

PROTOCON: Pseudo-label Refinement via Online Clustering and Prototypical Consistency for Efficient Semi-supervised Learning

Islam Nassar¹ Munawar Hayat¹ Ehsan Abbasnejad² Hamid RezaTofighi¹
 Gholamreza Haffari¹

¹ Data Science and AI Department, Monash University, Australia – firstname.lastname@monash.edu

² Australian Institute for Machine Learning, The University of Adelaide, Australia – firstname.lastname@adelaide.edu.au

Abstract

Confidence-based pseudo-labeling is among the dominant approaches in semi-supervised learning (SSL). It relies on including high-confidence predictions made on unlabeled data as additional targets to train the model. We propose PROTOCON, a novel SSL method aimed at the less-explored label-scarce SSL where such methods usually underperform. PROTOCON refines the pseudo-labels by leveraging their nearest neighbours' information. The neighbours are identified as the training proceeds using an online clustering approach operating in an embedding space trained via a prototypical loss to encourage well-formed clusters. The online nature of PROTOCON allows it to utilise the label history of the entire dataset in one training cycle to refine labels in the following cycle without the need to store image embeddings. Hence, it can seamlessly scale to larger datasets at a low cost. Finally, PROTOCON addresses the poor training signal in the initial phase of training (due to fewer confident predictions) by introducing an auxiliary self-supervised loss. It delivers significant gains and faster convergence over state-of-the-art across 5 datasets, including CIFARs, ImageNet and DomainNet.

1. Introduction

Semi-supervised Learning (SSL) [10, 40] leverages unlabeled data to guide learning from a small amount of labeled data; thereby, providing a promising alternative to costly human annotations. In recent years, SSL frontiers have seen substantial advances through confidence-based pseudo-labeling [21, 22, 38, 42, 43]. In these methods, a model iteratively generates pseudo-labels for unlabeled samples which are then used as targets to train the model. To overcome confirmation bias [1, 27] *i.e.*, the model being biased by training on its own wrong predictions, these methods only retain samples with high confidence predictions for pseudo-labeling; thus ensuring that only reliable samples are used to train the model. While confidence works well in moderately labeled data regimes, it usually strug-

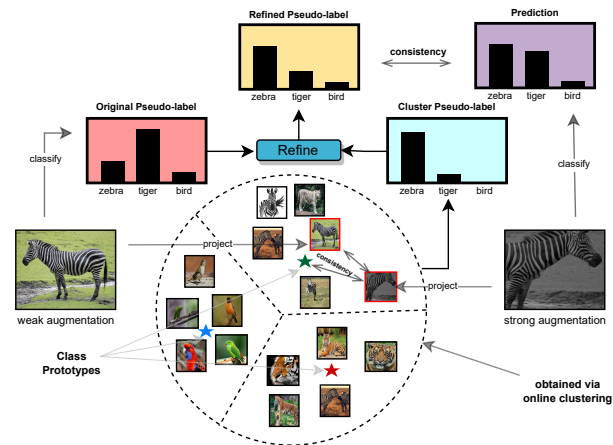


Figure 1. PROTOCON refines a pseudo-label of a given sample by knowledge of its neighbours in a prototypical embedding space. Neighbours are identified in an online manner using constrained K-means clustering. Best viewed zoomed in.

gles in label-scarce settings¹. This is primarily because the model becomes over-confident about the more distinguishable classes [17, 28] faster than others, leading to a collapse.

In this work, we propose PROTOCON, a novel method which addresses such a limitation in label-scarce SSL. Its key idea is to complement confidence with a label refinement strategy to encourage more accurate pseudo-labels. To that end, we perform the refinement by adopting a co-training [5] framework: for each image, we obtain two different labels and combine them to obtain our final pseudo-label. The first is the model’s softmax prediction, whereas the second is an aggregate pseudo-label describing the image’s neighbourhood based on the pseudo-labels of other images in its vicinity. However, a key requirement for the success of co-training is to ensure that the two labels are obtained using sufficiently different image representations [40] to allow the model to learn based on their disagreements. As such, we employ a non-linear projection to map our encoder’s representation into a different embed-

¹We denote settings with less than 10 images per class as “label-scarce.”

ding space. We train this projector jointly with the model with a prototypical consistency objective to ensure it learns a different, yet relevant, mapping for our images. Then we define the neighbourhood pseudo-label based on the vicinity in that embedding space. In essence, we minimise a sample bias by smoothing its pseudo-label in class space via knowledge of its neighbours in the prototypical space.

Additionally, we design our method to be fully online, enabling us to scale to large datasets at a low cost. We identify neighbours in the embedding space on-the-fly as the training proceeds by leveraging online K-means clustering. This alleviates the need to store expensive image embeddings [22], or to utilise offline nearest neighbour retrieval [23, 48]. However, applying naive K-means risks collapsing to only a few imbalanced clusters making it less useful for our purpose. Hence, we employ a constrained objective [6] lower bounding each cluster size; thereby, ensuring that each sample has enough neighbours in its cluster. We show that the online nature of our method allows it to leverage the entire prediction history in one epoch to refine labels in the subsequent epoch at a fraction of the cost required by other methods and with a better performance.

PROTOCON’s final ingredient addresses another limitation of confidence-based methods: since the model only retains high confident samples for pseudo-labeling, the initial phase of the training usually suffers from a weak training signal due to fewer confident predictions. In effect, this leads to only learning from the very few labeled samples which destabilises the training potentially due to overfitting [25]. To boost the initial training signal, we adopt a self-supervised instance-consistency [9, 15] loss applied on samples that fall below the threshold. Our choice of loss is more consistent with the classification task as opposed to contrastive instance discrimination losses [11, 16] which treat each image as its own class. This helps our method to converge faster without loss of accuracy.

