Semi-supervised learning made simple with self-supervised clustering

Enrico Fini[*1] Pietro Astolfi[*1,2] Karteek Alahari[2] Xavier Alameda-Pineda[2]
Julien Mairal[2] Moin Nabi[3] Elisa Ricci[1,4]

[1] University of Trento [2] Inria[†] [3] SAP AI Research [4] Fondazione Bruno Kessler

Abstract

Self-supervised learning models have been shown to learn rich visual representations without requiring human annotations. However, in many real-world scenarios, labels are partially available, motivating a recent line of work on semi-supervised methods inspired by self-supervised principles. In this paper, we propose a conceptually simple yet empirically powerful approach to turn clustering-based self-supervised methods such as SwAV or DINO into semi-supervised learners. More precisely, we introduce a multi-task framework merging a supervised objective using ground-truth labels and a self-supervised objective relying on clustering assignments with a single cross-entropy loss. This approach may be interpreted as imposing the cluster centroids to be class prototypes. Despite its simplicity, we provide empirical evidence that our approach is highly effective and achieves state-of-the-art performance on CIFAR100 and ImageNet.

1. Introduction

In recent years, self-supervised learning became the dominant paradigm for unsupervised visual representation learning. In particular, much experimental evidence shows that augmentation-based self-supervision [3, 8–10, 14–16, 18, 21, 29, 32, 35, 71] can produce powerful representations of unlabeled data. Such models, although trained without supervision, can be naturally used for supervised downstream tasks via simple fine-tuning. However, the most suitable way to leverage self-supervision is perhaps by multi-tasking the self-supervised objective with a custom (possibly supervised) objective. Based on this idea, the community has worked on re-purposing self-supervised methods in other sub-fields of computer vision, as for instance in domain adaptation [25], novel class discovery [31, 77], continual learning [30] and semi-supervised learning [4, 13, 19, 72].

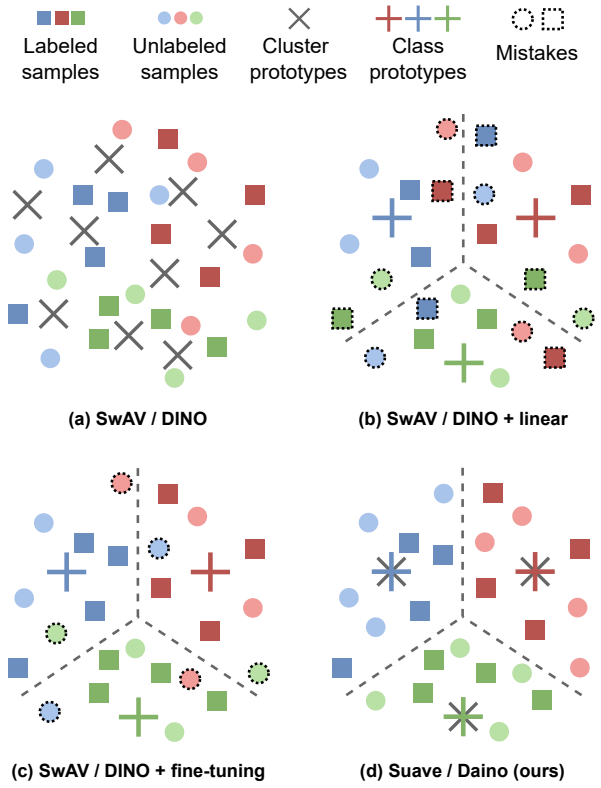


Figure 1. Schematic illustration of the motivation behind the proposed semi-supervised framework. (a) Self-supervised clustering methods like SwAV [15] and DINO [16] compute cluster prototypes that are not necessarily well aligned with semantic categories, but they do not require labeled data. (b) Adding a linear classifier provides class prototypes, but the labeled (and unlabeled) samples are not always correctly separated. (c) Fine-tuning can help separating labeled data. (d) Our framework learns cluster prototypes that are aligned with class prototypes thus correctly separating both labeled and unlabeled data.

One of the areas that potentially benefits from the advancements in unsupervised representation learning is semi-supervised learning. This is mainly due to the fact that in semi-supervised learning it is crucial to efficiently extract the information in the unlabeled set to improve the

[*]Enrico Fini and Pietro Astolfi contributed equally.
[†]Univ. Grenoble Alpes, CNRS, Grenoble INP, LJK, 38000 Grenoble, France.

classification accuracy on the labeled classes. Indeed, several powerful semi-supervised methods [6, 13, 61, 72] were built upon this idea.

In the self-supervised learning landscape, arguably the most successful methods belong to the clustering-based family, such as DeepCluster v2 [14], SwAV [15] and DINO [16]. These methods learn representations by contrasting predicted cluster assignments of correlated views of the same image. To avoid collapsed solutions and group samples together, they use simple clustering-based pseudo-labeling algorithms such as k-means and Sinkhorn-Knopp to generate the assignments. A peculiar fact about this family of methods is the discretization of the feature space, that allows them to use techniques that were originally developed for supervised learning. Indeed, similar to supervised learning, the cross-entropy loss is adopted to compare the assignments, as they represent probability distributions over the set of clusters.

In this paper, we propose a new approach for semi-supervised learning based on the simple observation that clustering-based methods are amenable to be adapted to a semi-supervised learning setting: the cluster prototypes can be replaced with class prototypes learned with supervision and the same loss function can be used for both labeled and unlabeled data. In practice, semi-supervised learning can be achieved by multi-tasking the self-supervised and supervised objectives. This encourages the network to cluster unlabeled samples around the centroids of the classes in the feature space. By leveraging on these observations we propose a new framework for semi-supervised methods based on self-supervised clustering. We experiment with two instances of that framework: *Suave* and *Daino*, the semi-supervised counterparts of SwAV and DINO. These methods have several favorable properties: i) they are efficient at learning representations from unlabeled data since they are based on the top-performing self-supervised methods; ii) they extract relevant information for the semantic categories associated with the data thanks to the supervised conditioning; iii) they are easy to implement as they are based on the multi-tasking of two objectives. The motivation behind our proposal is also illustrated in Fig. 1. As shown in the figure, our multi-tasking approach enables to compute cluster centers that are aligned with class prototypes thus correctly separating both labeled and unlabeled data.

