DATA-CENTRIC SEMI-SUPERVISED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

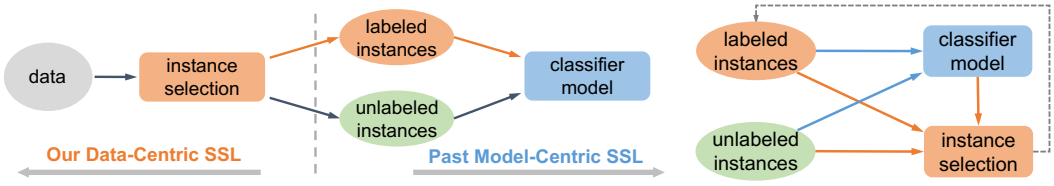
We study unsupervised data selection for semi-supervised learning (SSL), where a large-scale unlabeled data is available and a small subset of data is budgeted for label acquisition. Existing SSL methods focus on learning a model that effectively integrates information from *given* small labeled data and large unlabeled data, whereas we focus on selecting the right data for SSL without any label or task information, in an also stark contrast to supervised data selection for active learning. Intuitively, instances to be labeled shall collectively have maximum diversity and coverage for downstream tasks, and individually have maximum information propagation utility for SSL. We formalize these concepts in a three-step data-centric SSL method that improves FixMatch in stability and accuracy by 8% on CIFAR-10 (0.08% labeled) and 14% on ImageNet-1K (0.2% labeled). Our work demonstrates that a small compute spent on careful labeled data selection brings big annotation efficiency and model performance gain without changing the learning pipeline. Our completely unsupervised data selection can be easily extended to other weakly supervised learning settings.

1 INTRODUCTION

In many real-world applications, data are plenty but annotations are hard to get by. Semi-supervised learning (SSL), i.e., learning from a small set of labeled data and large-scale unlabeled data, thus becomes very relevant. We study the novel task of *unsupervised data selection* for SSL (Fig. 1).

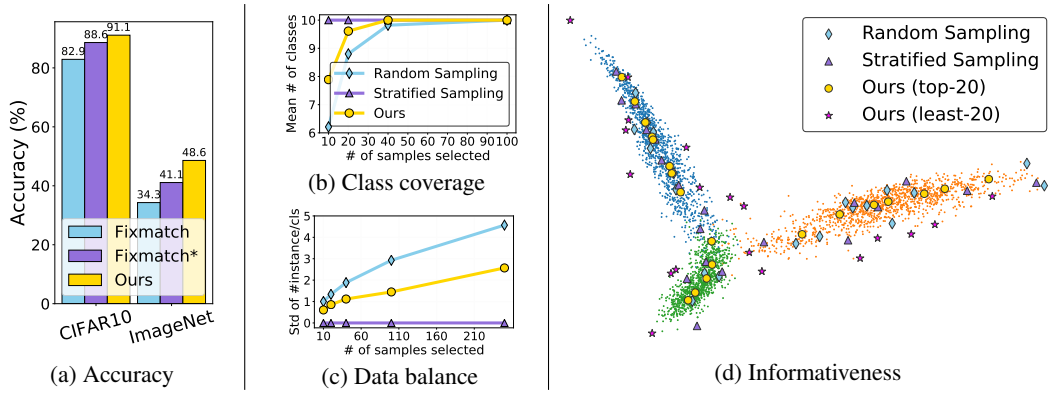
Existing SSL methods assume that labeled and unlabeled data are already given (Fig. 1**a**) and what's left is to optimize the classifier by integrating information from both small labeled data and large unlabeled data. The basic idea for SSL is that, while labeled data provides direct supervision on the classifier, unlabeled data helps regularize the classifier in terms of smoothness and sharpness, captured by consistent (Sajjadi et al., 2016; Tarvainen & Valpola, 2017; Xie et al., 2020; Lee et al., 2013; Berthelot et al., 2019b;a; Sohn et al., 2020) and minimal entropy (Grandvalet et al., 2005; Lee et al., 2013; Berthelot et al., 2019b) predictions respectively.

In practice, unlabeled data are ever expanding and labeled data have to be curated within a certain annotation budget. Selecting which instances to be annotated not only becomes a valid question, but can also play an important role in the model performance, fairness, robustness, safety, and scalability (Halevy et al., 2009; Carlini et al., 2019; Sohn et al., 2020; Mehrabi et al., 2021; Sambasivan et al.,



a) data-centric vs. model-centric SSL **b)** continuous active learning

Figure 1: We consider the novel task of unsupervised data selection for semi-supervised learning (SSL). **a)** Existing SSL methods focus on training the best model given labeled and unlabeled data, whereas we focus on optimizing labeled data selection. **b)** Existing AL methods learn a classifier based on random initial selection and alternate between training and instance selection, making its annotation pipeline both difficult to implement and inefficient under low shot settings.



(a) Accuracy (b) Class coverage (c) Data balance (d) Informativeness

Figure 2: We quantitatively and qualitatively compare our methods against previous works on four properties, i.e., (a) classification accuracies on downstream tasks, (b) class coverage, (c) semantic class distribution and (d) informativeness of queries, and drastically improve on all of them. FixMatch* and FixMatch in (a) denotes sampling with "supervised" stratified sampler, random sampler, respectively. Different dot colors in (d) denote different classes. 20 instances are sampled from these 3 classes with various sampling strategies. Our selections consistently cover informative samples across the space. (better viewed in color and zoomed in)

2021). We shift from past *model-centric* SSL to *data-centric* SSL, focusing on improving SSL performance by optimizing data selection from large-scale unlabeled data for annotation (Fig. 1**a**).

Two simple instance selection strategies are adopted for studying SSL on fully-labeled datasets: random sampling or stratified sampling by category (Berthelot et al., 2019a; Sohn et al., 2020). Each sampler has its own caveats: Random sampling can fail to cover all semantic categories and lead to poorer performance and model instability, whereas stratified sampling is in fact impossible in reality: If we can sample data by their categories, we would already have the label of every instance!

Data selection for optimizing model training is not new; it is in fact the focus of active learning (AL), where the task is defined by an initial set of labeled data and the goal is to select an additionally labeled subset (Fig. 1**b**) such that a model learned over the subset is competitive with that over the whole labeled dataset (Sener & Savarese, 2017; Ducoffe & Precioso, 2018; Yoo & Kweon, 2019).

However, our data-centric SSL addresses an *unsupervised data selection* problem: Given an SSL model, among many possible ways to select a small fixed number of instances in the sea of unlabeled data for labeling, which way would lead to the best SSL model performance, even though we have no labels and thus no idea what the downstream classification task would be? This task is also in a stark contrast to supervised data selection for AL, which is conditioned on an initial labeled set and for the benefit of the particular supervised classification task.

We develop a **D**ata-**C**entric SSL (**DC-SSL**) approach based on the intuition that ideal instances to be labeled shall collectively have maximum diversity and coverage for downstream classification tasks, and individually have maximum information propagation utility for SSL. It has three steps:
1. Unsupervised feature learning that maps data into a maximally discriminative feature space.
2. Select top-m representative instances for labeling per diversity, coverage, and information utility.
3. Apply an SSL method, e.g. FixMatch, on the selected labeled data and the rest unlabeled data.

