-1}(k)) \geq \tau)\, q_{b-1}(k), \tag{3}$$

$$o_{b-1}^{high}(k) = \mathbf{1}(\max(q_{b-1}(k)) \geq \tau)\, o_{b-1}(k), \tag{4}$$

$$o_{b-1}^{low}(k) = \mathbf{1}(\max(q_{b-1}(k)) < \tau)\, o_{b-1}(k). \tag{5}$$

Combining the dequeue data with current outputs o_b^u, o_b^x and ground truth labels p_b, we can have the compound labeled features, $o_s(k) = [o_b^x, o_{b-1}^{high}(k)]$, unlabeled features, $o_t(k) = [o_b^u, o_{b-1}^{low}(k)]$, and compound label information, $q_s(k) = [p_b, q_{b-1}^{high}(k)]$.

Subsequently, a standard LPA can be applied. First, a symmetric adjacency matrix $\Omega(k)$ with zero diagonal can be constructed by calculating the similarities of $o_s(k)$ and $o_t(k)$. Then the symmetrically normalized counterpart of $\Omega(k)$ is obtained by,

$$\widetilde{\Omega}(k) = D^{-1/2}\Omega(k)D^{1/2} \tag{6}$$

where D is the degree matrix of $\Omega(k)$. After that, the label information can be iteratively propagated to the unlabeled samples. A recursive equation is,

$$\Phi_{j+1}(k) = \alpha\widetilde{\Omega}(k)\Phi_j(k) + (1-\alpha)q_s(k) \tag{7}$$

where $\Phi_j(k)$ denotes the predicted labels on compound unlabeled samples at the j-th iteration. $\alpha \in (0, 1)$ controls the amount of propagated information. In LaSSL, we use the closed-form solution (Iscen et al. 2019b) to obtain the optimal result directly,

$$\Phi^*(k) = (I - \alpha\widetilde{\Omega}(k))^{-1} q_s(k). \tag{8}$$

Since we perform LPA at the iteration-level, the computation cost is relatively small, so that BLPA can be easily scaled up to large datasets. As a result, the prediction on current unlabeled samples with the k-th sampling result can be obtained, $\phi_b(k)$, where $\phi_b(k) = \Phi^*(k)[: \mu B]$. Averaging the K results, we can have another prediction for unlabeled u_b directly from the feature-embedding level,

$$\widetilde{q}_b^u = \frac{1}{K}\sum_{k=1}^{K} \phi_b(k). \tag{9}$$

To conclude the inference phase, we eventually have the pseudo label $\hat{q}_b$ for u_b,

$$\hat{q}_b^u = \eta \widetilde{q}_b^u + (1-\eta)\bar{q}_b^u, \tag{10}$$

where η is a weight parameter to combine two predictions.

Class-aware Contrastive Loss

In the training phase, we have the projection outputs $z_b^x = G(a(x_b))$ and $z_b^u = G(A(u_b))$ for labeled and unlabeled data, respectively. Meanwhile, we have the complete label information for all the samples, i.e., the ground-truth labels p_b and the pseudo-labels $\hat{q}_b^u$. Through concatenating them together $\hat{y} = [p_b, \hat{q}_b^u]$, we can explore all the instance relationships at the prediction level,

$$\omega_{i,j} = \begin{cases} 1, & \text{if}\, i = j \\ 0, & \text{if}\, i \neq j \,\text{and}\, \hat{y}_i \cdot \hat{y}_j < \varepsilon \\ \hat{y}_i \cdot \hat{y}_j, & \text{if}\, i \neq j \,\text{and}\, \hat{y}_i \cdot \hat{y}_j \geq \varepsilon \end{cases} \tag{11}$$

where ε is a similarity threshold to determine whether two distinct instances belongs to the same class. In addition to involving the labeled samples, we can have more sense about the instance classes compared to standard contrastive learning. Therefore, with the explored instance relations at the prediction level, we design a class-aware contrastive loss,

$$\mathcal{L}_b^c = -\sum_{i=1}^{|\hat{y}|} \log \frac{\sum_{j=1}^{|\hat{y}|} \omega_{i,j} \exp(z_i \cdot z_j / T)}{\sum_{j=1, j\neq i}^{|\hat{y}|} \exp(z_i \cdot z_j / T)}. \tag{12}$$

where T is a temperature parameter (Chen et al. 2020a).