We demonstrate PROTOCON’s superior performance against comparable state-of-the-art methods on 5 datasets including CIFAR, ImageNet and DomainNet. Notably, PROTOCON achieves 2.2%, 1% improvement on the SSL ImageNet protocol with 0.2% and 1% of the labeled data, respectively. Additionally, we show that our method exhibits faster convergence and more stable initial training compared to baselines, thanks to our additional self-supervised loss. In summary, our contributions are:

- We propose a memory-efficient method addressing confirmation bias in label-scarce SSL via a novel label refinement strategy based on co-training.
- We improve training dynamics and convergence of confidence-based methods by adopting self-supervised losses to the SSL objective.
- We show state-of-the-art results on 5 SSL benchmarks.

2. Background

We begin by reviewing existing SSL approaches with a special focus on relevant methods in the low-label regime.

Confidence-based pseudo-labeling is an integral component in most of recent SSL methods [20, 22, 27, 38, 42]. However, recent research shows that using a fixed threshold underperforms in low-data settings because the model collapses to the few easy-to-learn classes early in the training. Some researchers combat this effect by using class- [47] or instance-based [44] adaptive thresholds, or by aligning [3] or debiasing [42] the pseudo-label distribution by keeping a running average of pseudo-labels to avoid the inherent imbalance in pseudo-labels. Another direction focuses on pseudo-label refinement, whereby the classifier’s predictions are adjusted by training another projection head on an auxiliary task such as weak-supervision via language semantics [27], instance-similarity matching [48], or graph-based contrastive learning [22]. Our method follows the refinement approach, where we employ online constrained clustering to leverage nearest neighbours information for refinement. Different from previous methods, our method is fully online and hence allows using the entire prediction history in one training epoch to refine pseudo-labels in the subsequent epoch with minimal memory requirements.

Consistency Regularization combined with pseudo-labeling underpins many recent state-of-the-art SSL methods [4, 20, 22, 24, 35, 38, 43]; it exploits the smoothness assumption [40] where the model is expected to produce similar pseudo-labels for minor input perturbations. The seminal FixMatch [38] and following work [22, 27, 42] leverage this idea by obtaining pseudo-labels through a weak form of augmentation and applying the loss against the model’s prediction for a strong augmentation. Our method utilises a similar approach, but different from previous work, we additionally apply an instance-consistency loss in our projection embedding space.

Semi-supervision via self-supervision is gaining recent popularity due to the incredible success of self-supervised learning for model pretraining. Two common approaches are: 1) performing self-supervised pretraining followed by supervised fine-tuning on the few labeled samples [9, 11, 12, 15, 26], and 2) including a self-supervised loss to the semi-supervised objective to enhance training [22, 25, 41, 46, 48]. However, the choice of the task is crucial: tasks such as instance discrimination [11, 16], which treats each image as its own class, can hurt semi-supervised image classification as it partially conflicts with it. Instead, we use an instance-consistency loss akin to that of [9] to boost the initial training signal by leveraging samples which are not retained for pseudo-labeling in the early phase of the training.

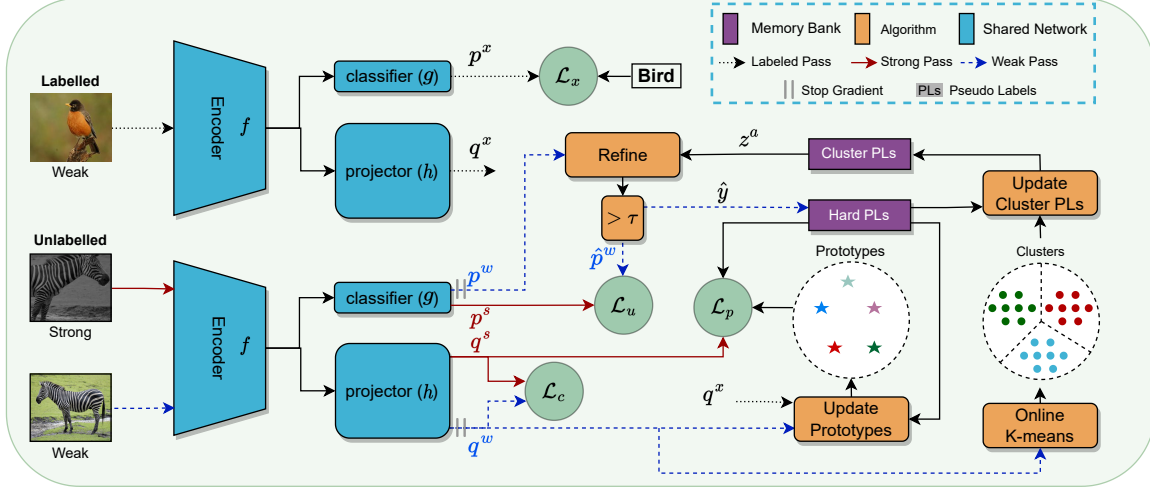


Figure 2. **Method overview.** A soft pseudo-label p^w is first obtained based on the weak view. Then it is refined using the sample’s cluster pseudo-label z^a before using it as target in \mathcal{L}_u . Clustering assignments a are calculated online using the projections of the weak samples q^w in the embedding space h which is trained via a prototypical loss \mathcal{L}_p . Prototype targets are updated once after each epoch by averaging the accumulated projections of reliable samples for each class throughout the epoch. Cluster pseudo-labels are updated after each epoch using the cluster assignments/scores of all the samples and their respective hard pseudo-labels \hat{y} . Finally, the self-supervised loss \mathcal{L}_c ensures consistency between projections q^s and q^w .

3. PROTOCON

Preliminaries. We consider a semi-supervised image classification problem, where we train a model using M labeled samples and N unlabeled samples, where $N \gg M$. We use mini-batches of labeled instances, $\mathcal{X} = \{(x_j, y_j)\}_{j=1}^B$ and unlabeled instances, $\mathcal{U} = \{u_i\}_{i=1}^{\mu \cdot B}$, where the scalar μ denotes the ratio between the number of unlabeled and labeled examples in a given batch, and y is the one-hot vector of the class label $c \in \{1, \dots, C\}$. We employ an encoder network f to get latent representations $f(\cdot)$. We attach a softmax classifier $g(\cdot)$, which produces a distribution over classes $p = g \circ f$. Moreover, we attach a projector $h(\cdot)$, an MLP followed by an ℓ_2 norm layer, to get a normalised embedding $q \in \mathbb{R}^d = h \circ f$. Following [38], we apply weak augmentations $\mathcal{A}_w(\cdot)$ on all images and an additional strong augmentation [13] $\mathcal{A}_s(\cdot)$ only on unlabeled ones.