Our **contributions** can be summarized as follows:

- We propose a new framework for semi-supervised learning based on the multi-tasking of a supervised objective on the labeled data and a clustering-based self-supervised objective on the unlabeled samples;
- We experiment with two representatives of such framework: Suave and Daino, semi-supervised extensions of SwAV [15] and DINO [16]. These methods, while simple to implement, are powerful and efficient semi-supervised learners;
- Our methods outperform state-of-the-art approaches, often relying on multiple *ad hoc* components, both on common small scale (CIFAR100) and large scale (ImageNet) benchmarks, setting a new state-of-the-art on semi-supervised learning.

2. Related works

Self-supervised learning. The self-supervised literature has rapidly become extremely vast [38]. Excluding a recent trend involving denoising autoencoders [59] combined with vision transformers [27] like in MAE [34], the vast majority of existing methods are still relying on multiple views derived from each sample via data augmentation. These augmentation-based methods [3, 8–10, 14–16, 18, 21, 29, 32, 35, 71] are capable to learn rich representations during unsupervised pre-training, achieving performance comparable to their supervised counterparts when fine-tuned with the labels. Basically, these methods enforce correlated views of the same input to have coherent representations in latent space such that the model becomes invariant to the augmentations applied. This corresponds to maximizing the mutual information between views' representations and, in the literature, it has been done using different loss functions.

Contrastive-based methods [8, 18, 35, 37] define a loss based on noise-contrastive estimation [33] as instance discrimination [63] between positive (correlated) and negative views (the remaining samples in the mini-batch). A major drawback of these methods is that they require large mini-batches to have representative negative samples. To overcome this difficulty, consistency-based methods like [21, 32] propose to maximize the cosine similarity of positive pairs without considering negatives, whereas redundancy-reduction-based methods [9, 10, 29, 71] employ principled regularization terms to minimize features redundancy. For instance, in [71] a loss is introduced to minimize the cross-correlation between features of the positive pairs. Similarly, in [9, 10] the features learning process minimizes the covariance alongside regulating the variance of the embeddings.

Clustering-based methods [3, 14–16, 45], instead, naturally discretize the latent space via clustering. They perform clustering either offline (using the whole dataset), as in DeepCluster v2 [14], PCL [45] and SeLa [3], or online (using mini-batches), as in SwAV [15] and DINO [16], and impose coherent clustering assignments of positive pairs through a cross-entropy. Noticeably, this loss contrasts targets (assignments) and predictions as in the standard supervised learning objective. Nonetheless, surprisingly, to the best of our knowledge there have been no previous attempts in the literature to exploit this favorable property in the semi-supervised scenario. Our work aims to fill this gap.

Semi-supervised learning. Semi-supervised approaches aim to exploit a limited amount of annotations and a large collection of unlabeled data. The most intuitive approach to this task is perhaps Pseudo-Labels [43], based on self-training via pseudo-labeling: a model trained on labeled data generates categorical pseudo-labels for the unlabeled examples, which will be then integrated into the labeled set for the next model training. However, hard (categorical) labels easily exacerbate the classification bias of the training model, a phenomenon known as *confirmation bias* [2]. To counteract this issue, researchers have shown benefits from soft labels and confidence thresholding [2] as well as from different training strategies like co- and tri-training [17,48,51], model distillation [65], consistency regularization (see below) and model de-biasing [54, 62].

Consistency regularization methods operate by introducing additional losses computed on unsupervised samples and enforce consistency of the network output under perturbation of the model and/or the input [7, 42, 47, 49, 53, 57, 58, 64, 75]. Recent approaches integrate pseudo-labeling techniques and consistency regularization. FixMatch [55] generates pseudo-labels from weak perturbations of the input that are used as target for strong input perturbations whenever they satisfy an arbitrary confidence threshold. [11, 12, 46] exploit MixUp [74] to improve class boundaries in low-density regions. Other works explore more advanced pseudo-labeling techniques based on adaptive confidence thresholds [66, 73] and meta-learning [50], uncertainty estimation [22, 60], latent structure regularization [44]. Recently, ConMatch [40] extended prediction consistency with self-supervised features consistency, while SimMatch [76] improved consistency regularization by applying it at both semantic-level and instance-level. Class-aware Contrastive Semi-Supervised Learning (CCSSL) is introduced in [67] to improve the quality of pseudo-labels in presence of unknown (out-of-distribution) classes.

As self-supervised methods have the ability to extract relevant information from unlabeled data, they can be leveraged to tackle the semi-supervised problem. Noticeably, self-supervised pre-training is found beneficial for many consistency regularization methods [13]. Moreover, semi-supervised self-supervised methods exist, which consider multi-tasking [61, 72], pre-training distillation [19], exponential moving average normalization [13] and supervised contrastive [4]. Finally, PAWS [6] borrows some principles from self-supervised clustering, but, somewhat similarly to SimMatch [76], it assumes labeled instances as anchors to compare views. However, it does not exploit the multitasking of the self- and semi- supervised objectives nor takes advantage of latent clustering.

3. Clustering-based semi-supervised learning

Clustering-based self-supervision is typically adopted in unsupervised representation learning scenarios where no labeled data is available [14–16]. However, in this work, we aim to take advantage of the few annotations available in the semi-supervised setting to learn even better representations. Our main intuition is to replace the cluster centroids with class prototypes learned with supervision. In this way, unlabeled samples will be clustered around the class prototypes, guided by the self-supervised clustering-based objective. To this end, we jointly optimize a supervised loss on the labeled data and a self-supervised loss on the unlabeled data. It turns out that using the same loss function (cross-entropy) is a very good choice since it promotes synergy between the two objectives and eases up the implementation. In the following, we formalize self-supervised clustering in Section 3.1. Then, we describe the details of our novel semi-supervised learning framework in Section 3.2 and show its application through two popular self-supervised approaches in Section 3.3.