Putting it all together

In summary, the total loss at each mini-batch is,

$$\mathcal{L}_b = \mathcal{L}_b^x + \lambda_u \mathcal{L}_b^u + \lambda_c \mathcal{L}_b^c, \tag{13}$$

where λ_u and λ_c are two weight parameter for the unsupervised consistency loss and the class-aware constrastive loss, respectively. Similar to (Sohn et al. 2020), we commonly set $\lambda_u = 1.0$. However, we set λ_c as a time-variant scaling parameter to wisely control the weight of CACL. It is worth noting that, CACL aims to obtain better representations but has no direct relationship with our downstream tasks. Therefore, we emphasize CACL to improve the model at the early stages of training, togather with BLPA to enhance the accuracy of pseudo-labels. As the training progresses, we gradually focus more on downstream tasks, i.e., more on $\mathcal{L}_b^u$. To achieve this goal, we adjust λ_c in an exponentially ramping-down manner. Besides, we stop performing BLPA when the weight λ_c becomes small. It is simply because BLPA relies upon the better representations derived from CACL. Mathematically, referring to (Laine and Aila 2016), given the total training epochs T_t and the ramp-down length $(T_t - T_r)$, the weight λ_c at the t-th epoch can be calculated as,

$$\lambda_c = \begin{cases} \lambda_c^0, & \text{if}\, t \leq T_r, \\ \lambda_c^0 \exp\left(-\frac{(t-T_r)^2}{2(T_t - T_r)}\right), & \text{otherwise}. \end{cases} \tag{14}$$

where λ_c^0 is set as the maximum value of λ_c. As a result, the whole training process of LaSSL can be treated as two different periods: it first exploits CACL and BLPA to update the model quickly, and then improve the model further by emphasizing downstream tasks. To further simplify the training, we stop applying CACL and BLPA when $\lambda_c \leq \hat{\lambda_c}$. These two parameters T_r and $\hat{\lambda_c}$, can affect how long the CACL and BLPA will be involved across the training process. Besides, following FixMatch and ReMixMatch, an exponential moving average (EMA) of model parameters with decay of 0.999 is utilized to produce more stable predictions.

4 Experiments

In this section, we conduct experiments on four classification datasets to test the effectiveness of LaSSL, including CIFAR-10 (Krizhevsky and Hinton 2009), CIFAR-100 (Krizhevsky and Hinton 2009), SVHN (Netzer and Wang 2011) and Mini-Imagenet (Ravi and Larochelle 2017). Following the standard protocol in SSL, we randomly select certain number of labeled data from the training set and treat the remaining training data as unlabeled data. The mean and standard deviation of five runs on testing set with different random seeds are reported. By default, we use a Wide ResNet-28-2 as the encoder $h(\cdot)$, one linear layer as the predictor $f(\cdot)$, and a 2-layer MLP as the projector $g(\cdot)$. The default settings for hyper-parameters in LaSSL is $B = 64, \mu = 7, K = 7, \alpha = 0.8, \eta = 0.2, \tau = 0.95, \varepsilon = 0.7, T_t = 512, \lambda_c^0 = 1.0, T_r = 30, \hat{\lambda_c} = 0.1$. Besides, we adopt a SGD optimizer with a momentum of 0.9 and a weight decay of 5e-4, and use a learning rate scheduler with cosine decay to train the model. Unless otherwise noted, we use same codebase and parameter settings to run experiments.