Motivation. Our aim is to refine pseudo-labels before using them to train our model in order to minimise confirmation bias in label-scarce SSL. We achieve this via a co-training approach (see Fig. 2). For each image, we obtain two pseudo-labels and combine them to obtain our final pseudo-label \hat{p}^w . The first is the classifier softmax prediction p^w based on a weakly augmented image, whereas the second is an aggregate pseudo-label z^a describing the sample’s neighbourhood. To ensure the two labels are based on sufficiently different representations, we define an image neighbourhood² via online clustering in an embedding space obtained via projector h and trained for prototypical consistency

instead of class prediction. Projector h and classifier g are jointly trained with the encoder f , while interacting over pseudo-labels. The classifier is trained with pseudo-labels which are refined based on their nearest neighbours in the embedding space, whereas the projector h is trained using prototypes obtained based on the refined pseudo-labels to impose structure on the embedding space.

Prototypical Space. Here, we discuss our procedure to learn our embedding space defined by h . Inspired by prototypical learning [37], we would like to encourage well-clustered image projections in our space by attracting samples to their class prototypes and away from others. Hence, we employ a contrastive objective using the class prototypes as targets rather than the class labels. We calculate class prototypes at the end of a given epoch by knowledge of the “reliable” samples in the previous epoch. Specifically, by employing a memory bank of $\mathcal{O}(2N)$, we keep track of samples hard pseudo-labels $\{\hat{y}_i = \arg \max(\hat{p}_i^w) \forall u_i \in \mathcal{U}\}$ in a given epoch; as well as a reliability indicator for each sample $\eta_i = \mathbb{1}(\max(\hat{p}_i^w) \geq \tau)$ denoting if its max prediction exceeds the confidence threshold τ . Subsequently, we update the prototypes $\mathcal{P} \in \mathbb{R}^{C \times d}$ as the average projections (accumulated over the epoch) of labeled images and reliable unlabeled images. Formally, let $\mathcal{I}_c^x = \{i \mid \forall x_i \in \mathcal{X}, y_i = c\}$ be the indices of labelled instances with true class c , and $\mathcal{I}_c^w = \{i \mid \forall u_i \in \mathcal{U}, \eta_i = 1, \hat{y}_i = c\}$ be the indices of the reliable unlabelled samples with hard pseudo-label c . The normalised prototype for class c can then be obtained as per:

$$\bar{\mathcal{P}}_c = \frac{\sum_{i \in \mathcal{I}_c^x \cup \mathcal{I}_c^w} \mathbf{q}_i}{|\mathcal{I}_c^x| + |\mathcal{I}_c^w|}, \quad \mathcal{P}_c = \frac{\bar{\mathcal{P}}_c}{\|\bar{\mathcal{P}}_c\|_2} \quad (1)$$

²We use “neighbourhood” and “cluster” interchangeably.

Algorithm 1 Pseudo-code of one epoch of PROTOCON

```

# f, g, h: encoder, classifier, and projector
# b_x: labeled batch
# b_w, b_s: weak, strong unlabeled batches
# u_id: unique index of unlabeled samples
# N, C: num unlabeled samples, num classes
# CA: cluster assignment bank (N x 2)
# CPL: clusters pseudo-label bank (N x C)
# PH: samples pseudo-label bank (N x 1)
# Q: cluster centers
# P: class prototypes
# P_acc: prototypes accumulator
# alpha: pseudo-label refinement ratio

for b_x, b_w, b_s, u_id in loader:
  # forward images and obtain p and q
  p_x, p_w, p_s = f(g(b_x, b_w, b_s))
  q_x, q_w, q_s = f(h(b_x, b_w, b_s))
  # calculate and save cluster assignment
  CA[u_id] = calc_clust_assignmnt(q_w, Q) # Eqn.5
  # update centers and dual variables
  Q = update_cluster_centers(q_w) # Eqn. 6 & 7
  #retrieve cluster pseudo-labels from previous epoch
  z = CPL[CA[u_id]]
  # refine p_w
  p_w_hat = alpha*p_w + (1 - alpha)*z # Eqn. 9
  # save hard pseudo-labels
  PH[u_id] = argmax(p_w_hat)
  # accumulate prototypes (of reliable samples only)
  P_acc = accum_prototypes(q_x, q_w, PH[u_id])
  # apply losses (except in first epoch)
  Lx, Lp, Lu, Lc = backward_losses() # Eqn. 2, 10-12
# after each epoch
P = update_prototypes(P_acc) # Eqn. 1
CPL = calc_cluster_pseudo_labels(CA, PH) # Eqn. 8

```

Subsequently, in the following epoch, we minimize the following contrastive prototypical consistency loss on unlabeled samples:

$$\mathcal{L}_p = -\frac{1}{\mu B} \sum_{i=1}^{\mu B} \log \frac{\exp(\mathbf{q}_i^s \cdot \mathcal{P}_{\hat{y}_i} / T)}{\sum_{c=1}^C \exp(\mathbf{q}_i^s \cdot \mathcal{P}_c / T)}, \quad (2)$$

where T is a temperature parameter. Note that the loss is applied against the projection of the strong augmentations to achieve consistency regularisation as in [38].