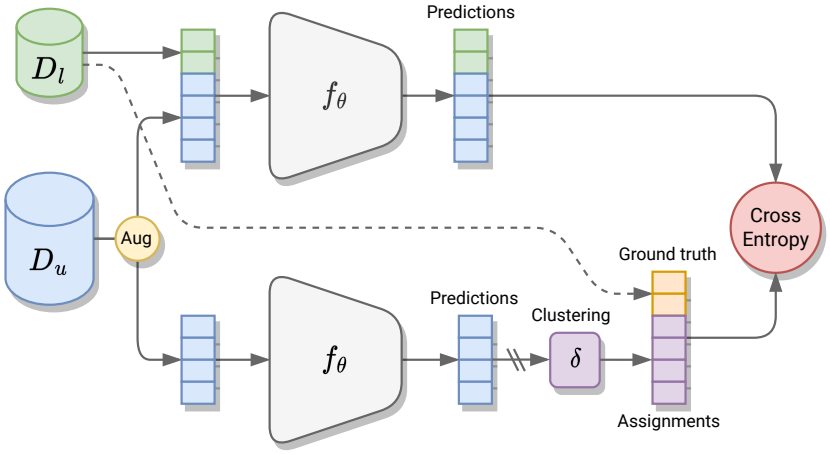
3.1. Clustering-based self-supervised learning

Given an unlabeled dataset $\mathcal{D}_u = \{\mathbf{x}_u^1, ..., \mathbf{x}_u^N\}$, two correlated views of the same input image, $(\mathbf{x}, \hat{\mathbf{x}})$, are generated via data augmentation and embedded through an encoder network f_θ, composed of a backbone g, a projector h and a set of prototypes (or centroids) p implemented as a bias-free linear layer. Performing the forward pass of the backbone and the projector produces two latent representations, $(\mathbf{h}, \hat{\mathbf{h}})$ for the two correlated views $(\mathbf{x}, \hat{\mathbf{x}})$, respectively. Subsequently, the cluster centroids p are used to produce two sets of logits $(\mathbf{p}, \hat{\mathbf{p}})$ corresponding to each of the two latent representations. A softmax function σ can be applied to these logits to obtain probability distributions over the set of clusters, also referred to as cluster assignments.

In principle, the two assignments could be immediately compared in order to encourage the network to output similar predictions for similar inputs (correlated views). However, this may lead to degenerate solutions where all samples are assigned to the same cluster. To avoid collapse, simple clustering techniques are usually employed to embed priors into the cluster assignment process and regularize the training procedure. In practice, we compute:

$$\hat{\mathbf{y}} = \delta(\hat{\mathbf{p}}, \mathbf{c}, \epsilon), \tag{1}$$

where δ is the clustering technique, which takes as input some predicted logits $\hat{\mathbf{p}}$, a context $\mathbf{c}$ and a temperature-like parameter ϵ that usually regulates the entropy of the assignment $\hat{\mathbf{y}}$ (sometimes addressed as pseudo-label). The context $\mathbf{c}$ can be implemented in different flavors depending on the nature of δ. For instance, for offline clustering $\mathbf{c}$ is represented by the features of all the samples in the dataset, while



Pseudo-code for Suave and Daino.

```
# x_l, x_u: labeled and unlabeled samples
# label: one-hot ground truth for x_l
def training_step(x_l, x_u, label):

    # correlated views
    x1_u, x2_u = aug(x_u), aug(x_u)

    # forward pass
    p_l = f(x_l)
    p1_u, p2_u = f(x1_u), f(x2_u)

    # compute assignments via clustering
    assign = delta(p2_u.detach())

    # concatenate predictions and targets
    pred = cat(p_l, p1_u)
    target = cat(label, assign)

    # compute loss
    loss = cross_entropy(pred, target)
    return loss
```

Figure 2. Left: overview of the proposed family of semi-supervised methods. Right: PyTorch-like pseudo-code for Suave and Daino.

for online clustering the context might contain the features of the current batch or simply a running mean of the overall distribution. The pseudo-label can be categorical (hard) or soft. Empirically soft cluster assignments were shown to yield superior performance [15].

The output of the clustering technique is then used as a target in the cross-entropy loss:

$$\ell(\mathbf{s}, \hat{\mathbf{y}}) = -\sum_{k=1}^{K} \hat{\mathbf{y}}_k \log\left(\mathbf{s}_k\right), \tag{2}$$

where $\mathbf{s} = \sigma(\mathbf{p}/\tau)$ has gone through softmax normalization with temperature τ, and K is the number of clusters. Note that a stop-gradient operation is performed, so that the gradient is not propagated through the pseudo-label. It is worth mentioning that the cross-entropy loss in Eq. 2 is asymmetric, but, it can be made symmetric by swapping $\mathbf{x}$ with $\hat{\mathbf{x}}$.

Intuitively, this loss leverages cluster assignments as a proxy to minimize the distance between latent representations $(\mathbf{h}, \hat{\mathbf{h}})$ of augmented views of the same image. As a by-product, the objective also learns a set of cluster prototypes encoded in the last linear layer p, which represent semantic information in the latent space. However, these clusters have no guarantee to be aligned with the true semantic categories represented in the dataset. Nonetheless, the discretization of the feature space is particularly interesting in the context of semi-supervised learning, as described in the following.

3.2. Our semi-supervised learning framework

In the semi-supervised scenario, we assume having access to a partially labeled dataset, $\mathcal{D} = \mathcal{D}_u \cup \mathcal{D}_l$, usually with $|\mathcal{D}_l| << |\mathcal{D}_u|$ and $\mathcal{D}_l$ contains a number of C known classes. We propose to exploit clustering-based self-supervised models described in Sec. 3.1 as a base to extract information from unlabeled data, and we extend them to take advantage of the labeled samples.

As mentioned, the main drawback of clustering-based self-supervised methods in the context of semi-supervised learning is that there are no guarantees that the stochastic optimization process will organize the clusters in the feature space according to the class labels. Indeed, they may be completely misaligned with the actual distribution of the classes. This also potentially hinders the effectiveness of the representations, as some prototypes may encode spurious correlations in the data. An ideal scenario is one where the clusters are centered on the actual class centroids such that the label can be propagated to the unlabeled samples by means of the clustering function. This will generate positive feedback that progressively transfers information from the labeled set into the unlabeled one, thus improving the feature representations learned by the network.