CIFAR-10, CIFAR-100, and SVHN

CIFAR-10 and CIFAR-100 are labeled subsets of the 80 million tiny images dataset with 10 and 100 classes, respectively. Both of them contain 50000 32x32 training images and 10000 32x32 testing images. For fair comparisons, we use Wide ResNet-28-2 as the backbone for CIFAR-10 and Wide ResNet-28-8 for CIFAR-100. In Table 1, we compare the testing accuracy of LaSSL against recent SOTA SSL approaches with a varying number of labeled samples. We can obviously see that our LaSSL consistently outperforms

	CIFAR-10		CIFAR-100		SVHN	
Methods	40 labels	250 labels	400 labels	2500 labels	40 labels	250 labels
Π-Model*	-	45.74±3.87	-	42.75±0.48	-	81.04±1.92
Pseudo-label*	-	50.22±0.43	-	42.62±0.46	-	79.79±1.09
Mean-Teacher*	-	67.68±2.30	-	46.09±0.57	-	96.43±0.11
MixMatch*	52.46±11.50	88.95±0.86	33.39±1.32	60.06±0.37	57.45±14.53	96.02±0.23
UDA*	70.95±5.93	91.18±1.08	40.72±0.88	66.87±0.22	47.37±20.51	94.31±2.76
ReMixMatch*	80.90±9.64	94.56±0.05	55.72±2.06	72.57±0.31	96.64±0.30	97.08±0.48
FixMatch*	86.19±3.37	94.93±0.65	51.15±1.75	71.71±0.11	96.04±2.17	97.52±0.38
ACR†	92.38	95.01	-	-	-	-
SelfMatch†	93.19±1.08	95.13±0.26	-	-	96.58±1.02	97.37±0.43
CoMatch†	93.09±1.39	95.09±0.33	-	-	-	-
Dash†	86.78±3.75	95.44±0.13	55.24±0.96	72.82±0.21	**96.97±1.59**	97.83±0.10
LaSSL	**95.07± 0.78**	**95.71 ±0.46**	**62.33±2.69**,	**74.67± 0.65**	96.91±0.52	**97.85± 0.13**

Table 1: Top-1 testing accuracy (%) for CIFAR-10, CIFAR-100 and SVHN on 5 different folds. All the related works are sorted by their publication date. Results with * was reported in FixMatch (Sohn et al. 2020), while results with † comes from the most recent papers (Kim et al. 2021; Li, Xiong, and Hoi 2020; Xu et al. 2021; Abuduweili et al. 2021), respectively.

other SOTA approaches on CIFAR-10 and CIFAR-100 under all settings. Especially when considering situations with very few labeled data, LaSSL improves over other SSL approaches by a large margin, e.g. achieving an average testing accuracy of 95.07% on CIFAR-10 with only 40 labels. When the number of classes is large like CIFAR-100, LaSSL can still perform well and achieve a accuracy gain of around 7% over the SOTA approach given four labels per class. Checking more details, we find that, achieving the accuracy of around 95% on CIFAR-10, LaSSL needs only four labels per class while other SSL approaches requires 25 or more labels per class. Obviously, LaSSL is more sample efficient and shows its great potential for label-scarce scenarios.

SVHN consists of 10-class colorful 32x32 house numbers. It has 73257 training images and 26032 testing images. The testing accuracy on SVHN in Table 1 also shows comparable results to recent state-of-the-art results achieved by Remixmatch and Dash. We can see that the results of all of recent SSL approaches on SVHN are close to the fully supervised baseline (97.3% (Hu et al. 2021)) with less than 1% difference. Though its superior is not apparent in such simple dataset, LaSSL can achieve the SOTA performance on SVHN with 250 labeled samples. Compared to Dash (Abuduweili et al. 2021) on SVHN with 40 labeled samples, LaSSL performs slightly worse in terms of the average accuracy but can achieve a lower variance.

Mini-ImageNet

Following the SIMPLE (Hu et al. 2021), we test LaSSL on more complicated dataset, Mini-ImageNet. It is initially designed for few-shot learning in (Ravi and Larochelle 2017). Sampled from ImageNet ILSVRC, it consists of 50000 training images and 10000 testing images, evenly distributed across 100 classes. We compare the performance of LaSSL against the SOTA SSL approach, SIMPLE, on Mini-ImageNet with 4000 labeled samples. For a fair comparison, ResNet-18 is set as the backbone, and each sample is center-cropped and resized to 84x84. Apart from default parameter configurations, we set $\lambda_c^0 = 5.0, \tau = 0.8$ in this experiment. SIMPLE can achieve an average testing accuracy of **49.39**%, while LaSSL obtains a result of **60.14** ± 0.26 %. LaSSL can obviously outperform SIMPLE with a better accuracy by a significant average gain of 10.75%.