Our result analysis shows several benefits over past model-centric SSL methods (Fig. 2):

- **On classification accuracy (Fig. 2a):** Just by selecting the right data to label, we improve the classification accuracy of FixMatch by an astonishing 8% and 14% in CIFAR-10 (with 0.08% labeled) and ImageNet-1k (with 0.2% labeled). These results are even better than from the practically infeasible stratified sampling which assumes perfectly balanced labeled instances.
- **On class coverage (Fig. 2b) and data balance (Fig. 2c):** Our method covers more classes of CIFAR-10 with a much smaller labeling budget and a more balanced distribution over classes: Ours cover 9.5 out of 10 classes on average when only 20 samples are selected for labeling. The standard deviation on the number of samples per class is also greatly reduced, resulting in a more uniform distribution over classes.
- **On informativeness (Fig. 2d):** Our method successfully discovers most representative and diversified samples from the unlabeled data of ImageNet: The 20 selected instances tend to be located

near density peaks; they are tightly connected to a large number of neighbors and spread out more uniformly in the feature space, allowing labeling information to propagate more effectively and widely. However, random and stratified sampling pick less informative outliers far too often, resulting in low learning efficiency and sometimes even learning collapse. These reasons underlie our better performance over stratified sampling, even though it is gauranteed to have balanced labeled data across classes. Note that our least-20 samples have the lowest utility values, lying in sparse areas with the weakest propagation abilities.

Our work shows that a small compute spent on careful labeled data selection brings big annotation efficiency and model performance gain without changing the learning pipeline. Our completely unsupervised data selection can be easily extended to other weakly supervised learning settings.

2 Related Work

Active Learning (AL) aims to select a small subset of labeled data to achieve competitive performance over supervised learning on fully labeled data (Cohn et al., 1994; Roy & Mccallum, 2001; Bilgic & Getoor, 2009). ***Conventional AL*** has three major camps (Settles, 2009; Ren et al., 2020): 1) Membership query synthesis (Angluin, 1988) requests to generate a membership query in the form of any unlabeled instance in the input space, including the membership queries generated from scratch; 2) Stream-based selective sampling (Dagan & Engelson, 1995; Atlas et al., 1990) selects one unlabeled instance at a time and uses informativeness to determine whether to query its label; 3) Pool-based active learning (Tong & Koller, 2001; Huang et al., 2010; Wei et al., 2015) generates queries in a greedy fashion by ranking unlabeled data. In ***Deep AL***, Core-Set (Sener & Savarese, 2017) approaches data selection as a set cover problem and with a derived upper bound it is equivalent to the k-Center problem. Adversarial-based approaches (Ducoffe & Precioso, 2018) use adversarial samples to estimate distance from decision boundaries based on sensitivity to adversarial attacks. Learning Loss for Active Learning (Yoo & Kweon, 2019) makes use of a novel parametric module to predict target losses of unlabeled data and queries the instance with largest loss for label. ***Semi-supervised Active Learning*** (SSAL) combines AL with SSL. Song et al. (2019) merges uncertainty-based metrics with MixMatch (Berthelot et al., 2019b). Gao et al. (2020) merges consistency-based metrics with SSL.

AL/SSAL often rely on initial labeled data to learn the model and sampler, requiring multiple (e.g. 10) rounds of sequential annotation and significant modifications of existing annotation pipelines. To our best knowledge, our work is the first *unsupervised* sampling method on large-scale recognition datasets that requests annotation only *once*, consistent with previous SSL pipelines. See more comparisons in Sec. 4.3.

Self-supervised Learning learns representations transferable to downstream tasks without annotations (Wu et al., 2018; Grill et al., 2020). ***Contrastive learning*** (Wu et al., 2018; He et al., 2020; Chen et al., 2020a; Wang et al., 2021) learns representations that map similar samples or different augmentations of the same instance close and dissimilar instances apart. ***Similarity-based*** methods (Grill et al., 2020) learn representations without negative pairs by predicting the embedding of a target network with an online network. ***Feature learning with grouping*** (Yang et al., 2010; Xie et al., 2016; Caron et al., 2018; Zhuang et al., 2019; Caron et al., 2020; Wang et al., 2021) respects the natural grouping of data by exploiting clusters in the latent representation. We study unlabeled data in such an unsupervisedly learned feature space, due to its high quality and low feature dimensions.

Semi-supervised Learning (SSL) integrates information from both small labeled data and large unlabeled data. ***Consistency based regularization*** (Sajjadi et al., 2016; Tarvainen & Valpola, 2017; Xie et al., 2020) applies a consistency term to the final loss by imposing invariance on unlabeled data under augmentations. ***Pseudo-labeling***, also known as self-training (Lee et al., 2013; Berthelot et al., 2019b;a; Sohn et al., 2020), relies on the model's high confidence predictions to produce pseudo-labels of unlabeled data and trains them jointly with labeled data. FixMatch (Sohn et al., 2020) integrates strong data augmentation (Cubuk et al., 2020), pseudo-labels filtering (Liu et al., 2019) and temperature re-scaling to achieve the current SOTA on SSL. It also explores training on the most representative samples ranked by (Carlini et al., 2019). However, (Carlini et al., 2019) is a supervised method that requires the use of all data labels. As a two-stage method, SimCLRv2 (Chen et al., 2020b) is a transfer learning method for SSL: It applies contrastive learning on unlabeled data, followed by supervised learning on labeled data. ***Entropy-minimization*** (Grandvalet et al., 2005; Lee et al., 2013; Berthelot et al., 2019b) assumes that classification boundaries do not pass through high-density area of marginal distributions, enforcing confident predictions on unlabeled data.

These SSL methods choose data to label by random sampling which leads to poor performance, or by stratified sampling that selects the same number of samples per category, assuming unrealistically that labels are available on all the data. Instead of focusing on novel models and algorithms, our method is *data-centric*, focusing on choosing the right samples for most effective SSL.

3 DATA-CENTRIC SEMI-SUPERVISED LEARNING

Existing SSL methods are *model-centric*, focusing on novel models and algorithms. We instead explore *Data-Centric* Semi-Supervised Learning (DC-SSL), focusing on selecting the most informative data to label for effective SSL. DC-SSL has three steps: 1) Unsupervised feature learning; 2) Unsupervised sample selection for annotation; 3) Semi-supervised learning on selected labeled data and rest unlabeled data. We also demonstrate how a large-scale model trained on general network-crawled text-image pairs accelerates our algorithm on novel datasets while improving the quality.

3.1 PROBLEM SETUP

Consider a dataset $\mathbb{D} = (\boldsymbol{x}_i, y_i)_{i=1}^n$ with n images, and the labels y_i are not known unless we request annotation. Since there is no preview of y_i for *any* of the $\boldsymbol{x}_i$, stratified sampling that results in balanced labeled data per class is impossible, as commonly assumed in (Chen et al., 2020b; Berthelot et al., 2019a; Sohn et al., 2020). Given a limited labeling budget, our goal is to identify the top-m representative samples $\mathbb{A} = (\boldsymbol{x}_j, y_j)_{j=1}^m$ from $\mathbb{D}$ to annotate, so as to improve SSL performance jointly trained on unlabeled data $\mathbb{D}$ and labeled data $\mathbb{A}$. Several popular SSL methods are utilized to verify the effectiveness of our data selection method while keeping the algorithms unchanged.