Online Constrained K-means Here, the goal is to cluster instances in the prototypical space as a training epoch proceeds, so the cluster assignments (capturing the neighbourhood of each sample) are used to refine their pseudo-labels in the following epoch. We employ a mini-batch version of K-means [36]. To avoid collapsing to one (or a few) imbalanced clusters, we ensure that each cluster has sufficient samples by enforcing a constraint on the lowest acceptable cluster size. Given our N unlabeled projections, we cluster them into K clusters defined by centroids $\mathcal{Q} = [\mathbf{c}_1, \dots, \mathbf{c}_K] \in \mathbb{R}^{d \times K}$. We use the constrained K-means objective proposed by [6]:

$$\min_{\mathcal{Q}, \mu \in \Delta} \sum_{i=1, k=1}^{i=N, k=K} \mu_{i,k} \|\mathbf{q}_i - \mathbf{c}_k\|_2^2 \quad \text{s.t.} \quad \forall k \quad \sum_{i=1}^N \mu_{i,k} \geq \gamma \quad (3)$$

where γ is the lower-bound of cluster size, $\mu_{i,k}$ is the assignment of the i -th unlabeled sample to the k -th cluster, and $\Delta = \{\mu | \forall i, \sum_k \mu_{i,k} = 1, \forall i, k, \mu_{i,k} \in [0, 1]\}$ is the domain of μ . Subsequently, to solve Eqn. 3 in an online mini-batch manner, we adopt the alternate solver proposed in [32]. For a fixed \mathcal{Q} , the problem for updating μ can be

simplified as an assignment problem. By introducing dual variables ρ_k for each constraint $\sum_i \mu_{i,k} \geq \gamma$, the assignment can be obtained by solving the problem:

$$\max_{\mu_i \in \Delta} \sum_k s_{i,k} \mu_{i,k} + \sum_k \rho_k^{t-1} \mu_{i,k} \quad (4)$$

where $s_{i,k} = \mathbf{q}_i^\top \mathbf{c}_k$ is the similarity between the projection of unlabeled sample \mathbf{u}_i and the k -th cluster centroid, and t is the mini-batch iteration counter. Eqn. 4 can then be solved with the closed-form solution:

$$\mu_{i,k} = \begin{cases} 1 & k = \arg \max_k s_{i,k} + \rho_k^{t-1} \\ 0 & \text{o.w.} \end{cases} \quad (5)$$

After assignment, dual variables are updated as³:

$$\rho_k^t = \max\{0, \rho_k^{t-1} - \lambda \frac{1}{B} \sum_{i=1}^B (\mu_{i,k}^t - \frac{\gamma}{N})\} \quad (6)$$

where λ is the dual learning rate. Finally, we update the cluster centroids after each mini-batch⁴ as:

$$\bar{\mathbf{c}}_k^t = \frac{\sum_i \mu_{i,k}^t \mathbf{q}_i^t}{\sum_i \mu_{i,k}^t}, \quad \mathbf{c}_k^t = \frac{\bar{\mathbf{c}}_k^t}{\|\bar{\mathbf{c}}_k^t\|_2} \quad (7)$$

where m denotes the total number of received instances until the t -th mini-batch. Accordingly, we maintain another memory bank ($\mathcal{O}(2N)$) to store two values for each unlabeled instance: its cluster assignment in the current epoch $a(i) = \{k | \mu_{i,k} = 1\}$ and the similarity score $s_{i,a(i)}$ (*i.e.* the distance to its cluster centroid).

Cluster Pseudo-labels are computed at end of each epoch by querying the memory banks. The purpose is to obtain a distribution over classes C for each of our clusters based on its members. For a given cluster k , we obtain its label $\mathbf{z}^k = [z_1^k, \dots, z_C^k]$ as the average of the pseudo-labels of its cluster members weighted by their similarity to its centroid. Concretely, let $\mathcal{I}_c^k = \{i | \forall \mathbf{u}_i \in \mathcal{U}, a(i) = k, \hat{y}_i = c\}$ be the indices of unlabeled samples which belong to cluster k and have a hard pseudo-label c . The probability of cluster k 's members belonging to class c is given as:

$$z_c^k = \frac{\sum_{i \in \mathcal{I}_c^k} s_{i,a(i)}}{\sum_{b=1}^C \sum_{j \in \mathcal{I}_b^k} s_{j,a(j)}} \quad (8)$$

Refining Pseudo-labels. At any given epoch, we now have two pseudo-labels for an image \mathbf{u}_i : the unrefined pseudo-label \mathbf{p}_i^w as well as a cluster pseudo-label $\mathbf{z}^{a(i)}$ summarising its prototypical neighbourhood in the previous epoch. Accordingly, we apply our refinement procedure as follows:

³Refer to [32] and supplements for proofs of optimality and more details.

⁴See supplements for a discussion about updating the centers every mini-batch opposed to every epoch.

first, as recommended by [3, 22], we perform distribution alignment ($DA(\cdot)$) to encourage the marginal distribution of pseudo-labels to be close to the marginal of ground-truth labels⁵, then we refine the aligned pseudo-label as per:

$$\hat{\mathbf{p}}_i^w = \alpha \cdot DA(\mathbf{p}_i^w) + (1 - \alpha) \cdot \mathbf{z}^{a(i)} \quad (9)$$

Here, the second term acts as a regulariser to encourage $\hat{\mathbf{p}}^w$ to be similar to its cluster members’ and α is a trade-off scalar parameter. Importantly, the refinement here leverages information based on the entire training set last-epoch information. This is in contrast to previous work [22, 48] which only stores a limited history of soft pseudo-labels for refinement, due to more memory requirement ($\mathcal{O}(N \times C)$).

Classification Loss. With the refined pseudo-label, we apply the unlabeled loss against the model prediction for the strong augmentation as per:

$$\mathcal{L}_u = \frac{1}{\mu_B} \sum_{i=1}^{\mu_B} \eta_i \cdot \text{CE}(\hat{\mathbf{p}}_i^w, \mathbf{p}_i^s), \quad (10)$$

where CE denotes cross-entropy. However, unlike [4, 38], we do not use hard pseudo-labels or sharpening, but instead use the soft pseudo-label directly. Also, we apply a supervised classification loss over the labeled data as per:

$$\mathcal{L}_x = \frac{1}{B} \sum_{i=1}^B \text{CE}(\mathbf{y}_i, \mathbf{p}_i^x), \quad (11)$$