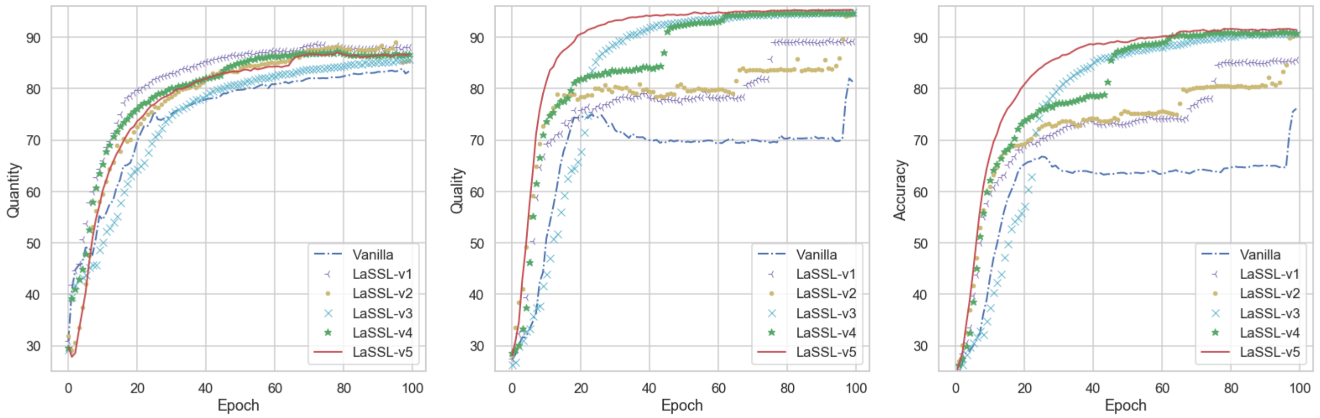
Method	CACL	BLPA	DA	Quantity	Quality	Accuracy
Vanilla	✗	✗	✗	83.91	81.98	75.54
LaSSL-v1	✓	✗	✗	88.66	89.38	85.50
LaSSL-v2	✓	✓	✗	89.08	94.31	90.24
LaSSL-v3	✗	✗	✓	85.73	94.90	90.42
LaSSL-v4	✓	✗	✓	87.46	94.89	91.11
LaSSL-v5	✓	✓	✓	87.03	**95.33**	**91.65**

Table 2: Ablation studies on CIFAR-10 with 40 labeled data after training 100 epochs (random seed is fixed to 1.)

Ablation Study

Effectiveness of different components To investigate the impact of three different components in LaSSL (i.e., CACL, BLPA and DA), we test LaSSL with different combinations of these components on CIFAR-10 with four labels per class. For fair comparisons, we compare their performance with the same random seed during the first 100 epochs. To better analyze the performance, we introduce two intuitive concepts, quantity and quality of pseudo-labels. "**Quantity**" refers to the amount of high-confidence pseudo-labels, calculated by the ratio of the number of high-confidence predictions to the total number of unlabeled samples. "**Quality**" measures how many high-confidence predictions are consistent to ground-truth labels, where we use the real labels of unlabeled data from CIFAR-10 to calculate the accuracy.

It can be seen from Table 2 that every component matters



(a) quantity

(b) quality

(c) accuracy

Figure 3: (a), (b), (c) represent curves of the quantity, quality, and EMA test accuracy of different combinations of CACL, BLPA, and DA (better view on screen). Numerical results are listed in Table 2.

