A Unified Contrastive Loss for Self-Training

Aurélien Gauffre, Julien Horvat, Massih-Reza Amini

A Unified Contrastive Loss for Self-Training

Aurélien Gauffre, Julien Horvat, and Massih-Reza Amini

Université Grenoble Alpes, CNRS, Laboratoire d'Informatique de Grenoble
38401 Grenoble, France
{Firstname.Lastname@univ-grenoble-alpes.fr}

Abstract. Self-training methods have proven to be effective in exploiting abundant unlabeled data in semi-supervised learning, particularly when labeled data is scarce. While many of these approaches rely on a cross-entropy loss function (CE), recent advances have shown that the supervised contrastive loss function (SupCon) can be more effective. Additionally, unsupervised contrastive learning approaches have also been shown to capture high quality data representations in the unsupervised setting. To benefit from these advantages in a semi-supervised setting, we propose a general framework to enhance self-training methods, which replaces all instances of CE losses with a unique contrastive loss. By using class prototypes, which are a set of class-wise trainable parameters, we recover the probability distributions of the CE setting and show a theoretical equivalence with it. Our framework, when applied to popular self-training methods, results in significant performance improvements across four different datasets with a limited number of labeled data. Additionally, we demonstrate further improvements in convergence speed, transfer ability, and hyperparameter stability.

Keywords: Semi-Supervised Learning · Contrastive Learning · Classification · Self-Training

1 Introduction

Semi-supervised learning benefits significantly from advances in unsupervised representation learning, particularly through self-supervised approaches, which excel at efficiently extracting information from unlabeled data. Among these approaches, contrastive learning [9, 18, 19, 27] has been particularly effective in the field of computer vision. Moreover, contrastive learning has been shown not limit its application to unsupervised settings. The standard practice for training deep neural networks in a supervised setting has traditionally involved using cross-entropy (CE) as the primary loss function. In recent works, [21] have developed a supervised contrastive loss function, dubbed *SupCon*, which achieves highly discriminative representations and comparable or even superior results in accuracy. It uses information from the labels to create positive pairs, instead of relying on data augmentation to generate two different views of the same unlabeled sample. Specifically, positive instances (instances from the same class within a batch) are pushed closer together while pushing them away from

negative instances (instances from other classes) in the embedding space. Recent research suggests that SupCon loss may have the potential to increase robustness and be less sensitive to various hyperparameter choices for data augmentation or optimizers [17, 20, 21].

However, while SupCon hinges on the presence of labeled training data, its unsupervised counterpart cannot leverage any label information. The primary objective of this study is to adapt the principles and advantages of SupCon to semi-supervised scenarios. Through the integration of both supervised and unsupervised aspects of contrastive learning, we introduce a Semi-Supervised Contrastive (SSC) framework which uses a single loss $\mathcal{L}_{\text{SSC}}$. Our approach enables the integration of existing self-training techniques such as FixMatch [29], allowing for a seamless transition between unsupervised and supervised paradigms.

Unlike CE loss trained with softmax activation function, contrastive loss does not provide directly a probability distribution needed to pseudo-label examples during self-training. To address this challenge, we propose a solution by introducing class prototypes and we establish a theoretical equivalence between classical cross-entropy and supervised contrastive learning with these prototypes. The main contributions of this work are threefold:

- We propose a new framework for semi-supervised learning based on a Semi-Supervised Contrastive loss $\mathcal{L}_{\text{SSC}}$ that handles labeled, pseudo-labeled and unconfident pseudo-labels examples at the same time.
- We show how to integrate class prototypes and establish a theoretical bridge between cross-entropy and supervised contrastive learning with prototypes.
- We apply our loss to FixMatch, a simple existing framework, and show significant improvement on three datasets and investigate the properties of our loss function, highlighting its faster convergence rate, adaptability to transfer learning and its stability to hyperparameters.

In the following, we begin by presenting the notations and background in Section 2. Next, in Section 3, we introduce our proposed approach. Then, we discuss the experiments carried out on three benchmarks in Section 4. Lastly, Section 5 presents our conclusions.

2 Notations and Background

Notations. We will now introduce necessary notations and then show how they connect to previous related works. We use matrix notation rather than vectors, which provides a more convenient framework for presenting our approach. In the semi-supervised context, the batch is divided into a matrix $\boldsymbol{X}$ consisting of B labeled examples and their associated label vector $\boldsymbol{y}^x$, and another matrix $\boldsymbol{U}$ containing μB unlabeled examples, where the integer μ denotes the factor size between $\boldsymbol{X}$ and $\boldsymbol{U}$. More specifically, we have :

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^\top \\ \vdots \\ \boldsymbol{x}_B^\top \end{bmatrix} \in \mathcal{X}^B, \quad \boldsymbol{y}^x = \begin{bmatrix} y_1 \\ \vdots \\ y_B \end{bmatrix} \in [1, ..., K]^B, \quad \boldsymbol{U} = \begin{bmatrix} \boldsymbol{u}_1^\top \\ \vdots \\ \boldsymbol{u}_{\mu B}^\top \end{bmatrix} \in \mathcal{U}^{\mu B}$$

We denote by $f : \mathcal{X} \cup \mathcal{U} \rightarrow \mathcal{Z}$, an encoder that maps examples into the hidden space $\mathcal{Z} \subseteq \mathbb{R}^h$. Then, a projection head p maps embeddings into a probability distribution over the K classes of our classification problem. In the context of self-training, *pseudo-labels* refers to labels automatically assigned to unlabeled data by a model's highest confidence prediction, defined as the argmax on the projection head p. The model is said to be confident in an unlabeled example if the maximum probability exceeds a threshold τ.

