ContraCluster: Learning to Classify without Labels by Contrastive Self-Supervision and Prototype-Based Semi-Supervision

Seongho Joe, Byoungjip Kim, Hoyoung Kang, Kyoungwon Park, Bogun Kim, Jaeseon Park, Joonseok Lee, Youngjune Gwon

Samsung SDS, Seoul, South Korea

Email: {drizzle.cho, bjip.kim, hoyoung.kang, kw621.park, bogun0.kim, jaeseon.park, js1985.lee,

gyj.gwon}@samsung.com

Abstract—The recent advances in representation learning inspire us to take on the challenging problem of unsupervised image classification tasks in a principled way. We propose ContraCluster, an unsupervised image classification method that combines clustering with the power of contrastive self-supervised learning. ContraCluster consists of three stages: (1) contrastive selfsupervised pre-training (CPT), (2) contrastive prototype sampling (CPS), and (3) prototype-based semi-supervised fine-tuning (PB-SFT). CPS can select highly accurate, categorically prototypical images in an embedding space learned by contrastive learning. We use sampled prototypes as noisy labeled data to perform semisupervised fine-tuning (PB-SFT), leveraging small prototypes and large unlabeled data to further enhance the accuracy. We demonstrate empirically that ContraCluster achieves new state-of-theart results for standard benchmark datasets including CIFAR-10, STL-10, and ImageNet-10. For example, ContraCluster achieves about 90.8% accuracy for CIFAR-10, which outperforms DAC (52.2%), IIC (61.7%), and SCAN (87.6%) by a large margin. Without any labels, ContraCluster can achieve a 90.8% accuracy that is comparable to 95.8% by the best supervised counterpart.

I. INTRODUCTION

Supervised learning approaches in deep learning have shown to provide a human-level performance in computer vision tasks such as image classification [1] and object detection [2]. Unsupervised learning, in contrast, has long been considered too challenging for discriminative machine learning tasks [3], [4] and difficult to provide an accuracy comparable to that of supervised learning.

But advances in self-supervised representation learning makes it possible for pre-train models to learn general features from unlabeled data by generating pretext tasks [5]–[8]. Recently, contrastive self-supervised representation learning such as CPC [9], DIM [10], MoCo [11], and SimCLR [12] has significantly enhanced the quality of learned representations by using the InfoMax principle [13].

It enables the paradigm of unsupervised pre-training followed by fine-tuning with few labels [12]. For example, SimCLRv2 [14] propose a distillation stage that takes place after supervised fine-tuning. S4L [15] and SelfMatch [16] show that semi-supervised fine-tuning further enhance the accuracy and label efficiency.



Fig. 1: An example clustering result of ContraCluster for CIFAR-10. For visualization, ten images are randomly sampled from the final clustering result. Each rows is a cluster discovered by ContraCluster. Red rectangles denotes misclassified images in each cluster. In this example, there are only 8 errors over 100 samples. This example approximately shows that ContraCluster provides such high accuracy (i.e., 90.8%) without labels.

Combining them, end-to-end unsupervised learning schemes emerge, such as SCAN [17] and RUC [18]. SCAN consists of three stages: (1) contrastive self-supervised pre-training (SimCLR), (2) fine-tuning with SCAN loss, and (3) finetuning with self-labeling. The main idea of SCAN is the SCAN loss neighborhood consistency that encourages the model to make consistent predictions between a sample and its neighboring samples. RUC is based on SCAN, and proposes additional fine-tuning stages: (4) clean sample selection based on confidence scores, and (5) semi-supervised fine-tuning with MixMatch [19], which is known as interpolation consistency regularization.

They are robust methods and show high performances. Some weak points, however, exist. the SCAN loss has high



Fig. 2: The overview of ContraCluster. It consists of three stages: (1) contrastive self-supervised pre-training (CPT), (2) contrastive prototype sampling (CPS), and (3) prototype-based semi-supervised fine-tuning (PB-SFT). It selects highly accurate prototypical samples (i.e., prototypes) from an embedding space learned by contrastive self-supervised pre-training. They are used as noisy labeled data in PB-SFT. Note that, in the proposed pipeline, ContraCluster does not use any human-labeled data to classify images.

computational complexity, as it needs to select k neighboring samples for each sample at every optimizing steps. And MixMatch used in RUC suffers from relatively low accuracy when only small number of labeled data available. Moreover, SCAN uses k-Nearest Neighbor (k-NN) for sematic clustering, and RUC utilizes both k-NN and confidence scores. But, k-NN is hard to reveal global structure of embedding space. It could mix noisy cluster boundaries in the pseudo-label, resulting memorization. It is also well known that confidence scores provided by neural networks are not good estimates for the uncertainty of class assignment.