Self-supervised Loss. Since we use confidence as a measure of reliability (see Eqn. 10), early epochs of training suffer from limited supervisory signal when the model is not yet confident about unlabeled samples, leading to slow convergence and unstable training. Our final ingredient addresses this by introducing a consistency loss in the prototypical space on samples which fall below the confidence threshold τ . We draw inspiration from instance-consistency self-supervised methods such as BYOL [15] and DINO [9]. In contrast to contrastive instance discrimination [11, 16], the former imposes consistency between two (or more) views of an image without using negative samples. Thereby, we found it to be more aligned with classification tasks than the latter. Formally, we treat the projection q as soft classes score over d dimensions, and obtain a distribution over these classes via a sharpened softmax ($SM(\cdot)$). We then enforce consistency between the weak and strong views as per:

$$\mathcal{L}_c = \frac{1}{\mu_B} \sum_{i=1}^{\mu_B} (1 - \eta_i) \cdot \text{CE}(SM(\mathbf{q}_i^w/5T), SM(\mathbf{q}_i^s/T)) \quad (12)$$

Note that, as in DINO [9], we sharpen the target distribution more than the source’s to encourage entropy minimization [14]. Unlike DINO, we do not use a separate EMA

⁵ $DA(\mathbf{p}^w) = \mathbf{p}^w / \mathbf{p}^{\bar{w}}$, where $\mathbf{p}^{\bar{w}}$ is a running average of \mathbf{p}^w during training.

model to produce the target, we just use the output of the model for the weak augmentation. Note that this does not lead to representation collapse [15] because the network is also trained with additional semi-supervised losses.

Final Objective. We train our model using a linear combination of all four losses $\mathcal{L} = \mathcal{L}_x + \lambda_u \mathcal{L}_u + \lambda_p \mathcal{L}_p + \lambda_c \mathcal{L}_c$. Empirically, we find that fixing $\forall \lambda = 1$, the coefficients to modulate each loss, works well across different datasets. Algorithm 1 describes one epoch of PROTOCON training.

3.1. Design Considerations

Number of Clusters is a crucial parameter in our approach. In essence, we refine a sample prediction obtained by the classifier by aggregating information from its n nearest neighbours. However, naively doing nearest-neighbour retrieval has two limitations: 1) it requires storing image features throughout an epoch which is memory expensive; and 2) it requires a separate offline nearest-neighbour retrieval step. Instead, we leverage online clustering to identify nearest-neighbours on-the-fly. To avoid tuning K for each dataset, we tuned n once instead, then K can be simply calculated as $K = N/n$. Additionally, we fix $\gamma = 0.9n$ to ensure that each cluster contains sufficient samples to guarantee the quality of the cluster pseudo-label while relaxing clusters to not necessarily be equi-partitioned. Empirically, we found that using $n = 250$ works reasonably well across datasets. To put it in context, this corresponds to $K = 4800$ for ImageNet, and $K = 200$ for CIFAR datasets.

Multi-head Clustering is another way to ensure robustness of our cluster pseudo-labels. To account for K-means stochastic nature, we can employ multi-head clustering to get different cluster assignments based on each head, at negligible cost. Subsequently, we can average the cluster pseudo-labels across the different heads. In practice, we find that for large datasets *e.g.* ImageNet, cluster assignments slightly vary between heads so it is useful to use dual heads, while for smaller datasets, a single head is sufficient.

Memory Analysis. PROTOCON is particularly useful due to its ability to leverage the entire prediction history in an epoch to approximate class density over neighbourhoods (represented by cluster pseudo-labels) with low memory cost. Particularly, it requires an overall of $\mathcal{O}(4N + K \times C)$: $4N$ to store hard pseudo-labels, reliability, cluster assignments, and similarity scores; and $K \times C$ to store the cluster pseudo-labels. In contrast, if we were to employ a naive offline refinement approach, this would require $\mathcal{O}(N \times d)$ to store the image embeddings for an epoch. For ImageNet dataset this translates to 9.6M memory units for PROTOCON opposed to 153.6M for the naive approach⁶ which is a $16\times$ reduction in memory; beside, eliminating the additional time needed to perform nearest neighbour retrieval.

⁶considering $d = 128$

Table 1. CIFAR and Mini-ImageNet accuracy for different amounts of labeled samples averaged over 5 different splits. All results are produced using the same codebase and same splits.

Total labeled samples	CIFAR-10			CIFAR-100			Mini-ImageNet	
	20	40	80	200	400	800	400	1000
FixMatch [38]	82.32±9.77	86.29±4.50	92.06±0.88	35.37±5.68	51.15±1.75	61.32±0.92	17.18±6.22	39.03±3.99
FixMatch + DA [3, 38]	83.84±8.35	86.98±3.40	92.29±0.86	41.28±6.03	52.65±2.32	62.12±0.79	19.40±5.87	40.92±4.71
CoMatch [22]	87.37±8.47	93.09±1.39	93.97±0.62	47.92±4.83	58.17±3.52	66.15±0.71	21.29±6.19	40.98±3.52
SimMatch [48]	89.31±7.73	94.51±2.56	94.89±1.32	46.01±6.12	57.95±2.37	65.50±0.93	25.75±5.90	39.76±3.77
FixMatch + DB [42]	89.02±6.37	94.60±1.31	95.60±0.12	46.36±5.05	57.88±3.34	64.84±0.85	27.37±7.01	41.05±3.34
PROTOCON	90.51±4.02	95.20±1.8	96.11±0.20	48.25±4.87	59.53±2.94	65.91±0.57	29.15±6.98	45.83±4.15
<i>delta against best baseline</i>	+1.20	+0.60	+0.51	+0.33	+1.36	-0.24	+1.78	+4.78

Table 2. DomainNet accuracy for 2, 4, and 8 labels per class.

Total labeled samples	Clipart			Sketch		
	690	1380	2760	690	1380	2760
FixMatch [38]	30.21	41.21	51.29	12.73	21.65	33.07
CoMatch [22]	35.49	48.62	54.98	24.30	33.71	41.02
FixMatch + DB [42]	38.97	51.44	58.31	25.34	35.58	43.98
PROTOCON	43.72	55.66	61.32	33.94	43.51	50.88
<i>delta</i>	+4.75	+4.22	+3.01	+8.60	+7.93	+6.90

4. Experiments

We begin by validating PROTOCON’s performance on multiple SSL benchmarks against state-of-the-art methods. Then, we analyse the main components of PROTOCON to verify their contribution towards the overall performance, and we perform ablations on important hyperparameters.