Following most of the recent semi-supervised learning approaches based on consistency regularization, we employ both a weak data augmentation, denoted as $\alpha(.)$, and a strong augmentation, denoted as $\mathcal{A}(.)$. During training, the encoder f is trained to compute three distinct embeddings:

- $\boldsymbol{Z}^x = f(c) = [\boldsymbol{z}_1^x, \ldots, \boldsymbol{z}_B^x]^\top \in \mathbb{R}^{B \times d}$, is the supervised embedding generated from the labeled training data.
- $\boldsymbol{Z}^u = \begin{bmatrix} \boldsymbol{Z}^{s1} \\ \boldsymbol{Z}^{s2} \end{bmatrix} = \begin{bmatrix} f(\mathcal{A}(\boldsymbol{U})) \\ f(\mathcal{A}(\boldsymbol{U})) \end{bmatrix} \in \mathbb{R}^{2\mu B \times d}$ denote the embeddings produced by applying two stochastic strong augmentation to unlabeled data. Using two augmentations ensures at least one positive pair for each example in the batch.
- $\boldsymbol{Z}^w = f(\alpha(\boldsymbol{U})) = [\boldsymbol{z}_1^w, \ldots, \boldsymbol{z}_{\mu B}^w]^\top \in \mathbb{R}^{\mu B \times d}$ is the unsupervised embedding created through the application of weak data augmentation. This embedding is employed for the estimation of a confidence score and the generation of pseudo-labels in self-training approaches.

Finally, we define two set of labels associated to unlabeled examples :

- $\boldsymbol{y}^{u\uparrow} = \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{q} \end{bmatrix}$ where $\boldsymbol{q} = \arg\max p(\boldsymbol{Z}^w)$ are the pseudo-labels computed with the weak-augmented examples. They are associated to unlabeled data with high confidence, above a given threshold τ.
- $\boldsymbol{y}^{u\downarrow} = \begin{bmatrix} \boldsymbol{i} \\ \boldsymbol{i} \end{bmatrix}$ where $\boldsymbol{i} = [1, 2, ..., \mu B]^\top$ are the labels associated to unlabeled examples τ. This definition will ensures that these unconfident labels will only have a unique positive example associated with them.

We will briefly present the contrastive and semi-supervised learning losses of related work with the previous notation, and then show how our approach is connected to these losses functions.

Supervised Contrastive Learning. In [21], labeled data is utilized to ensure that embeddings of samples with identical labels are pulled closer together, while ensuring that embeddings from samples with different labels are pushed farther apart. This is achieved by employing supervised embeddings $\mathbf{Z}^x$ along with their corresponding labels $\mathbf{y}^x$.

The objective involves calculating, for each embedding $\mathbf{z}_i^x$ (referred to as the anchor), its cosine similarity with all other embeddings $\mathbf{z}_p^x$ that share the same label (referred to as positive pairs). Subsequently, this similarity is normalized

by the sum of similarities across all pairs, following the principles of the classical InfoNCE loss [27]. More precisely, we have:

$$\mathcal{L}_{\text{SupCon}}(\boldsymbol{Z}^x, \boldsymbol{y}^x) = \frac{1}{|I|} \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in \mathcal{P}(i)} \log \left(\frac{\exp(\boldsymbol{z}_i^x \cdot \boldsymbol{z}_p^x / T)}{\sum_{j \in I \setminus \{i\}} \exp(\boldsymbol{z}_i^x \cdot \boldsymbol{z}_j^x / T)} \right) \quad (1)$$

Where $I = \{1, ..., B\}$ is the set of anchor indices, $P(i) = \{p \in I \setminus \{i\} : y_p^x = y_i^x\}$ is the set of positive examples associated with the example i and T is a temperature hyperparameter. Note that the labels $\boldsymbol{y}^x$ are used in the equation only to define of the positive pairs $\mathcal{P}$.

Unsupervised Contrastive Learning. As seen in methods SimCLR [9] or MoCo [19], unsupervised contrastive learning relies on two strong augmentations for each instance within the unsupervised dataset $\boldsymbol{U}$, and employs the InfoNCE loss [27].

Unlike SupCon, which utilizes explicit labels to identify positive and negative samples, self-supervised losses operate under an unsupervised paradigm where labels are not provided. Consequently, in self-supervised learning, every augmented sample has only one positive pair, effectively constituting a specialized form of the SupCon loss. Based on the previous definition of $\boldsymbol{y}^{u\downarrow}$, the self-supervised InfoNCE loss can be expressed simply as:

$$\mathcal{L}_{\text{Self}}(\boldsymbol{Z}^u) = \mathcal{L}_{\text{SupCon}}(\boldsymbol{Z}^u, \boldsymbol{y}^{u\downarrow}) \quad (2)$$

Interpreting the unsupervised contrastive loss as a specific instance of SupCon is central to the design of our unified loss.

Self-training [2]. Also commonly referred to as pseudo-labeling, self-training is a wrapper algorithm that is widely adopted in recent state-of-the-art semi-supervised learning approaches [4,5,7,8,10,24,29,38]. A classifier is first trained on the labeled training data, and then assigns iteratively pseudo-labels to unlabeled data and retrain the classifier with the augmented training set. Some approaches propose to use self-training in an online manner [4,5,29].

More specifically, FixMatch [29] apply a CE loss $\mathcal{L}_x$ to labeled examples X whereas an extra unsupervised CE loss $\mathcal{L}_u$ is applied to unlabeled training examples with their associated pseudo-labels, only if the model confidence exceeds the threshold τ :

$$\mathcal{L}_x = \frac{1}{B} \sum_{i=1}^{B} H(p(\boldsymbol{z}_i^x), y_i^x) \quad (3)$$

$$\mathcal{L}_u = \frac{1}{\mu B} \sum_{i=1}^{\mu B} H(p(\boldsymbol{z}_i^u), y_i^{u\uparrow}) \mathbb{1}(\max p(\boldsymbol{z}_i^w) > \tau) \quad (4)$$

This unsupervised loss is based on consistency regularization principle, which enforces the model to become invariant to perturbations of the input, like strong augmentations. It has become central to many popular recent semi-supervised approaches in computer vision [22, 30, 34].