ε	0.6	0.7	0.8	0.9	1.0
Accuracy(%)	87.64	**89.39**	87.70	87.36	85.17

Table 3: Effects with different similarity thresholds. The similarity is equal to 1 only when comparing the image instance with itself. Therefore, we use $\varepsilon = 1.0$ to investigate the effect of excluding the “class-aware” technique.

K	0	1	3	5	7
Accuracy(%)	92.71	92.10	94.64	93.43	**94.87**

Table 4: Effects with different number of samplings. In specific, $K = 0$ means the plain LPA without “buffer-aided”; $K = 1$ means exploiting the buffered data directly without sampling; while $K > 1$ investigates the complete BLPA.

compared to the vanilla version. Integrating all three components can achieve the best accuracy and highest quality while maintaining a considerably high quantity. In Figure 3, we show the detailed dynamics of the quantity, quality and accuracy w.r.t training epochs. We can observe from Figure 3(a) that LaSSL-v1 can consistently achieve the highest quantity in the 100 epochs, indicating that the CACL is very effective in quickly improving the number of high-confidence pseudo-labels. By comparing the LaSSL-v1 to LaSSL-v2, the vanilla to LaSSL-v3, we can find that BLPA and DA are two powerful strategies to improve the quality. Besides, the dynamics of Figure 3(b) and 3(c) are closely related, suggesting that the quality of pseudo-labels is the most crucial factor affecting the final performance. The increasing tendency also indicates that LaSSL-v5 (i.e., standard LaSSL) is the most stable and accurate one with the consistently highest testing accuracy.

Impact of different similarity threshold In Table 3, we compare the effect of CACL with different values of similarity threshold in terms of the testing accuracy. For fair comparisons, BLPA and DA are not involved. Since the similarity can never exceed 1.0, $\varepsilon = 1.0$ simply denotes that every instance belongs to a distinct class, i.e., without class-aware senses. We can observe that the “class-aware” strategy is indeed beneficial in SSL. Besides, there intuitively exists a trade-off, i.e. lower values of ε can involve more similarities among samples but inevitably introduce more errors. In contrast, large ε may fail exploiting the instance relations.

Impact of different sampling times We investigate the impact of BLPA with different values of K in Table 4. For fair comparisons, we adopt default settings for CACL and DA. To reduce effects of wrong pseudo-labels, we sample K-times on the buffered data and average the results in BLPA. $K = 0$ means no buffer-aided, while $K = 1$ uses all the buffer data without sampling. As a result, $K = 7$ achieve the highest testing accuracy. We can also find that directly involving all the buffered data (i.e. $K = 1$) will degrade the performance due to introducing more wrong high-confidence pseudo-labels. On the other hand, though large K may introduce more computational efforts, it can generally lead to more robust predictions and higher accuracy.

5 Conclusion

In this paper, we propose LaSSL, a novel SSL approach that exploits the label information to integrate a class-aware contrastive loss and buffer-aided label propagation algorithm into a self-training paradigm. The proposed strategies are tightly coupled and mutually boosted across the training process. Meanwhile, the label information is extensively utilized: to train models as a supervised loss, to generate instance relations for CACL, and to be propagated on unlabeled samples in BLPA. Experiment results show that LaSSL can effectively improve the pseudo-labels generations in terms of quantity and quality, resulting in better performance over other SSL approaches.

References

Abuduweili, A.; Li, X.; Shi, H.; Xu, C.-Z.; and Dou, D. 2021. Adaptive Consistency Regularization for Semi-Supervised Transfer Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6923–6932.

Arazo, E.; Ortego, D.; Albert, P.; O'Connor, N. E.; and McGuinness, K. 2020. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning (ICML)*, 41–48.

Berthelot, D.; Carlini, N.; Cubuk, E. D.; Kurakin, A.; Sohn, K.; Zhang, H.; and Raffel, C. 2020. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. In *8th International Conference on Learning Representations (ICLR)*.

Berthelot, D.; Carlini, N.; Goodfellow, I.; Papernot, N.; Oliver, A.; and Raffel, C. A. 2019. MixMatch: A Holistic Approach to Semi-Supervised Learning. In *Advances in Neural Information Processing Systems*.