3.2 UNSUPERVISED REPRESENTATION LEARNING

We apply unsupervised feature learning, contrastive learning in particular, to obtain a lower dimensional (128-D) representation that maximally maintains the individuality of each instance (Wu et al., 2018; Oord et al., 2018; He et al., 2020; Chen et al., 2020a). Its quality is shown close to or even better than the supervisedly learned representations on many downstream tasks. We learn a mapping function f such that in the $f(x)$ feature space, instance x_i is attracted to its augmented version x_i' and repulsive to a different instance x_j, $j \neq i$. We model f by a convolutional neural network, mapping x onto a d-dimensional hypersphere with L^2 normalization. To make a fair comparison with previous arts (Sohn et al., 2020; Cai et al., 2021), we use MoCo v2 to learn representations on ImageNet with instance-centric contrastive loss. The feature spaces of CIFAR-10 images are extracted with CLD (Wang et al., 2021), which adds a instance-group contrastive loss, due to its improved feature quality. See formulations and interpretations of MoCo v2 and CLD in Appendix A.2.

3.3 UNSUPERVISED SAMPLE SELECTION FOR ANNOTATION

We represent $(\boldsymbol{x}_i)_{i=1}^n$ as a weighted graph $G = (V, E; W)$, where nodes are points in the feature space, and edges between nodes are attached with weights of pairwise feature similarity (Bondy et al., 1976; Deo, 1975; Chung & Graham, 1997; Shi & Malik, 2000): $w_{ij} = 1/\|f(x_i) - f(x_j)\|$, where $f(x_i)$ is the L^2 normalized feature of V_i, learned from e.g. MoCo (He et al., 2020). While w_{ij} is equivalent to cosine similarity $w_{ij} = \cos(V_i, V_j)$ in theory, we find that $1/\|f(x_i) - f(x_j)\|$ works better empirically than $\cos(V_i, V_j)$. Please see an explanation in Appendix A.3.

Given a labeling budget of m instances, we aim to select m instances that are diverse for coverage and informative for propagating label information. We evaluate the utility of each instance in terms of information transport efficiency on a graph.

Density estimation with K-NN. The most straightforward idea of sample selection for labeling is to select the well-connected node, which is most likely to spread semantic information to nearby nodes with the smallest cost. It corresponds to a density peak in the feature space. We evaluate the density by K-Nearest Neighbor (K-NN) density estimation, whose key idea is to base estimation on a closest K observations (Fix & Hodges, 1989; Loftsgaarden & Quesenberry, 1965). We also conduct ablation studies on other density estimation methods and find our customized K-NN best performing (Fig. 7).

The basic K-nearest neighbour density (K-NN) estimate is constructed as follows (Fix & Hodges, 1989; Loftsgaarden & Quesenberry, 1965; Orava, 2011):

$$\hat{f}_{\text{KNN}}(V_i, K) = \frac{1}{n} \sum_{j=1}^{K} \frac{1}{A_d \cdot D(V_i, \hat{V}_j)}, \qquad \text{with } A_d = \frac{\pi^{d/2}}{\Gamma(\frac{d}{2} + 1)} \tag{1}$$

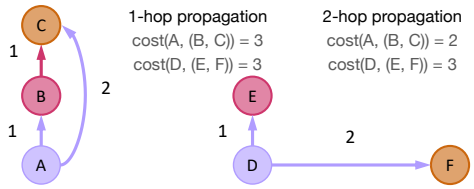


Figure 3: Two-hop propagation considers the indirect connection between two vertices, and more accurately estimate the information dissemination ability of each vertex. The overall distance, i.e., cost$(\cdot)$, of propagating semantic information from vertex A to vertices B, C and vertex D to vertices E, F are illustrated, where the purple and magenta lines denote one-hop and two-hop propagation, respectively. Numbers shown here are distances between two vertices.

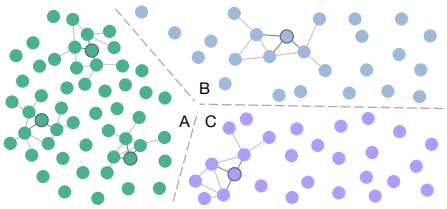


Figure 4: A case where local density alone gives a bad querying results. If we only sample three instances to be annotated from the region with the highest estimated local density, only the three samples in cluster A will be queried due to their highest density. However, this may impair the ability of queries to capture multiple data distribution patterns and is less efficient in propagating semantic information to all unlabeled data.

where A_d is the volume of a unit d-dimensional ball, $\Gamma(x)$ the Gamma function, $R_k^d(V_i)$ the distance between V_i and its Kth nearest neighbor, $D(V_i, \hat{V}_j)$ the radius $R_k^d(V_i)$ of the ball centered at V_i:

$$B(V_i, R_k^d(V_i)) = \{\hat{V}_j : \|V_i - \hat{V}_j\| \leq R_k^d(V_i)\}, j \in \{1, ..., K\}. \tag{2}$$

Drawbacks of K-NN density estimation on self-supervised models. **1)** Only the Kth nearest neighbor is referenced, while the relationships with all other $K-1$ points in the ball are not taken into consideration, rendering it sensitive to outliers. **2)** Densest peaks capture representativeness in local small regions but ignores diversity in the global feature space (Fig. 4). Nearby density peaks also introduce redundancy along with deficient coverage. **3)** Evaluating the utility of a node based on its direct neighbours ignores indirect connections between nodes (Fig. 3). It is likely to miss vertices (false negatives) without direct connections with anchored vertices, even if such vertices are well connected to anchored vertices through other intermediate vertices. **4)** Our unsupervisedly learned feature does not model semantics explicitly. We propose an extended KNN estimator that can mitigate the aforementioned issues.

Averaged distance function for robust density estimation. The basic $K-$NN estimator uses the distance to the kth nearest neighbor alone ($D(V_i, V_j) = R_k^d(V_i)$) for density estimation, which makes it highly sensitive to outliers. We consider all k neighbors instead:

$$\hat{f}_{\text{KNN}}(V_i, k) = \frac{k}{n} \frac{1}{A_d \cdot \frac{1}{k} \sum_{j=1}^{k} \frac{1}{w_{ij}}}. \tag{3}$$

We compare it to alternative formulations and show its advantage in Sec. 4.4.

Multi-hop propagation for measuring indirect connections between vertices. To model indirect connections, we formulate the utility of vertex V_i with two-hop propagation:

$$U(V_i) = \hat{f}_{\text{KNN}}(V_i, K) + \sum_{\hat{V}_j \in B(V_i, R_k^d(V_i))} \cos(V_i, \hat{V}_j) \cdot \hat{f}_{\text{KNN}}(\hat{V}_j, K) \tag{4}$$

Compared with a single-hop propagation for direct connections, the multi-hop propagation estimates the information dissemination ability, i.e., informativeness of each vertex, more accurately. In Fig. 3, the one-hop cost of information propagation from A to vertices (B, C) and D to (E, F) is the same for the left and right graphs, although C can be learned more effectively through B in the left graph instead of directly from A. The two-hop propagation can be naturally extended to multi-hop propagation by looking at higher order neighbors. In consideration of simplicity and efficiency, we only explore propagation of up to two-hop.

K-Means clustering for diversity and coverage. The instances to be selected for labeling shall not only be representative of its local neighbourhood, but also collectively capture the structure of the entire unlabeled data. We perform K-Means clustering that partitions the n instances into m ($\leq n$) clusters $\mathbb{S} = \{S_1, S_2, ..., S_m\}$ in which each instance belongs to the cluster with the nearest cluster centroid μ, serving as a prototype of the cluster (Lloyd, 1982; Forgy, 1965). m is approximately equal to our annotation budget. The objective is to find $\mathbb{S}$ that minimizes the within-cluster sum of squares (WCSS) (Kriegel et al., 2017):

$$\arg\min_{\mathbb{S}} = \sum_{i=1}^{m} \sum_{V \in S_i} \|V - \mu_i\|^2 = \arg\min_{\mathbb{S}} \sum_{i=1}^{m} |S_i| \text{Var}(S_i) \tag{5}$$

We then query one instance from each cluster according to their utility score evaluated according to Eqn. 4. The m centroids of K-Means clustering are initialized randomly and optimized with the EM algorithm (McLachlan & Krishnan, 2007) until convergence.