Recently, adaptive thresholding strategies for generating pseudo-labels have been proposed in Dash [35], FlexMatch [37], Adamatch [6], and FreeMatch [33]. SoftMatch [8] proposes to adjust pseudo-labels contributions based on their confidence levels by learning a parametric density function that adaptively assigns weights for each pseudo-labeled examples.

On the other hand, CoMatch [24] and SimMatch [38] introduce an additional contrastive loss that enforce similarity between representations having similar probability distribution. Other existing semi-supervised approaches [23, 31] have already proposed to use the SupCon loss, as an extra regularization term applied to labeled or pseudo-labeled examples. Other than the self-training techniques we mentioned, very successful semi-supervised learning exist and often rely on self-supervised principles. This may involve using an additional regularization loss as in S4L [7], or using contrastive pre-training combined with distillation as in SimCLR V2 [10], or using clustering approaches like PAWS [3] or Suave and Daino [15].

In contrast to all the aforementioned approaches, our method uses a single contrastive loss that handles both the labeled training data and all the unlabeled training examples at the same time, including those on which the model is not confident.

3 Method

Our approach is a wrapper algorithm that can be easily adapted to various self-training algorithms. We will use FixMatch as an example to illustrate our approach because of its simplicity. However, our proposed approach is flexible enough to be applied to more complex self-training algorithms.

3.1 Overview

In our approach, we aim to enhance the classical SupCon loss by integrating labeled, pseudo-labeled, and unlabeled examples on which the model is unconfident, simultaneously within the loss formulation. The fundamental architecture of our method is illustrated in Figure 1.

Using the encoder f, we first compute $\boldsymbol{Z}^x$, the embeddings of the labeled training data. In a similar way, we generate $\boldsymbol{Z}^u$ by applying two strong data augmentations to the unlabeled training data, and $\boldsymbol{Z}^w$ by applying a weak data augmentation.

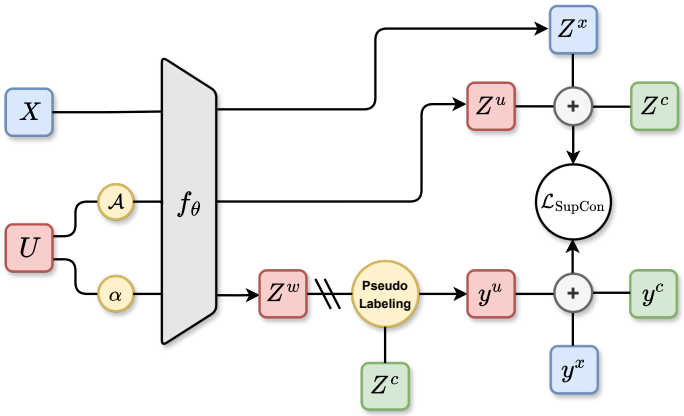

Fig. 1: **SSC framework**. $\boldsymbol{Z}^x$, $\boldsymbol{Z}^u$, and $\boldsymbol{Z}^c$ are supervised, unsupervised, and prototype embeddings and $\boldsymbol{y}^x$, $\boldsymbol{y}^u$, and $\boldsymbol{y}^c$ their corresponding labels, which aim to define positive pairs in the loss. The weakly augmented embeddings $\boldsymbol{Z}^w$ are used only during pseudo-labeling phase to compute $\boldsymbol{y}^u$ and does not propagate gradient back. Strongly augmented embeddings $\boldsymbol{Z}^u$ used two augmentations to ensure the existence of at lease one positive pair for unconfident examples.

Unsupervised part $\boldsymbol{Z^u, y^u}$. A pivotal innovation allowed in our framework is its way to handle all unlabeled examples :

$$y_i^u = \begin{cases} y_i^{u\uparrow} & \text{if } \max p(\boldsymbol{z}_i^w) > \tau \\ y_i^{u\downarrow} + K & \text{otherwise.} \end{cases} \tag{5}$$

Concerning high confidence examples, we adopt a strategy similar to online self-training methods, like FixMatch, by using pseudo-labels previously defined as $\boldsymbol{y}^{u\uparrow}$. However, rather than disregarding examples that have a posterior probability below the threshold τ, we assign unique labels to them using $\boldsymbol{y}^{u^\downarrow}$. Note that to make sure these labels are unique, values are shift by K to not interfere with existing classes.

This leads to the creation of singular positive pairs, mirroring the mechanics of unsupervised contrastive loss methods such as SimCLR as shown in equation 2. By incorporating both confident and unconfident examples within y^u, our method is able to leverages all unlabeled training data. Note that, even if the loss does not directly depend on the weakly augmented embeddings, Z^w is used to compute y^u.

Centroid part $\boldsymbol{Z^c, y^c}$. Computing y^u requires a projection head p that maps $\boldsymbol{Z}^w$ into a distribution probability. However, training a model with a supervised contrastive loss does not produce directly such a classifier, which is why an extra training phase with cross-entropy is employed [21].

In order to address this issue, and to maintain a fully contrastive framework, we propose the use of class prototypes [14,16,39]. It consists in using K trainable parametric centers $\mathbf{Z}^c \in \mathbb{R}^{K \times h}$ that lie directly in the embeddings space. We define the label prototypes as $\boldsymbol{y}^c = [1, 2, ..., K]^\top$ so that the k^{th} row of $\mathbf{Z}^c$ represents the prototype associated with class k. These parameters, initiated randomly, are then updated throughout the training process similarly to all other embeddings. A novel aspect of our method is to use these prototypes to define a probability distribution for a weakly augmented example $\mathbf{z}_i^w$, by applying a softmax function with temperature T' to its cosine similarity with all prototypes:

$$p(\boldsymbol{z}_i^w) := \text{softmax}(\frac{\boldsymbol{Z}^c \boldsymbol{z}_i^w}{T'}) \tag{6}$$

Training with these prototypes allows defining a classification head p, used to compute y^u, without the addition of an extra cross-entropy loss. Further analysis and a connection with the cross entropy are discussed below.