We propose to condition the cluster prototypes to encode the class information by resorting to multi-tasking of the self-supervised and supervised objectives. This can be achieved by optimizing the same loss function in Eq. 2 while replacing the pseudo-label with the ground-truth label when available:

$$\ell(\mathbf{s}, \bar{\mathbf{y}}), \quad \bar{\mathbf{y}} = \begin{cases} \mathbf{y}, & \mathbf{x} \in \mathcal{D}_l \\ \hat{\mathbf{y}}, & \mathbf{x} \in \mathcal{D}_u. \end{cases} \tag{3}$$

Since the linear layer p that contains the prototypes is now shared between the two objectives, we set $K = C$ to have a matching label space, although in principle this is not a hard constraint (as shown in [31]). In a nutshell, in our framework we compute the forward pass for both labeled and unlabeled samples, concatenate the associated predictions and the targets and apply the cross-entropy loss simultaneously as described in Fig. 2. We empirically demonstrate in Sec. 5

that, despite its simplicity, our framework equipped with these design choices is a strong semi-supervised learner.

3.3. Suave and Daino

Our proposed framework described above can convert any clustering-based self-supervised method into a semi-supervised learner, without dropping, adding or replacing any architectural components and reusing the same loss function. We select two representative self-supervised methods to showcase our framework: SwAV [15] and DINO [16], whose semi-supervised extensions we name **Suave** and **Daino**. This choice is motivated by their superior representation learning capabilities and ease of use. In particular, both are online clustering methods, which means that the cluster assignments can be computed on-the-fly without accessing the whole dataset simultaneously. This is a great advantage, especially for large scale datasets. For these reasons, we discard offline clustering methods like DeepCluster [14] and PCL [45], while in principle our approach can be also applied to them.

Suave. SwAV [15], following [3], casts the pseudo-label assignment problem as an instance of the optimal transport problem and proposes the swapped prediction task where the assignment of a view is predicted from the representation of another view. Simply put, SwAV generates pseudo-labels such that each cluster is approximately equally represented in the current batch, preventing the network from falling into degenerate solutions. This is especially convenient as we only need the information in the current batch for clustering. In light of our proposed framework, reusing Eq. 1, in Suave the target for the first sample in the batch can be obtained as follows (the same reasoning can be trivially applied to the other samples in the batch):

$$\hat{\mathbf{y}}_1 = \delta(\hat{\mathbf{p}}_1, [\hat{\mathbf{p}}_2, ..., \hat{\mathbf{p}}_B], \epsilon) \tag{4}$$

where the context $\mathbf{c} = [\hat{\mathbf{p}}_2, ..., \hat{\mathbf{p}}_B]$ contains all the logits in the batch except $\hat{\mathbf{p}}_1$. Now we define $\hat{\boldsymbol{P}} = [\hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2, \dots, \hat{\mathbf{p}}_B]$ and $\hat{\boldsymbol{Y}} = [\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \dots, \hat{\mathbf{y}}_B]^\top$, where $\hat{\boldsymbol{Y}}$ is the matrix that holds the unknown pseudo-labels of the whole batch. The clustering function δ will return the first column of $\hat{\boldsymbol{Y}}$ that is found by solving:

$$\hat{\boldsymbol{Y}} = \max_{\boldsymbol{Y} \in \Gamma} \mathrm{Tr}(\boldsymbol{Y}\hat{\boldsymbol{P}}) + \epsilon\, \mathrm{H}(\boldsymbol{Y}), \tag{5}$$

where $\epsilon > 0$ is the temperature-like parameter mentioned in Eq. 1, H is the entropy function, Tr is the trace operator, and Γ is the transportation polytope defined as:

$$\Gamma = \{\boldsymbol{Y} \in \mathbb{R}_+^{C \times B} | \boldsymbol{Y}\mathbf{1}_B = \frac{1}{C}\mathbf{1}_C, \boldsymbol{Y}^\top \mathbf{1}_C = \frac{1}{B}\mathbf{1}_B\}. \tag{6}$$

These constraints enforce that, on average, each cluster is selected $\frac{B}{C}$ times in each batch, automatically ensuring debiased assignments. The solution to Eq. 5 is obtained using the Sinkhorn-Knopp algorithm [1, 23].

Daino. DINO [16] aims at further simplifying the pipeline described above. Instead of using optimal transport on the predictions of the current batch it uses two practical tricks to avoid collapse: a momentum encoder and a pseudo-labeling strategy based on centering and sharpening. A momentum encoder is a "slow" version of the encoder f_θ updated using exponential moving average (EMA). After each gradient step on θ, the parameters ϕ of the momentum encoder f_ϕ are updated as follows:

$$\phi \leftarrow \eta\phi + (1-\eta)\theta, \tag{7}$$

where η is a rate parameter. The rationale behind this choice is that the momentum encoder serves as a teacher producing more stable representations throughout training, improving the optimization process. The teacher is used at every iteration to generate the logits $\hat{\mathbf{p}} = f_\phi(\hat{\mathbf{x}})$ which in turn are input to the clustering function δ in Eq. 1 to obtain the target assignment for Daino:

$$\hat{\mathbf{y}} = \delta(\hat{\mathbf{p}}, \gamma, \epsilon) = \sigma\left(\frac{\hat{\mathbf{p}} - \gamma}{\epsilon}\right), \tag{8}$$

where σ and ϵ are the softmax function and a temperature coefficient (here used to sharpen the distribution) respectively. The context $\mathbf{c} = \gamma$ in this case is a centering vector that approximates and de-biases the overall distribution of the data over the clusters and is also updated using EMA:

$$\gamma \leftarrow \mu\gamma + (1-\mu)\frac{1}{B}\sum_{b=0}^{B}\hat{\mathbf{p}}_b, \tag{9}$$

where μ adjusts the rate of the update. In brief, centering prevents one dimension to dominate but encourages high-entropy outputs, while sharpening does the opposite. Empirical evidence shows that this is enough to avoid collapse.

4. Implementation details

Architectures. For large-scale datasets (ImageNet), we adopt ResNet50 [36] and ViT-S/16 [27] backbones. For small-scale datasets (CIFAR100) we train a Wide ResNet (WRN-28-8) [70]. The convolutional backbone is followed by a projection head consisting of a multi-layer perceptron (with batch normalization in the hidden layers). We use 2 layers as in [6, 15]. After that, we perform L2-normalization and compute the predictions using a bias-free L2-normalized linear layer corresponding to the prototypes. We set the number of prototypes equal to the number of classes of the dataset at hand. Moreover, we perform an online linear evaluation at different depths of the network to identify which layer learns the best representations; we append a detached classification head on top of the backbone, the first and the second layer of the projection. We remark that such heads do not impact the efficiency of the training.