4.1. Experimental Settings

Datasets. We evaluate PROTOCON on five SSL benchmarks. Following [1, 38, 43], we evaluate on **CIFAR-10(100)** [19] datasets, which comprises 50,000 images of 32x32 resolution of 10(100) classes; as well as the more challenging **Mini-ImageNet** dataset proposed in [33], having 100 classes with 600 images per class (84x84 each). We use the same train/test split as in [18] and create splits for 4 and 10 labeled images per class to test PROTOCON in the low-label regime. We also test PROTOCON’s performance on the **DomainNet** [30] dataset, which has 345 classes from six visual domains: *Clipart*, *Infograph*, *Painting*, *Quick-draw*, *Real*, and *Sketch*. We evaluate on the *Clipart* and *Sketch* domains to verify our method’s efficacy in different visual domains and on imbalanced datasets. Finally, we evaluate on **ImageNet** [34] SSL protocol as in [2, 8, 9, 11]. In all our experiments, we focus on the low-label regime.

Implementation Details. For CIFAR-10(100), we follow previous work and use WideResnet-28-2(28-8) [45] as our encoder. We use a 2-layer projection MLP with an embedding dimension $d = 64$. The models are trained using SGD with a momentum of 0.9 and weight decay of 0.0005(0.001) using a batch size of 64 and $\mu = 7$. We set the threshold $\tau = 0.95$ and train our models for 1024 epochs for a

fair comparison with the baselines. However, we note that our model needs substantially fewer epochs to converge (see Fig. 3-b and c). We use a learning rate of 0.03 with a cosine decay schedule. We use random horizontal flips for weak augmentations and RandAugment [13] for strong ones. For the larger datasets: ImageNet and DomainNet, we use a Resnet-50 encoder and $d = 128$, $\mu = 5$ and $\tau = 0.7$ and follow the same hyperparameters as in [38] except that we use SimCLR [11] augmentations for the strong view. For PROTOCON-specific hyperparameters, we consistently use the same parameters across all experiments: we set n to 250 (corresponding to $K=200$ for CIFARs, and Mini-ImageNet, and 4800 for ImageNet), and dual learning rate $\lambda = 20$, mixing ratio $\alpha = 0.8$, and temperature $T = 0.1$.

Baselines. Since our method bears the most resemblance with CoMatch [22], we compare against it in all our experiments. CoMatch uses graph contrastive learning to refine pseudo-labels but uses a memory bank to store the last n -samples embeddings to build the graph. Additionally, we compare with state-of-the-art SSL method (DebiasPL) [42], which proposes a pseudo-labeling debiasing plug-in to work with various SSL methods in addition to an adaptive margin loss to account for inter-class confounding. Finally, we also compare with the seminal method FixMatch and its variant with Distribution alignment (DA). We follow Oliver et al. [29] recommendations to ensure a fair comparison with the baselines, where we implement/adapt all the baselines using the same codebase to ensure using the same settings across all experiments. As for ImageNet experiments, we also compare with representation learning baselines such as SwAV [8], DINO [9], and SimCLR [11], where we report the results directly from the respective papers. We also include results for PROTOCON and DebiasPL with additional pretraining (using MOCO [16]) and the Exponential Moving Average Normalisation method proposed by [7] to match the settings used in [7, 42].

4.2. Results and Analysis

Results. Similar to prior work, we report the results on the test sets of respective datasets by averaging the results of the last 10 epochs of training. For CIFAR and Mini-ImageNet,

Table 3. SSL results on ImageNet with different percentage of labels. † denotes results produced by our codebase. Other results are reported as appearing in the cited work.

Method	Pre.	Epochs	0.2%	1%	10%
Supervised	✗	300	–	25.4	56.4
<i>Representation learning methods:</i>					
SwAV [8]	✓	800	–	53.9	70.2
SimCLRv2++ [12]	✓	1200	–	60.0	70.5
DINO [9]	✓	300	–	55.1	67.8
PAWS++ [2]	✓	300	–	66.5	75.5
<i>PL & consistency methods:</i>					
MPL [31]	✗	800	–	65.3 [†]	73.9
CoMatch [22]	✗	400	44.3 [†]	66.0	73.6
FixMatch [38]	✗	300	–	51.2	71.5
FMatch + DA [3, 38]	✗	300	41.1 [†]	53.4	71.5 [†]
FMatch + EMAN [7]	✓	850	43.6	60.9	72.6
FMatch + DB [42]	✗	300	45.8 [†]	63.0 [†]	71.7 [†]
FMatch + DB + EMAN [42]	✓	850	47.9	63.1	72.8 [†]
PROTOCON	✗	300	47.8	65.6	73.1
PROTOCON + EMAN [7]	✓	850	50.1	67.2	73.5
<i>delta against best baseline</i>			+2.2	+0.7	-2.0

we report the average and standard deviation over 5 different labeled splits, whereas we report for only 1 split on larger datasets (ImageNet and DomainNet). Different from most previous work, we only focus on the very low-label regime (2, 4, and 8 samples per class, and 0.2% for ImageNet). As shown in Tab. 1 - 3, we observe that PROTOCON outperforms baselines in almost all the cases showing a clear advantage in the low-label regime. It also exhibits less variance across the different splits (and the different runs within each split). These results suggest that besides achieving high accuracy, PROTOCON shows robustness and consistency across splits in low-data regime.

Notably, our method performs particularly well on DomainNet. Unlike ImageNet and CIFARs, DomainNet is an imbalanced dataset, and prior work [39] shows that it suffers from high level of label noise. This shows that our method is also more robust to noisy labels. This can be explained in context of our co-training approach: using the prototypical neighbourhood label to smooth the softmax label is an effective way to minimise the effect of label noise. In line with previous findings [23], since in prototypical learning, all the instances of a given class are used to calculate a class prototype which is then used as a prediction target, it results in representations which are more robust to noisy labels.

Finally, on ImageNet (Tab. 3), we improve upon the closest baseline with gains of 2.2% in the challenging 0.2% setting; whereas we slightly fall behind PAWS [2] in the 10% regime, again confirming our method’s usefulness in the label-scarce scenario.