Regularization with inter-cluster information passing channels. We observe that our instance sampler still selects samples very close to other selected samples in adjacent regions (Fig. 5), especially when the densest area in a cluster is at a corner or boundary.

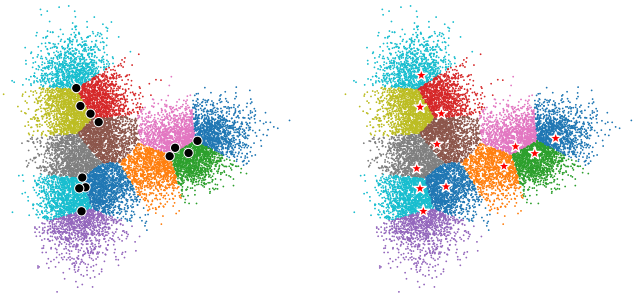


Figure 5: Sampling with regularization (right) can produce more uniform and diversified queries. On the contrary, the queries acquired without regularization (left) are usually located in a small dense region and have high correlations, losing the overall representativeness. (better viewed in color)

To avoid such repetitive sampling that reduces the total information carried in the labeled set, we apply an iterated regularization algorithm. At each iteration, different clusters exchange information about other clusters' selected samples and gradually adjust their selections for better diversity and coverage. Specifically, after the original sample selection, we compute a regularizer $\text{Reg}(V_i, t)$ for each sample V_i based on the distance from each candidate to all members of the selected query set $\hat{\mathbb{V}}^{t-1} = \{\hat{V}_1^{t-1}, ..., \hat{V}_m^{t-1}\}$ at iteration $t-1$, except for the query in the same cluster S_i, with a tunable strength hyperparameter α:

$$\text{Reg}(V_i, t) = \sum_{\hat{V}_j^{t-1} \notin S_i} \frac{1}{\|V_i - \hat{V}_j^{t-1}\|^\alpha}. \tag{6}$$

The regularizer is updated with an exponential moving average with momentum m_{reg}:

$$\bar{\text{Reg}}(V_i, t) = m_{\text{reg}} \cdot \bar{\text{Reg}}(V_i, t-1) + (1 - m_{\text{reg}}) \cdot \text{Reg}(V_i, t). \tag{7}$$

The new samples are chosen by subtracting the regularizer for that sample multiplied by a hyperparameter λ. λ controls the trade-off between diversity and informativeness: A low λ leads to samples close to each other and of low diversity; a high λ leads to uniformly distributed samples of low informativeness. After that, at iteration t, we select sample i with maximum **regularized utility** $U'(V_i, t)$ in each cluster, formulated as:

$$U'(V_i, t) = U(V_i) - \lambda \cdot \bar{\text{Reg}}(V_i, t). \tag{8}$$

The selection formed at the last iteration is our final choice. By adding a soft regularization, this formulation does not just push a data point selection away to a certain distance, but it considers tradeoff between diversity and representativeness and asks each selection to find another mode in the cluster which potentially brings new information if labeled, greatly improving the overall informativeness. We refer readers to Alg. 1 in appendix for pseudo-code of the regularization algorithm.

For measuring the regularizer on large datasets, calculating the distances between a sample and all queries in $\hat{\mathbb{V}}^{t-1}$ is no longer feasible. Fortunately, only selected samples that are close to the current candidate make a big difference in $\text{Reg}(V_i, t)$. Therefore, we set a horizon h, defined as the number of nearest neighbor selections from $\hat{\mathbb{V}}^{t-1}$ to be considered, for large datasets. We found that h ranging from 32 to 128 generally work well on 128-dim feature on MoCo or CLD and 512-dim feature on CLIP and changing h does not make a difference.

Acceleration with general-domain multi-modal models. We further improve our method on two aspects: 1) We need to train a self-supervised model for each new dataset, which is time-consuming and could potentially delay data annotation in real-world applications. 2) Self-supervised methods do not model semantic information explicitly, and datasets with varying intra-class variance assume regions of various sizes and may be treated unexpectedly. To address these issues, we resort to a large pretrained model that encodes semantics without manual human annotation.

We make use of publicly-available CLIP (Radford et al., 2021) models, a collection of models trained on Internet-crawled large-scale data with a wide general domain, and use CLIP's image model as feature extractor. We observe that such substitution does not hurt and even helps the performance when annotation budget is large when compared to self-supervised models, demonstrating the effectiveness of making use of semantic information. Since we do not need to train models with target dataset, the whole sample selection process could complete in *0.5 hours* on a commodity server with one GPU, which illustrates the possibility of our methods without delaying the schedule of human annotation or modifying the annotation pipeline to wait for self-supervised models and enables it for industry on real-world dataset collection. More discussions on CLIP can be found in Appendix A.4.

4 EXPERIMENTS

In this section, we perform a comprehensive evaluation on the proposed Data-Centric Semi-Supervised Learning (DC-SSL) integrated with two mainstream SSL approaches: pseudo-label-based FixMatch (Sohn et al., 2020) and approach based on transfer learning (Chen et al., 2020b).

4.1 CIFAR-10: EXTREMELY LOW-SHOT SETTINGS

Sample Selection	Accuracy (%) of Last Epoch (Best Epoch)		
	40 labels	100 labels	250 labels
Random	82.94±9.77 (86.67±4.71)	88.72±4.61 (91.32±2.05)	93.28±0.04 (93.69±0.05)
DC-SSL (Ours)	**91.10**±0.95 (**91.27**±0.95)	**93.22**±0.31 (**93.55**±0.33)	**93.63**±0.14 (**93.92**±0.15)
Δ (DC-SSL - Random)	**↑ 8.16 (↑ 4.60)**	**↑ 4.50 (↑ 2.23)**	**↑ 0.35 (↑ 0.23)**
Stratified †*	88.61±3.35	90.19±2.24	94.93 ± 0.33

Table 1: CIFAR-10 experiments with pseudo-label-based SSL method FixMatch (Sohn et al., 2020), with 3 different folds. †: not a fair comparison with us because it assumes balanced labeled data available and leaks information about ground truth labels. * indicates results reported in FixMatch. Note that the 0.35% improvement in 250 label task is *not marginal*, since 250 labels to 4000 labels (16×) only leads to around 0.7% improvement as reported in FixMatch.

Setup. We evaluate both approaches on low-shot settings from 40 samples to 250 samples in total (4 to 25 shots per class on average). We also re-evaluate FixMatch with random sampling as a fair comparison. Since our work is data-centric, we follow the same training recipe unless stated. Detailed setup is in Appendix A.7.