Semi-supervised pre-training. We pre-train our models using LARS [68] optimizer with linear warmup plus cosine learning rate schedule. Also, we adopt weight decay regularization. The backbone layers of the models can be initialized with self-supervised checkpoints of SwAV and DINO. At each training iteration, we sample a mini-batch composed of unlabeled images[1] and labeled images. Note that we count a training epoch considering a full pass over the unlabeled dataset. We optimize the cross-entropy loss of Eq. 3, re-weighting the labeled and unlabeled terms by their frequency in the batch. Moreover, we soften the supervised targets with label smoothing to mitigate overfitting. For the pseudo-labeling, in Suave, we re-use the same parameters for the Sinkhorn-Knopp as in SwAV; in Daino, we inherit the hyperparameters for the centering and the momentum encoder, whereas we tune the sharpening coefficient. More details are in the supplementary material.

Data augmentation. Images are augmented differently based on whether they are unlabeled or labeled. For the unlabeled images, we follow the default self-supervised augmentations of SwAV and DINO, while for the labeled, we adopt lighter augmentation Inception-style (random crop and flip) [56] and color distortion (jittering and greyscale). Note that it is important not to over-distort the labeled images as they are needed to align the clusters and classes.

To boost the self-supervised feature learning, we employ the multi-crop [15] augmentation scheme for unlabeled images. From each input image, we derive two global views from larger and higher-resolution crops and multiple smaller views from tighter crops. This is common practice in self-supervised learning. Similar to SwAV and DINO, we compute a clustering assignment only for the global views and use them as targets for the smaller ones. All the multi-crop views are taken into account when weighting the loss.

Another augmentation technique we empirically found to be useful is based on the combination of CutMix [69] and MixUp [74]. We apply it to both unlabeled (global views only) and labeled images, but separately. For the unlabeled images, we interpolate the learned clustering assignments due to the lack of ground-truth labels. This augmentation allows for shifting decision boundaries to low-density regions of the data [12, 58]. We mix the whole batch at every iteration, but to not over-regularize, we concatenate the mixed images to the current batch, instead of substituting it.

Semi-supervised fine-tuning. Since self-supervised methods are trained with strong augmentations, it is preferable after pre-training to fine-tune them to slightly improve the performance. We discover that a semi-supervised fine-tuning recipe works better than the typical fully supervised fine-tuning e.g., [4]. In practice, we just keep training the model with the same objective (our semi-supervised clustering-based loss) for a few more epochs, while relaxing some of the stronger augmentations adopted during pre-training, i.e., disable multi-crop and color distorsions.

[1] We sample unlabeled examples from dataset $\mathcal{D}$, instead of $\mathcal{D}_u$.

5. Experiments

5.1. Experimental protocol

Datasets. We perform our experiments using two common datasets, i.e., CIFAR100 [41] and ImageNet-1k [26]. In the semi-supervised setting the training set of each of these datasets is split into two subsets, one labeled and one unlabeled. For CIFAR100, which is composed of 50K images equally distributed into 100 classes, we investigate three splits as in [55], retaining 4 (0.8%), 25 (5%), and 100 (20%) labels per class, resulting in total to 400, 2500, and 10000 labeled images, respectively. For ImageNet-1k ($\sim$1.3K images per class, 1K classes), we adopt the same two splits of [19] using 1% and 10% of the labels. In both datasets, we evaluate the performance of our method by computing top-1 accuracy on the respective validation/test sets.

Baselines. We compare our methods, Suave and Daino, with state of the art methods from the semi-supervised literature (see Section 2). In particular, we compare against hybrid consistency regularization methods like SimMatch [76] and ConMatch [40] (and others [46, 60, 73]), and methods that are derived from self-supervised approaches, like PAWS [6] and S^4L-Rot [72]. We also compare with recent debiasing-based pseudo-labeling methods like DebiasPL [62]. For the sake of fairness, we leave out methods using larger architectures or pre-trained on larger datasets, e.g., DebiasPL with CLIP [52] and SimCLR v2 [19].

5.2. Results

First, we demonstrate the effectiveness of our semi-supervised framework in a small-scale dataset using the CIFAR100 benchmark. Then, we evaluate our models at large-scale on ImageNet-1k (see Section 5.2.1). Finally, we ablate the different components of our models (see Section 5.2.2).

5.2.1 Comparison with the state of the art

Results on CIFAR100. Table 1 shows a comparison between our methods and several semi-supervised approaches in the literature. In particular, we compare to consistency regularization semi-supervised methods (see Section 2) like ConMatch [40] and SimMatch [76], which are the strongest methods on this dataset. First, from the table we observe that both Suave and Daino achieve high performance in all the three splits (400, 2500, 10000). Daino obtains results comparable to the best competitors, while Suave outperforms all the baselines and beats the best methods CCSSL [67] and SimMatch [76] by +2.4%p, +1.3%p, +1.5%p in the three settings, respectively. Overall, these results clearly demonstrate how our clustering-based semi-

Table 1. Comparison with the state-of-the-art on CIFAR100.

Method	Acc@1		
	400	2500	10000
Π-Model [42]	-	42.8	62.1
Mean Teacher [57]	-	46.1	64.2
MixMatch [12]	32.4	60.2	72.2
UDA [64]	53.6	72.3	77.5
ReMixMatch [11]	55.7	72.6	77.0
FixMatch [55]	50.1	71.4	76.8
Dash [66]	55.2	72.8	78.0
CoMatch [44]	60.0	73.0	78.2
Meta Pseudo Labels [50]	55.8	72.3	77.5
FlexMatch [73]	60.1	73.5	78.1
FixMatch+DM [46]	59.8	74.1	79.6
NP-Match [60]	61.1	74.0	78.8
ConMatch [40]	61.1	74.6	-
SimMatch [76]	62.2	74.9	79.4
CCSSL [67]	61.2	75.7	80.1
Daino	61.1	75.2	79.2
Suave	**64.6**	**77.0**	**81.6**

Table 2. Comparison with the state-of-the-art on ImageNet-1k with ViT-S/16 [27]. DINO and MSN perform linear evaluation with labeled data on top of frozen features.