How does refinement help? First, we would like to investigate the role of pseudo-labeling refinement in improving

SSL performance. Intuitively, since we perform refinement by combining pseudo-labels from two different sources (the classifier predictions in probability space and the cluster labels in the prototypical space), we expect that there will be disagreements between the two and hence considering both the views is the key towards the improved performance. To validate such intuition, we capture a fine-grained view of the training dynamics throughout the first 300 epochs of CIFAR-10 with 40 labeled instances scenario, including: samples’ pseudo-labels before and after refinement as well as their cluster pseudo-labels in each epoch. This enables us to capture disagreements between the two pseudo-label sources up to the individual sample level. In Fig. 3-a, we display the average disagreement between the two sources over the initial phase of the training overlaid with the classifier, cluster and refined pseudo-label accuracy. We observe that initially, the disagreement (dashed black line) is high which corresponds to a larger gap between the accuracies of both heads. As the training proceeds, we observe that disagreement decreases leading to a respective decrease in the gap. Additionally, we witness that the refined accuracy curve (green) is almost always above the individual accuracies (orange and blue) which proves that, indeed, the synergy between the two sources improves the performance.

On the other hand, to get a qualitative understanding of where each of the pseudo-labeling sources helps, we dig deeper to classes and individual samples level where we investigate which classes/samples are the most disagreed-upon (on average) throughout the training. In Fig. 4, we display the most prototypical examples of a given class (middle) as identified by the prototypical scores obtained in the embedding space. We also display the examples which on average are always correctly classified in the prototypical space (right) opposed to those in the classifier space (left). As expected, we find that samples which look more prototypical, albeit with less distinctive features (e.g. blurry), are the ones almost always correctly classified with the prototypical head; whereas, samples which have more distinctive features but are less prototypical are those correctly classified by the discriminative classifier head. This again confirms our intuitions about how co-training based on both sources helps to refine the pseudo-label.

Finally, we ask: is it beneficial to use the entire dataset pseudo-label history to perform refinement or is it sufficient to just use a few samples? To answer this question, we use only a subset of the samples in each cluster (sampled uniformly at random) to calculate cluster pseudo-labels in Eqn. 8. For CIFAR-10 with 20 and 40 labels, we find that this leads to about 1-2% (4-5%) average drop in performance, if we use half (quarter) of the samples in each cluster. This reiterates the usefulness of our approach to leverage the history of all samples (at a lower cost) opposed to a limited history of samples.

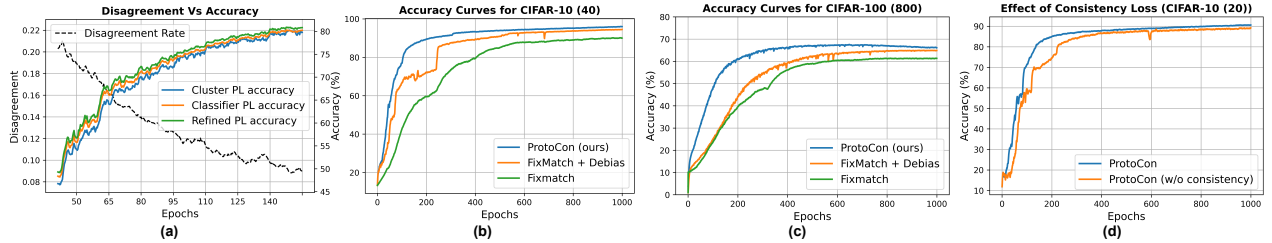


Figure 3. **Analysis Plots.** (a): Average disagreement between cluster and classifier pseudo-labels versus ground truth accuracy of the different pseudo-labels. The accuracy gap between refined pseudo-labels (green) and the cluster’s and classifier’s (blue and orange) decreases with disagreement rate (dashed black) showing that refinement indeed helps. (b), (c): Convergence plots on CIFAR10/100 show that PROTOCON converges faster due to the additional self-supervised training signal. (d): PROTOCON w and w/out consistency loss.

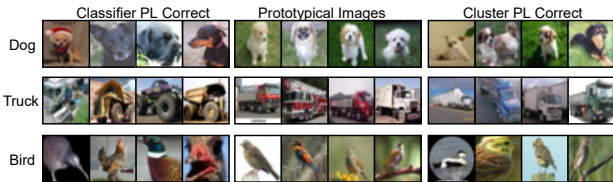


Figure 4. The middle panel shows the most prototypical images of CIFAR-10 classes as identified by our model. Left (resp. right) panels show images which have more accurate classifier (resp. cluster) pseudo-labels. Cluster labels are more accurate for prototypical images while classifier labels are more accurate for images with distinctive features (e.g. truck wheels) even if not so prototypical. Such diversity of views is key to the success of our co-training method.

Role of self-supervised loss. Here, we are interested to tear apart our choice of self-supervised loss and its role towards the performance. To recap, our intuition behind using that loss is to boost the learning signal in the initial phase of the training when the model is still not confident enough to retain samples for pseudo-labeling. As we see in Fig. 3-b and c. there is a significant speed up of our model’s convergence compared to baseline methods with a clear boost in the initial epochs. Additionally, to isolate the potentially confounding effect of our other ingredients, we display in Fig. 3-d the performance of our method with and without the self-supervised loss which leads to a similar conclusion. Finally, to validate our hypothesis that instance-consistency loss is more useful than instance-discrimination, we run a version of PROTOCON with an instance-discrimination loss akin to that of SimCLR. This version completely collapsed and did not converge at all. We attribute this to: 1) as verified by SimCLR authors, such methods work best with large batch sizes to ensure enough negative examples are accounted for; and 2) these methods treat each image as its own class and contrast it against every other image and hence are in direct contradiction with the image classification task; whereas instance-consistency losses only ensure that the representations learnt are invariant to common factors of variations such as: color distortions, orientation, etc.

Table 4. Ablation results. $-\mathcal{L}_*$ denotes that the respective loss is not applied, and **green** marks the best option. Results are average accuracy over 5 runs for CIFAR-10 (80).