$\mathcal{L}_{SSC}$; a unified loss Finally, our loss, denoted as Semi-Supervised Contrastive (SSC) loss, can be easily expressed using SupCon and previously defined quantities:

$$\mathcal{L}_{\text{SSC}} = \mathcal{L}_{\text{SupCon}} \left(\begin{bmatrix} \boldsymbol{Z}^x \\ \boldsymbol{Z}^u \\ \boldsymbol{Z}^c \end{bmatrix}, \begin{bmatrix} \boldsymbol{y}^x \\ \boldsymbol{y}^u \\ \boldsymbol{y}^c \end{bmatrix} \right) \tag{7}$$

Table 1, provides an overview of different state-of-the-art approaches that utilize labeled, pseudo-labeled and unconfident unlabeled data in their learning process.

Example Types	$\boldsymbol{Z}^x$	$\boldsymbol{Z}^{u\uparrow}$	$\boldsymbol{Z}^{u\downarrow}$
Standard Classification	CE	∅	∅
Supervised Contrastive [21]	CL	∅	∅
Unsupervised Contrastive (e.g. [9,19,27])	∅	∅	CL
FixMatch, FlexMatch [29,37]	CE	CE	∅
Fixmatch w. CR [23]	CE	CE	CL
CoMatch, SimMatch [24,38]	CE	CE + CL	∅
DualMatch [31]	CE + CL	CE + CL	∅
Ours	CL	CL	CL

Table 1: Comparison of loss types used in various online self-training algorithms. The table indicates which type of loss, either CE (Cross-Entropy), Contrastive Learning (CL), or none (∅) is applied to different parts of the input: $\boldsymbol{Z}^x$ for embeddings of supervised examples, $\boldsymbol{Z}^{u\uparrow}$ for high-confidence pseudo-labeled examples, and $\boldsymbol{Z}^{u\downarrow}$ for unconfident examples (confidence less than threshold τ).

Algorithm 3.1: Semi-Supervised Contrastive (SSC) Pseudocode

```
# Aug: strong augment, aug: weak augment
# T': softmax temperature for pseudo-labeling,
# tau : threshold for pseudo-labeling
# lambda : weight
def training_step(X, U, y_x, prototypes):
    # compute embeddings
    Z_x = f(X),
    Z_u = cat(f(Aug(U), f(Aug(U)))
    Z_w = f(aug(U))
    Z_c = prototypes
    # compute target using equation 5.
    y_u = compute_pseudo_labels(Z_w,Z_c,tau,T')
    y_c = [1,...,K]
    # compute loss
    Z = cat(Z_x, Z_u, Z_c)
    y = cat(y_x, y_u, y_c)
    loss = supcon_loss(Z, y, lambda)
    return loss
```

Comparatively, our proposed approach is the only one which takes advantage of all labeled and unlabeled training data for learning in a fully contrastive framework.

The pseudo-code of the proposed approach is provided in Algorithm 1. First, the algorithm computes embeddings for labeled examples, $\boldsymbol{Z}^x$, unlabeled examples using a robust augmentation, $\boldsymbol{Z}^u$, and for prototypes, $\boldsymbol{Z}^c$. Subsequently, it merges these embeddings and generates pseudo-labels by applying a weak augmentation on unlabeled data, $\boldsymbol{Z}^w$. Finally, it combines the labels of labeled examples, $\boldsymbol{y}^x$, pseudo-labeled examples, $\boldsymbol{y}^u$, and labels of the prototypes, $\boldsymbol{y}^c$, to train the model using the SSC loss function, defined in Equation 7.

3.2 Weigthed Semi-SupCon Loss

Similar to other mixed-loss frameworks that include parameters to balance between supervised, pseudo-labeled, or unsupervised parts, we extend the previously defined semi-supervised contrastive loss $\mathcal{L}_{\text{SSC}}$ to feature weights, by using an additional parameter λ:

$$\mathcal{L}_{\text{SSC}}(\boldsymbol{Z}, \boldsymbol{y}, \boldsymbol{\lambda}) = \frac{1}{\sum_{k\in I}\lambda_k} \sum_{i\in I} \frac{-\lambda_i}{|P(i)|} \sum_{p\in\mathcal{P}(i)} \log\left(\frac{\exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_p/T)}{\sum_{j\in I\setminus\{i\}} \exp(\boldsymbol{z}_i \cdot \ \boldsymbol{z}_j/T)}\right) \quad (8)$$

These weights provide a mechanism to give higher importance to some anchors. In practise, we use a very simple strategy where we use a constant value

that depends only on the nature of the anchor :

$$\lambda_i = \begin{cases} \lambda^x & \text{if } \boldsymbol{z}_i \in \boldsymbol{Z}^x \\ \lambda^{u\uparrow} & \text{if } \boldsymbol{z}_i \in \boldsymbol{Z}^{u\uparrow} \\ \lambda^{u\downarrow} & \text{if } \boldsymbol{z}_i \in \boldsymbol{Z}^{u\downarrow} \\ \lambda^c & \text{if } \boldsymbol{z}_i \in \boldsymbol{Z}^c \end{cases} \tag{9}$$

More advanced approaches using adaptive weighting can be easily implemented, for instance with weights based on the confidence of the classifier p [8, 24]. From now, $\mathcal{L}_{\text{SSC}}$ always refer to this weighted version of the loss.