Main Results. ***Pseudo-label-based SSL.*** Three charming properties of DC-SSL are observed in the CIFAR-10 experiments: *efficiency, stability and counter-collapsing*. In 40/100 labels task, collapsing happens since outliers could mislead the network into classifying a range of samples to a wrong class, which further generates wrong pseudo-labels and misleads the classifier. Our method reduces this effect by selecting labels that are both representative and diverse, shown by reducing the difference between best and last epoch from 3.73% to negligible in 40 samples task. In 250 samples task, although FixMatch already performs well, we are able to push the limit further by a small amount. In 40 samples task, two of first 5 seeds of random selection has labels from fewer than 10 classes and to better understand the effect of our methods, we skip these two seeds in random selection, whereas it does not occur in our method, even when we sample 100 times with 100 seeds. ***Transfer-learning-based SSL.*** Since transfer learning only considers labeled samples in fine-tuning stage, the quality of the trained classifier greatly depends on selected samples' quality when compared to FixMatch. Table. 2 shows that our improvement is prominent especially when the number of selected samples is low. In 40 (250) labels case, we are able to achieve a 15.5% (2.7%) improvement.

4.2 IMAGENET

Setup. To further evaluate the effectiveness of our method on large-scale datasets with more classes. We perform evaluation of our method on ImageNet (Russakovsky et al., 2015) with about 1.28M images and 1k classes. We perform supervised learning (SL) and SSL on both 1% labeled data (12820 samples) and 0.2% (2911 samples). More detailed setup on ImageNet are in Appendix A.8.

Main Results. As summarized in Table 3, samples selected from both MoCo and CLIP models boost the performance of SL and SSL. In 1% case, DC-SSL provides 1.2% to 3.8% (2.8% to 3.4%) gains in SL (SSL) setting. What is more interesting is the 0.2% setting, where our method leads

Sampling Methods	40 Labels	100 Labels	150 Labels	250 Labels
Random	60.8 ± 3.2	73.7 ± 2.5	76.2 ± 2.2	79.4 ± 1.7
DC-SSL (Ours)	**76.3** ± 1.6 ↑ **15.5**	**79.0** ± 0.3 ↑ **5.3**	**80.8** ± 0.5 ↑ **4.6**	**82.1** ± 0.3 ↑ **2.7**
Stratified †	66.5 ± 1.6	74.5 ± 0.8	78.3 ± 1.1	80.4 ± 0.8

Table 2: CIFAR-10 experiments with transfer-learning-based SSL method, as proposed in SimCLRv2 (Chen et al., 2020b), with the mean and std of 5 different folds and 2 runs in each fold. †: assumes prior label information and thus not a fair comparison.

	Supervised Learning		Semi-supervised Learning	
Sampling Methods	1%	0.20%	1%	0.20%
Random	23.5	6.0	58.8	34.3
DC-SSL with MoCo (Ours)	24.7 ↑ 1.2	9.2 ↑ 3.2	61.6 ↑ 2.8	**48.6** ↑ **14.3**
DC-SSL with CLIP (Ours)	**27.3** ↑ **3.8**	**9.7** ↑ **3.7**	**62.2** ↑ **3.4**	47.5 ↑ 13.2
Stratified †	22.9	6.3	60.9*	41.1

Table 3: ImageNet experiments with fully supervised training and SSL algorithm FixMatch (Sohn et al., 2020) and Exponential Moving Average Normalization (EMAN) (Cai et al., 2021). * indicates results reported in EMAN. Note that † is not a fair comparison, as explained earlier.

to an improvement ranging from 13.2% to 14.3%. We found that samples selected from MoCo performs 1.1% better than samples from CLIP in the FixMatch setting. This is, in part, due to mismatch between parameter initialization (MoCo) and the feature space used for the sampling process (CLIP). However, for 1% case, we find that this dependency disappears and samples with CLIP perform 0.6% better than MoCo counterparts, which demonstrates the benefits of using a model with explicit semantic information during pretraining with sufficient general knowledge.

Universality in Results with CLIP: Since CLIP *does not use ImageNet samples* in training and the downstream SSL task *has never seen the CLIP model* in training either, we stress the importance of *universality*: it means that 1) when a new dataset is collected, we could use a general model with knowledge of the target domain to select samples without waiting for self-supervised learning (see Sec. 4.4); 2) different from active learning, where sample selection process is strictly coupled with training, our annotated dataset works universally rather than only with the model used to select it.

4.3 Comparison with Active Learning and SSAL Methods

Following previous deep AL/SSAL works, we mainly use CIFAR-10 for comparison. Our method is under disadvantage in comparison because we request annotation only once, but we still outperform previous AL/SSAL algorithms with higher label efficiency. We first compare with AL. Since AL only utilizes labeled samples, even recent AL algorithm Cho et al. (2021) requires at least 10,000 samples to achieve 90% accuracy (250× more than ours). For fair comparison, we attempt to allocate 20 samples for initial learning, use AL to select another 20 samples, and utilize the 40 samples in SSL. After implementation of Yoo & Kweon (2019), we found that since 20 is smaller than batch size 128, AL model has about 20% accuracy given 20 initial labeled samples and thus could not serve as a useful comparison. We then compare with SSAL. Similar to our work, SSAL (Gao et al., 2020; Song et al., 2019) uses labeled samples with SSL methods. However, it still requires initial annotation from random sampling. Since random sampling is not efficient on low-shot selection (Fig. 2 (b)), to cover all classes in initial random sampling, such methods typically require much more samples than DC-SSL as in Table 4. In contrast, our method could select samples efficiently even in low-shot. We believe that it is possible to further improve our work if we use DC-SSL for initial selection in SSAL and we leave it to future works due to the complexity of both algorithms.

4.4 Ablation Study and Analysis

Component Analysis. To systematically investigate the contributions of each component we propose, we evaluate our method with an extensive ablation study. We adhere to previous settings unless stated. Illustrated in Fig. 6, applying the vanilla version of our method based on first order scores achieves large improvements on both SSL methods, especially transfer learning approach, where the accuracy rises by 11.9%. Adding regularization gives another 4.6% and 3.3% by contributing to sample diversity. In the end, 2-hop propagation leads to a marginal of 1.2% and 0.3% improvement on both methods. These contributions together lead to our final 8.2% and 15.5% improvements.

Sample Selection	Accuracy (%)	# Initial Labels*	Label Budget
Core-Set (Sener & Savarese, 2017) †	48.33 ± 0.49	100	150
Core-Set †	50.96 ± 0.45	100	200
CBSSAL (Gao et al., 2020)	87.57 ± 0.31	100	150
DC-SSL (Ours)	91.10 ± 0.95 (↑ 3.53)	0	40
DC-SSL (Ours)	**93.22** ± 0.31 (↑ **5.65**)	0	100
Core-Set †	53.77 ± 0.49	100	250
Core-Set + SSL †	88.75 ± 0.42	100	250
CBSSAL	90.23 ± 0.39	100	250
MMA (Song et al., 2019)	91.69 ± 0.52	250	500
MMA + k-Means	91.46 ± 0.38	250	500
DC-SSL (Ours)	**93.63** ± 0.14 (↑ **2.17**)	0	250

Table 4: Comparison with active learning methods. Note that although MMA also experimented with k-Means, it does not benefit their performance. †: from Gao et al. (2020). *: We give all queries in one round and do not need initial labels. CBSSAL considers interacting with FixMatch for better performance, however DC-SSL can still yield better performance without any initial labels.

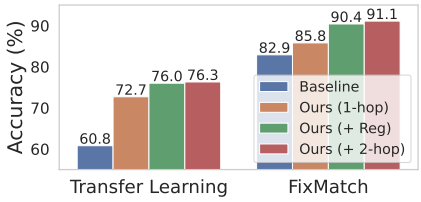


Figure 6: Illustration of our methods with different ablations on CIFAR-10 with 40 samples. In both SSL methods, applying 1-hop propagation gives a large improvements (11.9% and 2.9%, respectively), and regularization as well as 2-hop propagation together give an additional improvement of 3.6% on Transfer Learning and 5.3% on FixMatch.