Losses	$-\mathcal{L}_c$	$-\mathcal{L}_p$	$-(\mathcal{L}_c, \mathcal{L}_p)$	All
	95.3	94.8	92.3	96.1
Cluster size (n)	50	250	500	1000
	95.7	96.1	94.3	92.1
Refinement Ratio (α)	0.5	0.7	0.8	0.9
	86.7	94.5	96.1	95.2

and are hence more suitable for semi-supervised image classification tasks.

Ablations. Finally, we present an ablation study about the important hyperparameters of PROTOCON. Specifically, we find that n (minimum samples in each cluster) and α (mixing ratio between classifier pseudo-label and cluster pseudo-label) are particularly important. Additionally, we find that the projection dimension needs to be sufficiently large for larger datasets (we use $d = 64$ for CIFARs and 128 for all others). In Tab. 4, we present ablation results on CIFAR-10 with 80 labeled instances.

5. Conclusion

We introduced PROTOCON, a novel SSL learning approach targeted at the low-label regime. Our approach combines co-training, clustering and prototypical learning to improve pseudo-labels accuracy. We demonstrate that our method leads to significant gains on multiple SSL benchmarks and better convergence properties. We hope that our work helps to commodify deep learning in domains where human annotations are expensive to obtain.

Acknowledgement. This work was partly supported by DARPA’s Learning with Less Labeling (LwLL) program under agreement FA8750-19-2-0501. I. Nassar is supported by the Australian Government Research Training Program (RTP) Scholarship, and M. Hayat is supported by the ARC DECRA Fellowship DE200101100.

References

- [1] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020. [4321](#), [4326](#)
- [2] 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*, pages 8443–8452, 2021. [4326](#), [4327](#)
- [3] 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. *arXiv preprint arXiv:1911.09785*, 2019. [4322](#), [4325](#), [4326](#), [4327](#)
- [4] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5049–5059, 2019. [4322](#), [4325](#)
- [5] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998. [4321](#)
- [6] Paul S Bradley, Kristin P Bennett, and Ayhan Demiriz. Constrained k-means clustering. *Microsoft Research, Redmond*, 20(0):0, 2000. [4322](#), [4324](#)
- [7] Zhaowei Cai, Avinash Ravichandran, Subhansu Maji, Charles 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. [4326](#), [4327](#)
- [8] 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. [4326](#), [4327](#)
- [9] 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. [4322](#), [4325](#), [4326](#), [4327](#)
- [10] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009. [4321](#)
- [11] 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. [4322](#), [4325](#), [4326](#)
- [12] 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. [4322](#), [4327](#)
- [13] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. [4323](#), [4326](#)
- [14] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005. [4325](#)
- [15] 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. [4322](#), [4325](#)
- [16] 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. [4322](#), [4325](#), [4326](#)
- [17] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 41–50, 2019. [4321](#)
- [18] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5070–5079, 2019. [4326](#)
- [19] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009. [4326](#)
- [20] Chia-Wen Kuo, Chih-Yao Ma, Jia-Bin Huang, and Zsolt Kira. Featmatch: Feature-based augmentation for semi-supervised learning. *arXiv preprint arXiv:2007.08505*, 2020. [4322](#)
- [21] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, 2013. [4321](#)
- [22] 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. [4321](#), [4322](#), [4325](#), [4326](#), [4327](#)
- [23] Junnan Li, Caiming Xiong, and Steven CH Hoi. Learning from noisy data with robust representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9485–9494, 2021. [4322](#), [4327](#)
- [24] Yiting Li, Lu Liu, and Robby T Tan. Decoupled certainty-driven consistency loss for semi-supervised learning. *arXiv preprint arXiv:1901.05657*, 2019. [4322](#)
- [25] Thomas Lucas, Philippe Weinzaepfel, and Gregory Rogez. Barely-supervised learning: semi-supervised learning with very few labeled images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1881–1889, 2022. [4322](#)

- [26] Islam Nassar, Munawar Hayat, Ehsan Abbasnejad, Hamid Rezatofighi, Mehrtash Harandi, and Gholamreza Haffari. Lava: Label-efficient visual learning and adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 147–156, 2023. [4322](#)
- [27] 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. [4321](#), [4322](#)
- [28] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015. [4321](#)
- [29] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in neural information processing systems*, pages 3235–3246, 2018. [4326](#)
- [30] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019. [4326](#)
- [31] 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*, pages 11557–11568, 2021. [4327](#)
- [32] Qi Qian, Yuanhong Xu, Juhua Hu, Hao Li, and Rong Jin. Unsupervised visual representation learning by online constrained k-means. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16640–16649, 2022. [4324](#)
- [33] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016. [4326](#)
- [34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. [4326](#)
- [35] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in neural information processing systems*, pages 1163–1171, 2016. [4322](#)
- [36] David Sculley. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178, 2010. [4324](#)
- [37] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017. [4323](#)
- [38] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020. [4321](#), [4322](#), [4323](#), [4324](#), [4325](#), [4326](#), [4327](#)
- [39] Shuhan Tan, Xingchao Peng, and Kate Saenko. Class-imbalanced domain adaptation: An empirical odyssey. In *European Conference on Computer Vision*, pages 585–602. Springer, 2020. [4327](#)
- [40] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020. [4321](#), [4322](#)
- [41] Erik Wallin, Lennart Svensson, Fredrik Kahl, and Lars Hammarstrand. Doublematch: Improving semi-supervised learning with self-supervision. *arXiv preprint arXiv:2205.05575*, 2022. [4322](#)
- [42] Xudong Wang, Zhirong Wu, Long Lian, and Stella X Yu. Debaised learning from naturally imbalanced pseudo-labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14647–14657, 2022. [4321](#), [4322](#), [4326](#), [4327](#)
- [43] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019. [4321](#), [4322](#), [4326](#)
- [44] 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*, pages 11525–11536. PMLR, 2021. [4322](#)
- [45] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. [4326](#)
- [46] 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. [4322](#)
- [47] 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. [4322](#)
- [48] 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. [4322](#), [4325](#), [4326](#)