3.3 Link with cross-entropy

We now establish a relationship between cross-entropy (CE) loss and our framework using contrastive learning loss using prototypes, in the classical supervised framework, under mild assumptions. As already observed in previous work [28], both loss functions have inherent similarities, particularly in treating negative embeddings similarly to the weights of a linear classification layer. Our prototype-based approach builds on this analogy. If we remove the bias of the last projection layer, the CE loss H can be expressed in terms of the weights of the final linear projection layer $\boldsymbol{W} \in \mathbb{R}^{K \times h}$ as such :

$$\begin{aligned} H(\boldsymbol{Z}^x, \boldsymbol{y}) &= \frac{1}{B} \sum_{i=1}^{B} -\log \operatorname{softmax}(\boldsymbol{W}\boldsymbol{z}_i^x)_{y_i} \\ &= \frac{1}{B} \sum_{i=1}^{B} -\log \frac{\exp(\boldsymbol{z}_i^x \cdot \boldsymbol{w}_{y_i})}{\sum_{k=1}^{K} \exp(\boldsymbol{z}_i^x \cdot \boldsymbol{w}_k)} \end{aligned} \tag{10}$$

If we set the temperature of the SupCon loss to $T = 1$, and ensure the normalization of all embeddings, it is now easy to see that by replacing the weights $\boldsymbol{W}$ of the last layer with the prototypes $\boldsymbol{Z}_c$, we get :

$$H(\boldsymbol{Z}^x, \boldsymbol{y}) = \frac{1}{B} \sum_{i=1}^{B} \mathcal{L}_{\text{SupCon}} \left(\begin{bmatrix} \boldsymbol{z}_i^x \\ \boldsymbol{Z}^c \end{bmatrix}, \begin{bmatrix} y_i \\ \boldsymbol{y}^c \end{bmatrix} \right) \tag{11}$$

The CE loss is equivalent to applying separate SupCon losses to each example, each of which has only one positive pair that is its class prototype. Both losses aim fundamentally to learn prototypes $\boldsymbol{Z}^c$ or equivalently weights $\boldsymbol{W}$ to be aligned with their corresponding feature vectors given by the labels, which supports the use of the prototypes to learn a similar distribution probability p on the embeddings space.

4 Experiments

In the following section, we present our experimental setup, compare our method with established self-training approaches, and evaluate the impact of individual

components through an ablation study. We also investigate the transfer performance of our approach and its synergy with self-supervised pre-training, focusing on convergence speed. Finally, we analyze the stability of the hyperparameters in our proposed loss function.

4.1 Experimental setup

Our framework is evaluated on three classical benchmark datasets: CIFAR-100 [1], STL-10 [12], and SVHN [26]. For each dataset, we explore two splits, keeping a limited number of 4 and 25 labeled examples per class. We conducted each experiment using 3 random seeds and present both the mean and the standard deviation for each experiment. Following the setup in [29], baseline models are reported for 1024 epochs, where an epoch is arbitrary defined as 2^{10} steps following the literature. However, to demonstrate the efficiency of our approach, we only train with $\mathcal{L}_{\text{SSC}}$ on 256 epochs.

We use a Wide ResNet WRN-28-2 [36] for all experiments on CIFAR-100 and SVHN, while a larger WRN-37-2 is used for STL-10. Additionally, on top of these architectures, we added a projection head as mentioned in SupCon, which consists of a 2-layer MLP with dimensions of 128 for WRN-28-2 and 256 for WRN-37-2 (following the dimension of the original projection used with CE). For FixMatch, the strong augmentation used is RandAugment [13].

It is important to note that although RandAugment is commonly used in semi-supervised settings, it is not specifically designed for contrastive learning. Nevertheless, we decided to keep the same augmentation parameters as those used in FixMatch. To be fair, we adopted the exact hyperparameters from the original work, including all optimizer settings such as learning rate, schedule, weight decay, batch size B and ratio μ. Concerning the extra hyperparameter introduced in our framework, we keep them the same for all the experiments. We take $T = 0.01$ which is a common temperature value used in SupCon loss, and we take $T' = 0.04$.

Tuning this last parameters is actually equivalent to tuning the pseudo-labeling threshold τ, which is kept at $\tau = .95$ to be consistent with Fixmatch. Indeed, increasing T' will cause the posterior distribution p to approach the uniform distribution, which will have the same effect on pseudo-labeling as increasing τ. We chose to give the same importance to all embeddings by setting $\lambda^x = \lambda^{u\uparrow} = \lambda^c = 1$ except for unconfident one where we define $\lambda^{u\downarrow} = 0.2$

4.2 Experimental Results

Performance of $\mathcal{L}_{\text{SSC}}$ We begin our evaluation by comparing FixMatch with and without the use of our proposed semi-supervised contrastive loss against other leading self-training approaches. We conduct this comparison on CIFAR-100 and SVHN datasets, employing both 4 and 25 labeled training samples. We report the results of the state-of-art approaches that have been previously found

in the literature[1] and in order to see the effect of the proposed approach, we ran FixMatch with and without SemiSupCon loss on our servers.