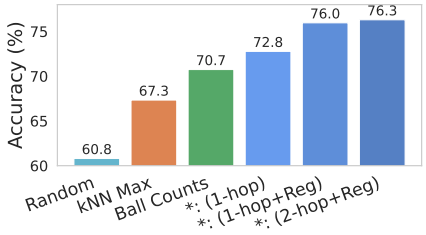


Figure 7: Illustration of the behavior of different utility estimations. *KNN Max* refers to vanilla K-NN density estimation that takes the max distance of K nearest-neighbors. *Ball Counts* counts neighbors within a radius. Our methods (with *): *1-hop* is our method without regularization or second order term. *1-hop+Reg* adds regularization and *2-hop+Reg* adds a second order propagation.

Formulation Analysis. We also systematically evaluated variants of our formulations used in representativeness estimation, as presented in Fig. 7, by comparing different ablations of our methods to other metrics. We compare against two common metrics: *KNN Max* (vanilla K-NN) and *Ball Counts*, which uses neighbor counts within a radius. Although all three methods are much better than random selection, our metric leads to the best outcome. More ablation studies on hyperparameters, such as weight of regularizer and number of neighbors, can be found in Appendix A.5.

Run Time Analysis. We only add a negligible delay to label selection compared to SSL methods. On ImageNet, we introduce only less than 1 GPU hour overhead when compared to about 2300 GPU hours with the original FixMatch pipeline, which is in turn, much less than the time for labelling 1.28 million images with 1k categories. With CLIP, the data selection stage could start as soon as data is available, which enables informative labeling with very low delay. Compared to active learning which requires up to 10 rounds of sequential human annotations, we only request annotation *once*, consistent to SSL pipelines. More discussions are in Appendix A.6.

5 Summary

Existing SSL methods are model-centric, focusing on models and algorithms that integrate both labeled and unlabeled data. Instead, our DC-SSL is the first work that turns to labeled data selection in SSL in an unsupervised way. By simply optimizing what is fed into a model, we demonstrate significant gains on annotation efficiency, model stability and accuracy on all experimented benchmarks. Our work is also in a stark contrast to supervised data selection for active learning. Our completely unsupervised data selection can be easily extended to other weakly supervised learning settings.

REPRODUCIBILITY STATEMENT

In order to make our results reproducible, we listed all used parameters and changes we applied for training in the Section A.7 and Section A.8 of Appendix and Section 4.1, 4.2 and 4.3 of Experiment, and cited the corresponding papers on which our method is build on which provide publicly available Github repositories themselves. We are committed to reproducible results reported in the paper and public code release.

REFERENCES

Dana Angluin. Queries and concept learning. *Machine learning*, 2(4):319–342, 1988.

Les E Atlas, David A Cohn, and Richard E Ladner. Training connectionist networks with queries and selective sampling. In *Advances in neural information processing systems*, pp. 566–573. Citeseer, 1990.

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*, 2019a.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019b.

Mustafa Bilgic and Lise Getoor. Link-based active learning. In *NIPS Workshop on Analyzing Networks and Learning with Graphs*, volume 4, 2009.

John Adrian Bondy, Uppaluri Siva Ramachandra Murty, et al. *Graph theory with applications*, volume 290. Macmillan London, 1976.

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*, pp. 194–203, 2021.

Nicholas Carlini, Ulfar Erlingsson, and Nicolas Papernot. Distribution density, tails, and outliers in machine learning: Metrics and applications. *arXiv preprint arXiv:1910.13427*, 2019.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

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, 2020.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020a.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020b.

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

Jae Won Cho, Dong-Jin Kim, Yunjae Jung, and In So Kweon. Mcdal: Maximum classifier discrepancy for active learning. *arXiv preprint arXiv:2107.11049*, 2021.

Fan RK Chung and Fan Chung Graham. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine learning*, 15(2):201–221, 1994.

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*, pp. 702–703, 2020.

Ido Dagan and Sean P Engelson. Committee-based sampling for training probabilistic classifiers. In *Machine Learning Proceedings 1995*, pp. 150–157. Elsevier, 1995.

N Deo. Graph theory with applications to engineering and computer science. *Networks*, 5(3):299–300, 1975.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Melanie Ducoffe and Frederic Precioso. Adversarial active learning for deep networks: a margin based approach. *arXiv preprint arXiv:1802.09841*, 2018.

Evelyn Fix and Joseph Lawson Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique*, 57 (3):238–247, 1989.

Edward W Forgy. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *biometrics*, 21:768–769, 1965.

Mingfei Gao, Zizhao Zhang, Guo Yu, Sercan Ö Arık, Larry S Davis, and Tomas Pfister. Consistency-based semi-supervised active learning: Towards minimizing labeling cost. In *European Conference on Computer Vision*, pp. 510–526. Springer, 2020.

Yves Grandvalet, Yoshua Bengio, et al. Semi-supervised learning by entropy minimization. *CAP*, 367:281–296, 2005.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.

Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, 2009.

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*, pp. 9729–9738, 2020.

Sheng-Jun Huang, Rong Jin, and Zhi-Hua Zhou. Active learning by querying informative and representative examples. *Advances in neural information processing systems*, 23:892–900, 2010.

Hans-Peter Kriegel, Erich Schubert, and Arthur Zimek. The (black) art of runtime evaluation: Are we comparing algorithms or implementations? *Knowledge and Information Systems*, 52(2):341–378, 2017.

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

Bin Liu, Zhirong Wu, Han Hu, and Stephen Lin. Deep metric transfer for label propagation with limited annotated data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2): 129–137, 1982.

Don O Loftsgaarden and Charles P Quesenberry. A nonparametric estimate of a multivariate density function. *The Annals of Mathematical Statistics*, 36(3):1049–1051, 1965.

Geoffrey J McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Jan Orava. K-nearest neighbour kernel density estimation, the choice of optimal k. *Tatra Mountains Mathematical Publications*, 50(1):39–50, 2011.

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. *arXiv preprint arXiv:2103.00020*, 2021.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *arXiv preprint arXiv:2009.00236*, 2020.

Nicholas Roy and Andrew Mccallum. Toward optimal active learning through sampling estimation of error reduction. *Proceedings of the 18th International Conference on Machine Learning*, 08 2001.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *arXiv preprint arXiv:1606.04586*, 2016.

Nithya Sambasivan, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, and Lora M Aroyo. "everyone wants to do the model work, not the data work": Data cascades in high-stakes ai. In *proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–15, 2021.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

Burr Settles. Active learning literature survey. 2009.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

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, 2020.

Shuang Song, David Berthelot, and Afshin Rostamizadeh. Combining mixmatch and active learning for better accuracy with fewer labels. *arXiv preprint arXiv:1912.00594*, 2019.

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

Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

Xudong Wang, Ziwei Liu, and Stella X Yu. Unsupervised feature learning by cross-level instance-group discrimination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12586–12595, 2021.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning*, pp. 1954–1963. PMLR, 2015.

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*, pp. 3733–3742, 2018.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, 2016.

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.

Yi Yang, Dong Xu, Feiping Nie, Shuicheng Yan, and Yueting Zhuang. Image clustering using local discriminant models and global integration. *TIP*, 2010.

Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 93–102, 2019.

Chengxu Zhuang, Alex Lin Zhai, Daniel Yamins, , et al. Local aggregation for unsupervised learning of visual embeddings. In *ICCV*, 2019.