Method	**Epochs**	**Batch size**		**Acc@1**	
		Unlab.	Lab.	10%	1%
DINO [16]	(800)	1024	-	72.2	64.5
MSN [5]	(800)	1024	-	-	**67.2**
Daino	(800) 60	1024	512	**76.6**	67.1

supervised learning methods achieves state-of-the-art performance without requiring any *ad-hoc* confidence thresholds for pseudo-labels as in most recent consistency regularization methods. Interestingly, comparing Suave and Daino with their self-supervised counterparts, SwAV and DINO, a remarkable improvement is achieved. In fact, SwAV and DINO obtain an accuracy of 64.9% and 66.8% [24], respectively, when linearly evaluated using 100% of the labels after self-supervised pre-training.

Results on ImageNet-1k. We also perform large scale experiments on ImageNet-1k considering the label splits of 1% and 10% as common in previous works [55, 76]. Due to limited computational resources, we primarily focus on Suave with a ResNet50 backbone, which enables us to compare with most of the related state-of-the-art methods. In addition, we also provide experimental evidence that our framework works well with a different clustering algorithm (Daino), backbone (ViT [27]), and a simpler training recipe (described in the supplementary meterial).

We compare against three families of methods, self-supervised-inspired semi-supervised approaches [6, 72], consistency regularization methods [13, 44, 50, 55, 64, 76], and debiasing-based methods [62], and present results in Table 3, Table 2, and Figure 3. To provide full context to our results, in Table 3, we also report the performance of self-supervised models when simply fine-tuned with labels.

Table 3. Comparison with the state-of-the-art on ImageNet-1k. All the models reported use ResNet-50. In the first and the second column, we indicate within brackets whether a model is initialized from a self-supervised checkpoint and the number of epochs of that pre-training. For brevity, we refer to FixMatch as FM. *SimMatch does not report the batch size, and its value is inferred from the public repository. † refers to epochs on labeled data.

Method	**Epochs**	**Batch size**		**Acc@1**	
		Unlab.	Lab.	10%	1%
with similar batch size and number of epochs					
S^4L-Rotation [72]	200†	256	256	61.4	-
FM-DA (MoCo v2) [44]	(800) 400	640	160	72.2	59.9
PAWS [6]	100	256	1680	70.2	-
CoMatch (MoCo v2) [44]	(800) 400	640	160	73.7	67.1
FM-EMAN (MoCo-EMAN) [13]	(800) 300	320	64	74.0	63.0
SimMatch [76]	400	320*	64*	74.4	**67.2**
DebiasPL (MoCo-EMAN) [62]	(800) 50	640	128	-	65.3
	(800) 200	640	128	-	66.5
Suave	(100) 100	256	128	73.6	63.8
	(200) 100	256	128	74.3	65.0
	(800) 100	256	128	**75.0**	66.2
with larger batch size or number of epochs					
UDA [55]	∼480	15360	512	68.8	-
Meta Pseudo Labels [50]	∼800	2048	2048	73.9	-
PAWS [6]	100	4096	6720	73.9	63.8
	200	4096	6720	75.0	66.1
	300	4096	6720	75.5	66.5
DebiasPL (MoCo-EMAN) [62]	(800) 300	1280	256	-	67.1
self-supervised pre-training with fine-tuning					
MoCo v2 [20]	(800)	256	-	66.1	49.8
SimCLR v2 [18]	(1000)	4096	-	68.4	57.9
BYOL [32]	(1000)	2048	-	68.8	53.2
MoCo-EMAN [13]	(800)	256	-	68.1	57.4
SwAV [13]	(800)	4096	-	70.2	53.9
NNCLR [28]	(1000)	4096	-	70.2	56.4
Barlow Twins [71]	(1000)	2048	-	69.7	55.0
FNC [37]	(1000)	4096	-	71.1	63.7

Suave and Daino obtain performances that are comparable with state-of-the-art methods on ImageNet-1k. In particular, when compared against related methods (e.g., DebiasPL, SimMatch, FixMatch-EMAN, CoMatch) with similar batch-size and number of epochs, Suave obtains the best score (+0.6%p) on the 10% setting and the third best score on the 1% setting, -1.0%p from the best baseline SimMatch [76]. Importantly, other approaches like PAWS [4] require larger batches to obtain comparable results. This aspect is even more evident by looking at Fig. 3 (bottom), where PAWS significantly underperforms with respect to our method when decreasing the batch size. This behavior can be ascribed to the fact that PAWS adopts a k-nearest neighbor approach on the labeled instances to generate the assignment vectors (pseudo-labels), and thus it requires a sufficiently high number of labeled examples to well rep-

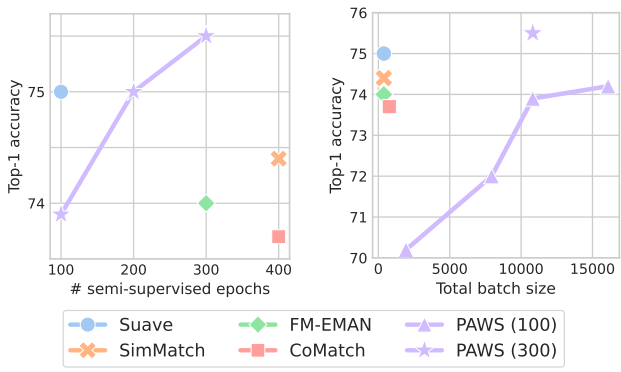


Figure 3. Comparison with related works on the training efficiency in terms of pre-training epochs (top) and size of the mini-batches (bottom). Plotted results refer to the ImageNet 10% split.

resent the class distributions. In contrast, Suave generates the assignments using the learnable cluster/class prototypes, which are a fixed number independent of the batch size.

Figure 3 (top) compares Suave with the best methods in Table 3, highlighting the fast convergence of our method. Performing 100 epochs of semi-supervised pre-training is enough to achieve comparable results to other related methods like FixMatch-EMAN, PAWS, and SimMatch, which require at least twice the semi-supervised epochs. It is worth noting that Suave naturally benefits from self-supervised initialization, as it basically shares the objective with the self-supervised counterpart SwAV. Indeed, by looking at Table 3 we can observe that better SwAV checkpoints lead to higher accuracy. At the same time, the difference between Suave (SwAV-100) and Suave (SwAV-800) is rather small, 1.4%p and 2.4%p on 10% and 1%, respectively.