Dataset	CIFAR-100		SVHN	
labels/class	4	25	4	25
Π-Model [22]	12.87 ±1.25	39.92 ±0.61	22.62 ±5.36	86.45 ±0.42
MixMatch [5]	20.05 ±0.29	50.42 ±0.62	20.37 ±5.78	96.29 ±0.2
VAT [25]	23.58 ±2.57	46.83 ±0.57	23.01 ±6.59	95.41 ±0.13
RemixMatch [4]	42.91 ±0.01	65.23 ±0.32	68.73 ±18.79	93.62 ±1.09
UDA [34]	46.56 ±2.06	65.63 ±0.28	97.71 ±0.02	97.72 ±0.03
FixMatch	46.62 ±2.38	65.35 ±0.62	97.83 ±0.03	97.96 ±0.07
FixMatch w. $\mathcal{L}_{SSC}$	**48.45 ±1.32**	**67.05 ±0.48**	**97.94 ±0.06**	**98.15 ±0.08**
Fully supervised (CE)	77.45 ± 0.02	77.54 ± 0.23	97.91 ± 0.02	97.91 ± 0.01

Table 2: Top-1 validation accuracy (%) of various self-training methods compared to FixMatch, without and with the integration into our proposed wrapper approach (denoted as FixMatch w. $\mathcal{L}_{SSC}$) obtained after convergence.

Based on the results presented in Table 2, it comes that UDA [34] demonstrates comparable performance to FixMatch. However, when employing FixMatch with the proposed approach, denoted as $\mathcal{L}_{SSC}$, the method notably enhances its competitiveness, particularly evident when training the model with only 4 labeled examples per class. These results underscore the effectiveness of our approach in leveraging all unlabeled data, particularly in scenarios where labeled data is scarce.

Transfer Performance Classical semi-supervised learning benchmarks typically require training models from scratch, a process that consumes considerable time. Due to these constraints, certain studies advocate for leveraging pre-trained models in semi-supervised approaches [15, 32].

In this line, we explore the efficacy of integrating self-supervised pre-training using MoCo v2 [11], into our methodology. Specifically, in this section, we use a ResNet-50 architecture[2] either trained from scratch or starting with MoCo v2 weights obtained after pretraining on ImageNet on 800 epochs[3].

Figure 2 plots the Top-1 accuracy in percentage with respect to the number of epochs. We first observe that, in addition to having higher accuracy, using $\mathcal{L}_{\text{SSC}}$ loss requires substantially fewer epochs to converge. With only 50 epochs,

[1] https://github.com/microsoft/Semi-supervised-learning/blob/main/results/classic_cv.csv

[2] We follow a standard adaptation of ResNet for smaller images, replacing the initial 7x7 convolutional layer with a 3x3 kernel and removing the final max pooling layer.

[3] https://github.com/facebookresearch/moco

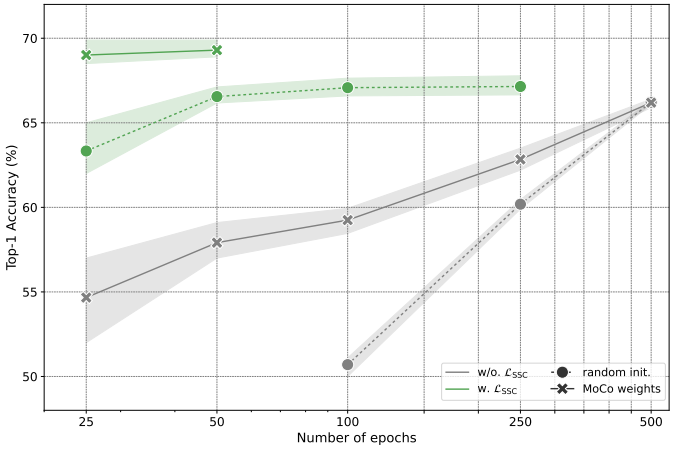

Fig. 2: Transfer Performance with FixMatch on CIFAR-100 with 25 labels per class. The color gray (resp. green) corresponds to FixMatch without $\mathcal{L}SSC$ (resp. with $\mathcal{L}SSC$), while the dashed (resp. solid) line represents training from scratch (resp. using MoCo v2 weights).

training with $\mathcal{L}_{\text{SSC}}$ from scratch already outperforms the standard approach with 500 epochs. Only 25 epochs are needed when using pretrained weights, achieving a significantly higher validation accuracy of 69.3%. Using all unlabeled data, including instances where the model has lower confidence, facilitates efficient training.

As noted, we observe a significant gain from the self-supervised pre-training with $\mathcal{L}_{\text{SSC}}$, which is not the case when using the classical CE loss. This underscores that our proposed loss seem to facilitate a smoother transition from pre-training methods, particularly those with a contrastive nature like MoCo.

Ablation study In order to investigate the effect of different components of $\mathcal{L}_{\text{SSC}}$, we perform an extensive ablation study, as reported in table 3 on CIFAR-100 and STL-10 by training the models with 256 epochs. We observed that using two strong augmentations slightly enhances the FixMatch technique. However, adding the self-supervised SimCLR loss tends to degrade performance, as already observed in [23]. Similarly, ignoring unconfident embeddings $\boldsymbol{Z}^{u\downarrow}$ or applying them with a separate SimCLR loss also degrades the performance of $\mathcal{L}_{\text{SSC}}$. The use of $\mathcal{L}_{\text{SSC}}$ consistently achieves the highest accuracy. These results justify our decision to incorporate them directly into our loss, thus facilitating global interaction with all other embeddings and prototypes.

Dataset	CIFAR-100		STL-10	
labels/class	4	25	4	25
(1) Base (FixMatch)	43.96 ±1.58	64.13 ±0.23	63.13 ±6.48	88.57 ±1.19
(2) w. Double Aug.	44.57 ±1.14	65.98 ±0.31	66.14 ±5.82	88.71 ±1.26
(3) w. Double Aug. + $\mathcal{L}_{\text{Self}}$	42.16 ±1.76	64.47 ±0.18	55.74 ±5.33	86.22 ±0.96
(4) $\mathcal{L}_{\text{SSC}}$, $\lambda_{u\downarrow} = 0$	45.03 ±1.26	65.96 ±0.14	70.21 ±6.95	87.69 ±1.65
(5) $\mathcal{L}_{\text{SSC}}$, $\lambda_{u\downarrow} = 0 + \mathcal{L}_{\text{Self}}$	43.06 ±1.42	62.12 ±0.17	56.45 ±7.21	86.72 ±1.36
(6) $\mathcal{L}_{\text{SSC}}$	**46.53 ± 1.18**	**66.28 ± 0.22**	**73.21 ± 6.73**	**88.94 ± 1.26**

Table 3: Ablation study on CIFAR-100 and STL-10 with 256 epochs for all experiments. Starting from FixMatch (1), we use a double strong augmentation (2) and add a separate SimCLR loss on all unlabeled data (3). On the other hand, we try to remove the unconfident embeddings from Fixmatch with $\mathcal{L}_{\text{SSC}}$ (4), and then to use these unconfident examples in a separate SimCLR loss. (5).