A APPENDIX

Algorithm 1 The iterative regularization algorithm

Require:
$\{U(V_i)|V_i \in \mathbb{V}\}$: The unregularized utility for each vertex V_i
λ: weight for applying regularization
m_{reg}: momentum in exponential moving average
l: the number of iterations
Procedure:
$\bar{\text{Reg}}(V_i, 0) \leftarrow 0, \forall V_i \in \mathbb{V}$
$\hat{\mathbb{V}}^0 \leftarrow$ samples with largest $U(V_i)$ in each cluster
for $t = 1$ **to** l **do**
  for all $V_i \in \mathbb{V}$ **do**
    $\text{Reg}(V_i, t) \leftarrow \sum_{\hat{V}_j^{t-1} \notin S_i} \frac{1}{\|V_i - \hat{V}_j^{t-1}\|^\alpha}$
    $\bar{\text{Reg}}(V_i, t) \leftarrow m_{\text{reg}} \cdot \bar{\text{Reg}}(V_i, t-1) + (1 - m_{\text{reg}}) \cdot \text{Reg}(V_i, t)$
    $U'(V_i, t) \leftarrow U(V_i) - \lambda \cdot \bar{\text{Reg}}(V_i, t)$
  end for
  $\hat{\mathbb{V}}^t \leftarrow$ samples with largest $U'(V_i, t)$ in each cluster
end for
return $\hat{\mathbb{V}}^l$

A.1 DETAILED DESCRIPTION FOR REGULARIZATION ALGORITHM

We summarized the regularization algorithm in pseudo-code in Alg. 1. In Alg. 1, we first obtain $\hat{\mathbb{V}}^0$, the selection without regularization, and set the moving average regularizer $\hat{\text{Reg}}(V_i, 0)$ to 0 for every $V_i \in \mathbb{V}$; then in each iteration, we update $\hat{\text{Reg}}(V_i, t)$ with moving average from a closeness measurement to other previously selected samples, where t is the index of current iteration. We re-select samples according to regularized utility at the end of each iteration, with λ being a balancing factor. In the end, the selection from the last iteration is returned.

A.2 OVERVIEW ON UNSUPERVISED REPRESENTATION LEARNING

In self-supervised learning stage, we aim to learn a mapping function f such that in the $f(x)$ feature space, the positive instance x_i' is attracted to instance x_i, meanwhile, the negative instance x_j (with $j \neq i$) is repelled, and we model f by a convolutional neural network, mapping x onto a d-dimensional hypersphere with L^2 normalization. To make a fair comparison with previous arts (Cai et al., 2021), we use MoCo v2 (Chen et al., 2020c) to learn representations on ImageNet with the instance-centric contrastive loss:

$$C\left(f_i, f_i^+, f_{\neq i}^-\right) = -\log \frac{\exp(< f_i, f_i^+ > /T)}{\exp(< f_i, f_i^+ > /T) + \sum_{j \neq i} \exp(< f_i, f_j^- > /T)} \tag{9}$$

where T is a regulating temperature. Minimizing it can be viewed as maximizing the mutual information (MI) lower bound between the features of the same instance (Hadsell et al., 2006; Oord et al., 2018). For experiments on ImageNet, the MoCo model pre-trained for 800 epochs is used for initializing SSL model, as in (Cai et al., 2021).

The feature spaces of CIFAR-10 data we work on are extracted with CLD (Wang et al., 2021). The instance-group contrastive loss is added in symmetrical terms over views x_i and x_i':

$$\begin{aligned} L(f; T_I, T_G, \lambda) = & \sum_i C(f_I(x_i), v_i, v_{\neq i}; T_I) + C(f_I(x_i'), v_i, v_{\neq i}; T_I) \\ & + \lambda \sum_i C(f_G(x_i'), M_{\Gamma(i)}, M_{\neq \Gamma(i)}; T_G) + C(f_G(x_i), M'_{\Gamma'(i)}, M'_{\Gamma'(i)}; T_G) \end{aligned} \tag{10}$$

Cross-level discrimination of Eqn. 10 (second term) can be understood as minimizing the cross entropy between hard clustering assignment based on $f_G(x_i)$ and soft assignment predicted from $f_G(x_i')$ in a different view, where f_G (f_I) is instance (group) branch, and $M_{\Gamma(i)}$ denotes the cluster centroid of instance x_i with a cluster id $\Gamma(i)$ (Wang et al., 2021). Empirically, we found that CLD has

great feature quality on CIFAR-10 and better respects the underlying semantic structure of data. To be consistent with original FixMatch settings, our semi-supervised learner on CIFAR-10 is trained from scratch, without using pretrained weights.

A.3 USING EUCLIDEAN DISTANCE OR COSINE SIMILARITY?

Because the features of all instances are projected to a unit hypersphere with L2 normalization, theoretically, maximizing the cosine similarity between two nodes is equivalent to maximizing the inverse of Euclidean distance between two nodes: $\arg\max_{i,j}(\|f(x_i) - f(x_j)\|_2)^{-1} = \arg\max_{i,j}(2 - 2\cos(f(x_i), f(x_j)))^{-1} = \arg\max_{i,j}(\cos(f(x_i), f(x_j)))$. However, empirically, using maximizing the inverse of Euclidean distance $1/d(\cdot)$ as the objective function performs better than maximizing the cosine similarity $\cos(x)$. The reason is that, when two nodes are very close to each others, $1/d(\cdot)$ is more sensitive to the change of its Euclidean distance, whereas $\cos(\cdot)$ tends to be saturated. Therefore, the function $1/d(\cdot)$ has the desired property of non-saturating and can better focus on the distance difference with closest neighbors.

A.4 GENERAL-DOMAIN MULTI-MODAL MODELS

Although our method works well in both small and large scale datasets, as we will demonstrate in Section 4, there are still two interesting aspects that we would like to explore. First, in our approach, self-supervised models need to be re-trained for each new dataset, which is time-consuming and could potentially delay the schedule for data annotation in real-world industry. In addition, unsupervised models do not model semantic information explicitly, which may lead to confusion that could potentially be mitigated (e.g. datasets with varying intra-class variance will take regions of different sizes and may be treated differently in an unexpected way).

To address these issues, we put our focus on a large pretrained model that encodes semantic information. However, a large fully-annotated dataset, which is required to train a large supervised model, is very costly to obtain, and labelling the target dataset to train a supervised model defeats our purpose of requiring only a small amount of annotation on the target dataset. Fortunately, the availability of large-scale text-image pairs online makes it possible to train a large-scale model that encodes images in the general domain with semantic information. In this paper, we make use of publicly-available CLIP (Radford et al., 2021) models, a large-scale collection of models trained on Internet-crawled data with a wide general domain and use CLIP's image model as feature extractor.

As showed in Section 4, using models trained on multi-modal datasets answers these two issues. Even through CLIP is never trained on our target dataset, nor does the categories in its training set exactly matches the dataset we are using, using it to select does not degrade our performance of sample selection and labeling pipeline. This indicates that the effectiveness of our label selection does not necessarily depend whether same pretrained model is used in the downstream. In addition, we observe that such substitution even helps with a slightly larger annotation budget, demonstrating the effectiveness of making use of semantic information. Since we only perform inference on the CLIP model, the whole sample selection process could complete in *0.5 hours* on a commodity server using one GPU, indicating the possibility of our methods without delaying the schedule of human annotation or modifying the annotation pipeline and enables it to be used by industry on real-world dataset collection.