5.2.2 Ablation study

Here, we assess the importance of the main components of our method: (i) the multi-task training objective, (ii) the adopted fine-tuning strategy, and (iii) the quality of representations at different depth of the network.

In Table 4, we demonstrate the importance of our multi-tasking strategy. Suave significantly outperforms vanilla SwAV, even in its improved version, i.e., SwAV (repro) obtained with the fine-tuning protocol borrowed from PAWS [6]. More importantly, when comparing Suave to SwAV+CT (SwAV where the self-supervised loss is multi-tasked with a supervised contrastive loss [39]) we still observe that our method is far superior. This clearly indicates that it is better to exploit the available labels to condition the prototypes rather than using them to sample positive instances for contrastive training.

Table 5 provides evidence that fine-tuning Suave with our semi-supervised fine-tuning strategy (see Section 4) is more effective than adopting the classical fully supervised recipe. We believe that feeding the model with unlabeled data during the fine-tuning phase allows the model to better

Table 4. Impact of our multi-task training strategy. Legend: **UPT** = "unsupervised pre-training", **MT** = "multi-tasking" and **SL** = "same loss for all training phases".

Method	UPT	MT	SL	Epochs	Acc@1	
					10%	1%
SwAV	✓		✓	800	70.2	53.9
SwAV (repro)	✓		✓	800	72.3	57.0
SwAV+CT [4]	✓	✓		(400)30	70.8	-
Suave	✓	✓	✓	(800)100	**75.0**	**66.2**

Table 5. Impact of our semi-supervised fine-tuning strategy.

Method	Labels (%)	Semi-supervised pre-training	Supervised fine-tuning	Semi-supervised fine-tuning
Suave	1%	64.1	64.8	**66.2**
	10%	73.4	74.8	**75.0**

Table 6. Online linear evaluation on the projector layers. We attach a linear layer (preceded by a stop-grad) after each projector layer and train it with labeled data.

Method	Labels (%)	Backbone	Projection layer 1	Projection layer 2
Suave	1%	65.2	**66.2**	65.1
	10%	74.2	**75.0**	74.8

fit the real distribution. As shown in the table, the semi-supervised fine-tuning recipe enables a larger improvement over the performance after semi-supervised pre-training.

Finally, Table 6 shows how the quality of the learned representations changes at different depths of the projector. It turns out that regardless of the label split, we obtain the most discriminative representations at the first layer of the projection. This is consistent with what was observed by other works that use similar augmentations (e.g., PAWS [6]). We ascribe this behavior to the fact that the last layer tries to build as much invariance as possible to strong augmentations, which hurts the classification accuracy.

6. Conclusion

We have presented a novel approach for semi-supervised learning. Our framework leverages from clustering-based self-supervised methods and adopts a multi-task objective, combining a supervised loss with the unsupervised cross-entropy loss typically adopted for clustering assignments in [15, 16]. Despite its simplicity, we demonstrate that our approach is highly effective, setting a new state-of-the-art for semi-supervised learning on CIFAR-100 and ImageNet.

Acknowledgements. This work was supported by the European Institute of Innovation & Technology (EIT) and the H2020 EU project SPRING (000040103521), funded by the European Commission under GA 871245, and partially supported by the PRIN project LEGO-AI (2020TA3K9N). It was carried out under the "Vision and Learning joint Laboratory" between FBK and UNITN. Karteek Alahari was funded by the ANR grant AVENUE (ANR-18-CE23-0011). Julien Mairal was funded by the ERC grant number 714381 (SOLARIS project) and by ANR 3IA MIAI@Grenoble Alpes (ANR-19-P3IA-0003). Xavier Alameda-Pineda was funded by the ARN grant ML3RI (ANR-19-CE33-0008-01). This project was granted access to the HPC resources of IDRIS under the allocation 2021-[AD011013084] made by GENCI.

References

[1] Elad Amrani, Leonid Karlinsky, and Alex Bronstein. Self-supervised classification network. In *European Conference on Computer Vision*, pages 116–132. Springer, 2022. 5

[2] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020. 3

[3] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations*, 2020. 1, 2, 5

[4] Mahmoud Assran, Nicolas Ballas, Lluis Castrejon, and Michael Rabbat. Supervision accelerates pre-training in contrastive semi-supervised learning of visual representations. *arXiv preprint arXiv:2006.10803*, 2020. 1, 3, 6, 7, 8

[5] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Mike Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. In *European Conference on Computer Vision*, pages 456–473. Springer, 2022. 7

[6] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Armand Joulin, Nicolas Ballas, and Michael Rabbat. Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 2, 3, 5, 6, 7, 8

[7] Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. There are many consistent explanations of unlabeled data: Why you should average. In *International Conference on Learning Representations*, 2019. 3

[8] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, 32, 2019. 1, 2

[9] Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022. 1, 2

[10] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicregl: Self-supervised learning of local visual features. In *NeurIPS*, 2022. 1, 2

[11] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. In *International Conference on Learning Representations*, 2020. 3, 7

[12] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019. 3, 6, 7

[13] Zhaowei Cai, Avinash Ravichandran, Subhransu Maji, Charless Fowlkes, Zhuowen Tu, and Stefano Soatto. Exponential moving average normalization for self-supervised and semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 194–203, 2021. 1, 2, 3, 7

[14] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, 2018. 1, 2, 3, 5

[15] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020. 1, 2, 3, 4, 5, 6, 8

[16] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 1, 2, 3, 5, 7, 8

[17] Dongdong Chen, Wei Wang, Wei Gao, and Zhi-Hua Zhou. Tri-net for semi-supervised deep learning. In *Proceedings of twenty-seventh international joint conference on artificial intelligence*, pages 2014–2020, 2018. 3

[18] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 1, 2, 7

[19] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020. 1, 3, 6

[20] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 7

[21] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021. 1, 2

[22] Yanbei Chen, Xiatian Zhu, Wei Li, and Shaogang Gong. Semi-supervised learning under class distribution mismatch. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3569–3576, 2020. 3