Blum, A.; and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the 17th annual conference on computational learning theory*, 92–100.

Bridle, J. S.; Heading, A. J.; and MacKay, D. J. 1992. Unsupervised Classifiers, Mutual Information and 'Phantom Targets'. In *Advances in Neural Information Processing Systems, NIPS Conference, Denver, Colorado, USA, December 2-5, 1991*.

Cascante-Bonilla, P.; Tan, F.; Qi, Y.; and Ordonez, V. 2020. Curriculum Labeling: Revisiting Pseudo-Labeling for Semi-Supervised Learning. *arXiv preprint arXiv:2001.06001*.

Chapelle, O.; Scholkopf, B.; and Zien, A. 2009. Semi-supervised learning [book reviews]. *IEEE Transactions on Neural Networks*, 20(3): 542–542.

Chen, D.; Wang, W.; Gao, W.; and Zhou, Z. 2018. Tri-net for semi-supervised deep learning. In *International Joint Conferences on Artificial Intelligence*.

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

Chen, T.; Kornblith, S.; Swersky, K.; Norouzi, M.; and Hinton, G. 2020b. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*.

Goodfellow, I.; Bengio, Y.; Courville, A.; and Bengio, Y. 2016. *Deep learning*. MIT press Cambridge.

Grandvalet, Y.; and Bengio, Y. 2005. Semi-supervised learning by entropy minimization. In *CAP*, 281–296.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9729–9738.

Hu, Z.; Yang, Z.; Hu, X.; and Nevatia, R. 2021. SimPLE: Similar Pseudo Label Exploitation for Semi-Supervised Classification. *arXiv preprint arXiv:2103.16725*.

Iscen, A.; Tolias, G.; Avrithis, Y.; and Chum, O. 2019a. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5070–5079.

Iscen, A.; Tolias, G.; Avrithis, Y.; and Chum, O. 2019b. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5070–5079.

Jaiswal, A.; Babu, A. R.; Zadeh, M. Z.; Banerjee, D.; and Makedon, F. 2021. A survey on contrastive self-supervised learning. *Technologies*, 9(1): 2.

Kim, B.; Choo, J.; Kwon, Y.-D.; Joe, S.; Min, S.; and Gwon, Y. 2021. SelfMatch: Combining Contrastive Self-Supervision and Consistency for Semi-Supervised Learning. *arXiv preprint arXiv:2101.06480*.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 1(4).

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

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*.

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

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

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

Netzer, Y.; and Wang, T. 2011. Reading Digits in Natural Images with Unsupervised Feature Learning. *nips workshop on deep learning and unsupervised feature learning*.

Oliver, A.; Odena, A.; Raffel, C.; Cubuk, E. D.; and Goodfellow, I. J. 2018. Realistic evaluation of deep semi-supervised learning algorithms. *arXiv preprint arXiv:1804.09170*.

Ouali, Y.; Hudelot, C.; and Tami, M. 2020. An Overview of Deep Semi-Supervised Learning. *arXiv preprint arXiv:2006.05278*.

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

Rasmus, A.; Valpola, H.; Honkala, M.; Berglund, M.; and Raiko, T. 2015. Semi-supervised learning with ladder networks. *arXiv preprint arXiv:1507.02672*.

Ravi, S.; and Larochelle, H. 2017. Optimization as a model for few-shot learning. In *5th International Conference on Learning Representations (ICLR)*.

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. *arXiv preprint arXiv:2101.06329*.

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

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

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

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

Xu, Y.; Shang, L.; Ye, J.; Qian, Q.; Li, Y.-F.; Sun, B.; Li, H.; and Jin, R. 2021. Dash: Semi-Supervised Learning with Dynamic Thresholding. In *International Conference on Machine Learning (ICML)*, 11525–11536.

Yalniz, I. Z.; Jégou, H.; Chen, K.; Paluri, M.; and Mahajan, D. 2019. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*.

Zhai, X.; Oliver, A.; Kolesnikov, A.; and Beyer, L. 2019. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 1476–1485.

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