Note that although CLIP supports zero-shot inference by using text input (e.g. class names) to generate weights for classifier, it is not always possible to define a class with names or even know all the classes beforehand. Since we only make use of the image part of the CLIP model, we do not make use of prior text information (e.g. class descriptions) that are sometimes available in the real world. We leave better integration of our methods and zero-shot multi-modal models to future work.

A.5 HYPERPARAMETER ANALYSIS

We focus on two most important hyperparameter here: λ, the weight for regularization, and k, the number of neighbors we use for kNN. For hyperparam λ, we evaluated label selections with different λ values used in regularization. In this series of experiments, we select $\lambda \in \{0, 0.1, 0.5, 1.0, 3.5, 7.0\}$, where 0 indicates no regularization and larger λ indicates a stronger regularization. We then evaluate the mean accuracy from 10 runs, the standard deviation of classes, the difference with selection without regularization (i.e., $\lambda = 0$), and mean first order metric. We observe that as λ gets larger, we select more different samples compared to the case without regular-

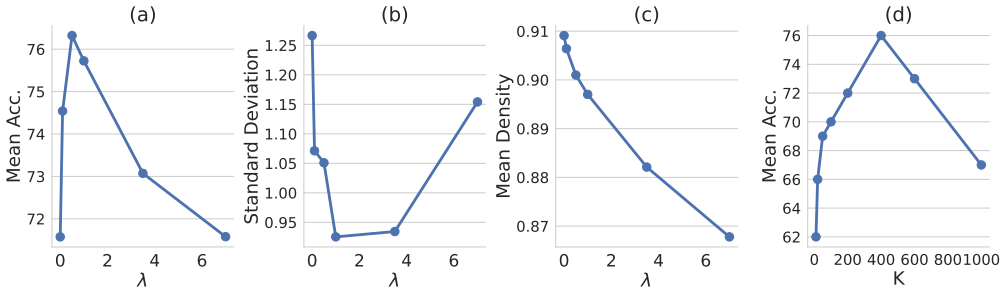


Figure 8: Effect of different hyperparameters, λ (Fig.a,b,c) and K (Fig.d). λ balances representative and uniformity across the feature space. Larger λ indicate more uniform choices but potentially less representative samples, or vice versa. Larger K indicates that we are taking more neighbors into account when estimating the representativeness, which reduces variance but may consider non-relevant samples as being represented. Due to this trade-off, considering more neighbors usually leads to better representativeness estimates, until K is greater than its optimal choice, 400.

ization, and instance standard deviation gets lower since we are sampling more uniformly across the space. However, as a trade-off, we could not sample from area which has as high density as before because selecting samples from that area leads to selections that are close to each other, leading to high penalty. Here, uniformity and representativeness show a trade-off and the optimal choice is to balance each other at $\lambda = 0.5$, which is demonstrated by the mean accuracy. For hyperparam k, we find that although initially adding k contributes to a better representation estimation by considering more neighbors, adding too much to k makes the selection algorithm lose focus. We find that k around 400 is optimal for our scenario.

A.6 Run Time Discussion

CLD only takes about 4 hours to train on CIFAR-10 on a single GPU and sample selection with our method takes less than 10 minutes on CLD with one GPU. This takes significantly less GPU-time than FixMatch (120 GPU hours with 4 GPUs), which is, in turn, much less than the time for labelling the whole dataset of 50, 000 samples. On ImageNet, MoCo takes about 12 days with 8 GPUs to achieve 800 epochs (He et al., 2020), our algorithm takes about an hour on one GPU to select samples for both 1% and 0.2% labels and in the end, FixMatch takes another 20 hours on 4 GPUs to train. Although it sounds like we are using a lot of compute time just to train a self-supervised learning model for selecting what samples to annotate, the fact is that FixMatch requires a self-supervised pretrained checkpoint to work well when the number of labeled samples is low, as demonstrated in Cai et al. (2021). The only compute overhead introduced is the sample selection process, which is *negligible* when compared to the other two stages. In addition, shown in our experiments, CLIP, as a model trained on a general and diverse image-text dataset, could also be used to select samples with comparable and sometimes even better samples to label, which indicates that the self-supervised training stage is not required in our method for sample selection.

A.7 Experiment Setup For CIFAR-10

Setup for FixMatch. To maintain consistency with the original FixMatch (Sohn et al., 2020), we evaluate FixMatch trained on CIFAR-10 with 2^{20} steps in total. To illustrate the ability of our algorithm to select informative samples, we evaluate both approaches on an extremely-low setting from 40 samples to 250 samples in total (4 shots to 25 shots per class on average). Since the original FixMatch is evaluated with stratified sampling on CIFAR-10, we also retrain FixMatch with random sampling with the same number of samples in total as a fair comparison. Unless otherwise stated, we train FixMatch with a learning rate of 0.03, and weight decay 10^{-3} on 4 Nvidia RTX 2080 Ti GPUs with batch size 64 for labeled samples and with 2^{20} steps in total. All experiments are conducted with same training and evaluation recipe for fair comparisons.

Setup For Transfer Learning. In addition to pseudo-label based semi-supervised learning methods, we also evaluate our algorithm on two-stage SSL based on transfer learning (Chen et al., 2020b) by fine-tuning the linear layer of a ResNet-18 pretrained with self-supervised learning algorithm CLD (Wang et al., 2021). Specifically, we fine-tune the linear layer on a ResNet-18 trained with CLD

(Wang et al., 2021). Since it is easy for the network to overfit the few-shot labeled samples, we freeze the backbone and fine-tune only the linear layer. We use SGD with learning rate 0.01, momentum 0.9, and weight decay 10^{-4} for 5 epochs because longer training time will lead to overfitting.

A.8 Experiment Setup for ImageNet

To evaluate the effectiveness of our method on large-scale datasets with more classes, we perform evaluation of our method on ImageNet (Russakovsky et al., 2015) with approximately 1 million images and 1000 classes. We perform supervised learning and semi-supervised learning on both 1% labeled data (12820 samples) and the extremely data-scarce setting 0.2% (2911 samples). For each of these settings, we use either a MoCo-pretrained model with Exponential Moving Average Normalization (EMAN) (Cai et al., 2021) or a CLIP ViT/16 model (Dosovitskiy et al., 2020) to select samples to annotate. For ImageNet 1%, we run K-Means clustering with 12900 clusters, which is slightly more than 12820 samples we are selecting, because we observe that there will sometimes be empty clusters. For ImageNet 0.2%, we use the same number of clusters as the number of samples that we will select. In the supervised learning setting, we use a batch size of 256, learning rate 0.01, and weight decay 10^{-3} and train a ResNet-50 for 1000 epochs, with learning rate decayed by 0.1 at 700, 800, and 900 epochs. Since supervised learning only uses labeled data, training for 1000 epochs with 1% data takes about the same time to train the same network for 10 epochs with full data labeled. In the semi-supervised learning setting, we use FixMatch Sohn et al. (2020) with EMAN as our semi-supervised learning algorithm, and to maintain consistency with prior works, we use the same setting as in Cai et al. (2021) besides the selection of input labeled data, unless otherwise stated. Specifically we use a learning rate of 0.03 with weight decay $10^{-4}$and train a ResNet-50 for 50 epochs with a MoCo (He et al., 2020) model as pretrained model. We perform learning rate warmup for 5 epochs and decay the learning rate by 0.1 at 30 and 40 epochs. Note that we load MoCo model as the pretrained model for FixMatch for fair comparison so that the only difference between MoCo and CLIP setting is the sample selection.