Hyperparameter stability analysis We examine the sensitivity of our framework to classical self-training hyperparameters, such as the pseudo-labeling confidence threshold τ, the imbalance ratio between labeled and unlabeled examples in the batch μ, and the strength of strong augmentation $\mathcal{A}(\cdot)$. Figure 3 illustrates the distribution of model performances across various hyperparameter settings.

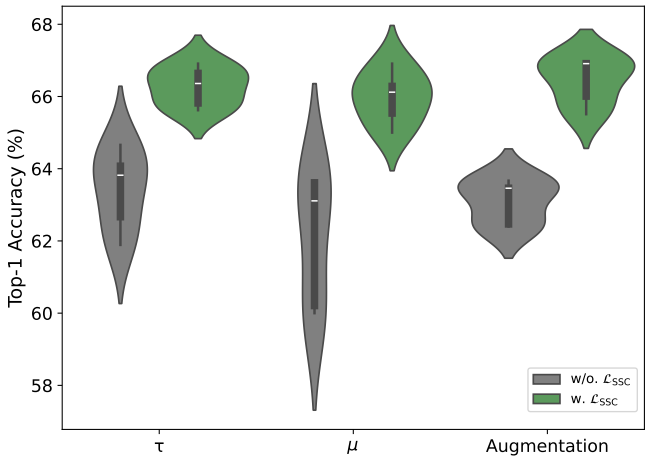

Fig. 3: Hyperparameter stability analysis with FixMatch on CIFAR-100. For each hyperparameter, 10 experiments are conducted using the same seed with different values uniformly distributed in the followings range : $\tau \in [.9, 0.98]$, $\mu \in \{3, ..., 12\}$ and RandAugment strength parameter in {3,...,20}. All experiments are run on 256 epochs with 25 labels/class.

Our contrastive approach, depicted in green, demonstrates significantly lower variance concerning the τ and μ parameters compared to alternative methods. However, both approaches appear equally sensitive to the augmentation strength. This outcome was anticipated since our contrastive framework relies on $\mathcal{A}(\cdot)$ for both consistency regularization and unsupervised contrastive learning through the utilization of unconfident embeddings $\boldsymbol{Z}^{u\downarrow}$.

5 Conclusion

In this paper, we introduce a new semi-supervised contrastive framework that combines SupCon with an unsupervised contrastive loss, effectively operating within a self-training setting. The proposed framework allows taking advantage of labeled, pseudo-labeled, and unconfident examples simultaneously in the training process.

Moreover, we propose the incorporation of class prototypes into contrastive learning to derive class probabilities, enhancing the interpretability and performance of the model.

By applying our approach to the FixMatch framework, we observe substantial performance gains across three datasets. Our method exhibits rapid convergence, benefits from pretraining, and showcases stability across various hyperparameters, underscoring its effectiveness and reliability in semi-supervised learning scenarios.

Future research avenues may explore further enhancements to the contrastive learning framework, such as incorporating domain-specific knowledge or adapting the framework to handle noisy or incomplete data. Additionally, investigating the interplay between contrastive learning and other semi-supervised learning techniques could lead to synergistic approaches with even greater performance gains.

References

1. Alex Krizhevsky and Geoffrey Hinton: Learning Multiple Layers of Features from Tiny Images. Tech. rep., University of Toronto (2009)
2. Amini, M.R., Feofanov, V., Pauletto, L., Hadjadj, L., Devijver, E., Maximov, Y.: Self-training: A survey (2023)
3. Assran, M., Caron, M., Misra, I., Bojanowski, P., Joulin, A., Ballas, N., Rabbat, M.G.: Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples. In: International Conference on Computer Vision (ICCV). pp. 8423–8432 (2021)
4. Berthelot, D., Carlini, N., Cubuk, E.D., Kurakin, A., Sohn, K., Zhang, H., Raffel, C.: ReMixMatch: Semi-Supervised Learning with Distribution Alignment and Augmentation Anchoring. In: International Conference on Learning Representations (ICLR) (2020)
5. Berthelot, D., Carlini, N., Goodfellow, I., Oliver, A., Papernot, N., Raffel, C.: MixMatch: A holistic approach to semi-supervised learning. In: Advances in Neural Information Processing Systems (NeurIPS). No. NeurIPS (2019)