[23] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013. 5

[24] Victor Guilherme Turrisi da Costa, Enrico Fini, Moin Nabi, Nicu Sebe, and Elisa Ricci. solo-learn: A library of self-supervised methods for visual representation learning. *J. Mach. Learn. Res.*, 23:56–1, 2022. 7

[25] Victor G Turrisi da Costa, Giacomo Zara, Paolo Rota, Thiago Oliveira-Santos, Nicu Sebe, Vittorio Murino, and Elisa Ricci. Dual-head contrastive domain adaptation for video action recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1181–1190, 2022. 1

[26] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255, 2009. 6

[27] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner,

Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2, 5, 7

[28] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9588–9597, 2021. 7

[29] Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In *International Conference on Machine Learning*, pages 3015–3024. PMLR, 2021. 1, 2

[30] Enrico Fini, Victor G Turrisi da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. Self-supervised models are continual learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 1

[31] Enrico Fini, Enver Sangineto, Stéphane Lathuilière, Zhun Zhong, Moin Nabi, and Elisa Ricci. A unified objective for novel class discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 1, 4

[32] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 1, 2, 7

[33] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304, 2010. 2

[34] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 2

[35] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 1, 2

[36] 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*, pages 770–778, 2016. 5

[37] Tri Huynh, Simon Kornblith, Matthew R Walter, Michael Maire, and Maryam Khademi. Boosting contrastive self-supervised learning with false negative cancellation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2785–2795, 2022. 2, 7

[38] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020. 2

[39] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 2020. 8

[40] Jiwon Kim, Youngjo Min, Daehwan Kim, Gyuseong Lee, Junyoung Seo, Kwangrok Ryoo, and Seungryong Kim. Conmatch: Semi-supervised learning with confidence-guided consistency regularization. In *European Conference on Computer Vision*, 2022. 3, 6, 7

[41] Alex Krizhevsky, Geoffrey Hinton, et al. *Learning multiple layers of features from tiny images*. Toronto, ON, Canada, 2009. 6

[42] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*, 2017. 3, 7

[43] 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*, 2013. 3

[44] Junnan Li, Caiming Xiong, and Steven CH Hoi. Comatch: Semi-supervised learning with contrastive graph regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9475–9484, 2021. 3, 7

[45] Junnan Li, Pan Zhou, Caiming Xiong, and Steven Hoi. Prototypical contrastive learning of unsupervised representations. In *International Conference on Learning Representations*, 2021. 2, 5

[46] Zicheng Liu, Siyuan Li, Ge Wang, Cheng Tan, Lirong Wu, and Stan Z Li. Decoupled mixup for data-efficient learning. *arXiv preprint arXiv:2203.10761*, 2022. 3, 6, 7

[47] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. Distributional smoothing with virtual adversarial training. In *International Conference on Learning Representations*, 2016. 3

[48] Islam Nassar, Samitha Herath, Ehsan Abbasnejad, Wray Buntine, and Gholamreza Haffari. All labels are not created equal: Enhancing semi-supervision via label grouping and co-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7241–7250, 2021. 3

[49] Sungrae Park, JunKeon Park, Su-Jin Shin, and Il-Chul Moon. Adversarial dropout for supervised and semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 3

[50] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. Meta pseudo labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 3, 7

[51] Siyuan Qiao, Wei Shen, Zhishuai Zhang, Bo Wang, and Alan Yuille. Deep co-training for semi-supervised image recognition. In *Proceedings of the european conference on computer vision (eccv)*, pages 135–152, 2018. 3

[52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 6

[53] 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, 2016. 3

[54] Hugo Schmutz, Olivier Humbert, and Pierre-Alexandre Mattei. Don't fear the unlabelled: safe deep semi-supervised learning via simple debiaising. *arXiv preprint arXiv:2203.07512*, 2022. 3

[55] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020. 3, 6, 7

[56] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015. 6

[57] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017. 3, 7

[58] Vikas Verma, Kenji Kawaguchi, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. In *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI*, pages 3635–3641, 2019. 3, 6

[59] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010. 2

[60] Jianfeng Wang, Thomas Lukasiewicz, Daniela Massiceti, Xiaolin Hu, Vladimir Pavlovic, and Alexandros Neophytou. Np-match: When neural processes meet semi-supervised learning. In *International Conference on Machine Learning*, pages 22919–22934. PMLR, 2022. 3, 6, 7

[61] Xiao Wang, Daisuke Kihara, Jiebo Luo, and Guo-Jun Qi. Enaet: A self-trained framework for semi-supervised and supervised learning with ensemble transformations. *IEEE Transactions on Image Processing*, 30:1639–1647, 2020. 2, 3

[62] Xudong Wang, Zhirong Wu, Long Lian, and Stella X Yu. Debiased learning from naturally imbalanced pseudo-labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 3, 6, 7

[63] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018. 2

[64] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33, 2020. 3, 7

[65] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020. 3

[66] Yi Xu, Lei Shang, Jinxing Ye, Qi Qian, Yu-Feng Li, Baigui Sun, Hao Li, and Rong Jin. Dash: Semi-supervised learning with dynamic thresholding. In *International Conference on Machine Learning*. PMLR, 2021. 3, 7

[67] Fan Yang, Kai Wu, Shuyi Zhang, Guannan Jiang, Yong Liu, Feng Zheng, Wei Zhang, Chengjie Wang, and Long Zeng. Class-aware contrastive semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14421–14430, 2022. 3, 6, 7

[68] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017. 6

[69] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 6

[70] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 5

[71] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021. 1, 2, 7

[72] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1476–1485, 2019. 1, 2, 3, 6, 7

[73] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34:18408–18419, 2021. 3, 6, 7

[74] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 3, 6

[75] Liheng Zhang and Guo-Jun Qi. Wcp: Worst-case perturbations for semi-supervised deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3912–3921, 2020. 3

[76] Mingkai Zheng, Shan You, Lang Huang, Fei Wang, Chen Qian, and Chang Xu. Simmatch: Semi-supervised learning with similarity matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14471–14481, 2022. 3, 6, 7

[77] Zhun Zhong, Enrico Fini, Subhankar Roy, Zhiming Luo, Elisa Ricci, and Nicu Sebe. Neighborhood contrastive learning for novel class discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10867–10875, 2021. 1