6. Berthelot, D., Roelofs, R., Sohn, K., Carlini, N., Kurakin, A.: Adamatch: a Unified Approach To Semi-Supervised Learning and Domain Adaptation. In: International Conference on Learning Representations (ICLR) (2022)
7. Beyer, L., Zhai, X., Oliver, A., Kolesnikov, A.: S4L: Self-supervised semi-supervised learning. In: International Conference on Computer Vision (ICCV) (2019)
8. Chen, H., Tao, R., Fan, Y., Wang, Y., Wang, J., Schiele, B., Xie, X., Raj, B., Savvides, M.: SoftMatch: Addressing the Quantity-Quality Trade-off in Semi-supervised Learning. In: International Conference on Learning Representations (ICLR) (2023)
9. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning (ICML) (2020)
10. Chen, T., Kornblith, S., Swersky, K., Norouzi, M., Hinton, G.: Big Self-Supervised Models are Strong Semi-Supervised Learners (NeurIPS), 1–18 (2020)
11. Chen, X., Fan, H., Girshick, R., He, K.: Improved Baselines with Momentum Contrastive Learning pp. 1–3 (2020), http://arxiv.org/abs/2003.04297
12. Coates, A., Lee, H., Ng, A.Y.: An analysis of single-layer networks in unsupervised feature learning. Journal of Machine Learning Research **15**, 215–223 (2011)
13. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical automated data augmentation with a reduced search space. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (2020)
14. Cui, J., Zhong, Z., Liu, S., Yu, B., Jia, J.: Parametric Contrastive Learning. Proceedings of the IEEE International Conference on Computer Vision pp. 695–704 (2021)
15. Fini, E., Astolfi, P., Alahari, K., Alameda-Pineda, X., Mairal, J., Nabi, M., Ricci, E.: Semi-supervised learning made simple with self-supervised clustering. In: Conference on vision and Pattern Recognition (CVPR) (2023)
16. Graf, F., Hofer, C., Niethammer, M., Kwitt, R.: Dissecting supervised contrastive learning. In: International Conference on Machine Learning (ICML). pp. 3821–3830 (2021)
17. Gunel, B., Du, J., Conneau, A., Stoyanov, V.: Supervised Contrastive Learning for Pre-Trained Language Model Fine-Tuning. In: International Conference on Learning Representations (ICLR). pp. 1–15 (2021)
18. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition pp. 1735–1742 (2006)
19. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum Contrast for Unsupervised Visual Representation Learning. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition pp. 9726–9735 (2020)
20. Islam, A., Chen, C.F., Panda, R., Karlinsky, L., Radke, R., Feris, R.: A Broad Study on the Transferability of Visual Representations with Contrastive Learning. In: International Conference on Computer Vision (ICCV). pp. 8825–8835 (2021)
21. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised Contrastive Learning. Advances in Neural Information Processing Systems (NeurIPS) (2020)
22. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. In: International Conference on Learning Representations (ICLR). pp. 1–13 (2017)
23. Lee, D., Kim, S., Kim, I., Cheon, Y., Cho, M., Han, W.S.: Contrastive Regularization for Semi-Supervised Learning. In: Conference on Vision and Pattern Recognition (CVPR) (2022)

24. Li, J., Xiong, C., Hoi, S.C.: CoMatch: Semi-supervised Learning with Contrastive Graph Regularization. Proceedings of the IEEE International Conference on Computer Vision (2021)
25. Miyato, T., Maeda, S.I., Koyama, M., Ishii, S.: Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(8), 1979–1993 (2019)
26. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading Digits in Natural Images with Unsupervised Feature Learning (2011)
27. Oord, A.v.d., Li, Y., Vinyals, O.: Representation Learning with Contrastive Predictive Coding (2018)
28. Sohn, K.: Improved deep metric learning with multi-class N-pair loss objective. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 1857–1865 (2016)
29. Sohn, K., Berthelot, D., Li, C.L., Zhang, Z., Carlini, N., Cubuk, E.D., Kurakin, A., Zhang, H., Raffel, C.: FixMatch: Simplifying semi-supervised learning with consistency and confidence. In: Advances in Neural Information Processing Systems (NeurIPS) (2020)
30. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 1196–1205 (2017)
31. Wang, C., Cao, X., Guo2, L., Shi, Z.: DualMatch: Robust Semi-Supervised Learning with Dual-Level Interaction. In: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD) (2023)
32. Wang, Y., Chen, H., Fan, Y., Sun, W., Tao, R., Hou, W., Wang, R., Yang, L., Zhou, Z., Guo, L.Z., Qi, H., Wu, Z., Li, Y.F., Nakamura, S., Ye, W., Savvides, M., Raj, B., Shinozaki, T., Schiele, B., Wang, J., Xie, X., Zhang, Y.: USB: A Unified Semi-supervised Learning Benchmark for Classification. In: Advances in Neural Information Processing Systems (NeurIPS) (2022)
33. Wang, Y., Chen, H., Heng, Q., Hou, W., Fan, Y., Wu, Z., Wang, J., Savvides, M., Shinozaki, T., Raj, B., Schiele, B., Xie, X.: FreeMatch: Self-adaptive Thresholding for Semi-supervised Learning. In: The 11^{th} International Conference on Learning Representations, ICLR. pp. 1–20 (2022)
34. Xie, Q., Dai, Z., Hovy, E., Luong, M.T., Le, Q.V.: Unsupervised data augmentation for consistency training. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 6256–6268 (2020)
35. Xu, Y., Shang, L., Ye, J., Qian, Q., Li, Y.F., Sun, B., Li, H., Jin, R.: Dash: Semi-Supervised Learning with Dynamic Thresholding. In: 38th International Conference on Machine Learning (ICML). pp. 11525–11536 (2021)
36. Zagoruyko, S., Komodakis, N.: Wide Residual Networks. In: British Machine Vision Conference (BMVC). pp. 1–87 (2016)
37. Zhang, B., Wang, Y., Hou, W., Wu, H., Wang, J., Okumura, M., Shinozaki, T.: FlexMatch: Boosting Semi-Supervised Learning with Curriculum Pseudo Labeling. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 18408–18419 (2021)
38. Zheng, M., You, S., Huang, L., Wang, F., Qian, C., Xu, C.: SimMatch: Semi-supervised Learning with Similarity Matching. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14451–14461 (2022)
39. Zhu, J., Wang, Z., Chen, J., Chen, Y.P.P., Jiang, Y.G.: Balanced Contrastive Learning for Long-Tailed Visual Recognition. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2022)