We introduce ContraCluster, an unsupervised image classification method that leverages the advances of contrastive self-supervised learning via clustering. Figure 1 shows an example clustering result of ContraCluster. As shown in Figure 2, ContraCluster consists of three stages: (1) contrastive self-supervised pre-training (CPT), (2) contrastive prototype sampling (CPS), and (3) prototype-based semi-supervised finetuning (PB-SFT).

In the first stage, we aim to discover a linearly separable embedding space by using only unlabeled data. To achieve this goal, we perform *contrastive self-supervised pre-training* (CPT). Among many promising methods, we adopt SimCLR [12]. Unlike SCAN, we directly use SimCLR results, resulting simpler pipeline. (for details see III-A).

For the second stage, we develop *contrastive prototype sampling* (CPS) that selects prototypical images that are highly categorical from the learned embedding space in the first stage. (see Figure 4). Conceptually, highly categorical prototypes are sampled based on cluster centroids in a projected embedding space (see Figure 2). The main idea is that cluster centroids in a low-dimensional space approximately represent the most discriminative samples. (for details see III-B).

In the third stage, we use the prototypes as noisy labeled data to perform *prototype-based semi-supervised fine-tuning* (PB-SFT) that can increase the accuracy by leveraging both small noisy labeled data (i.e., prototypes) and large unlabeled data. PB-SFT can avoid the problem of over-fitting during fine-tuning with few labeled data. For leveraging unlabeled data, we adopt FixMatch [20], one of the most successful single-stage semi-supervised learning method. (for details see III-C).

We empirically demonstrate that ContraCluster achieves new state-of-the-art results for standard benchmark datasets including CIFAR-10 [21], STL-10 [22], and ImageNet-10 [23] (see Table III). For CIFAR-10, ContraCluster achieves about 90.8% accuracy that outperforms strong previous method such as DAC [3] (52.2%), IIC [4] (61.7%), and SCAN [17] (87.6%) by a large margin. Note also that without labels, ContraCluster can achieve about 90.8% accuracy that is comparable with the accuracy of supervised learning with full labels (95.8%) [24]. Note that our method cannot be directly compared to deep clustering like [25]–[27], because ours does not repeat clustering procedure for new data, but inference classes with the trained model.

Our contributions are summarized as follows.

- We propose novel unsupervised image classification method of robust prototype sampling.
- We empirically achieve new state-of-the-art results for standard benchmark datasets.

The rest of this paper is organized as follows. Section 2 discusses related work. In Section 3, we explain ContraCluster in detail. Section 4 presents our experimental results on the standard image benchmarks. The paper concludes in Section 5.

II. RELATED WORK

A. Self-supervised representation learning

a) Task-specific self-supervised learning.: Selfsupervised learning aims to learn general representations from unlabeled data by performing a pretext task, then to reuse them in downstream tasks. For example, the pretext task includes context prediction [5], jigsaw puzzle solving [6], image colorization [7], and rotation prediction [8].

b) Contrastive self-supervised learning.: Contrastive self-supervised learning extracts general representations from unlabeled data by using contrastive loss, which is based on the InfoMax principle [13] that encourages the agreement between

multiple views from an instance. It provides much higher quality visual representations in terms of the linear separability than the task-specific self-supervised learning. They include CPC [9], DIM [10], AM-DIM [28], MoCo [11], and SimCLR [12].

B. Semi-supervised learning

Semi-supervised learning enables to learn from small labeled data by leveraging large unlabeled data. Approaches to semi-supervised learning includes pseudo-labeling [29], entropy minimization [30], and consistency regularization [31]– [33]. We mainly discuss consistency regularization in this paper.

a) Consistency regularization.: Consistency regularization aims to use unlabeled data to regularize the cross-entropy loss with few labeled data. Its objective encourages a model to predict consistent class probabilities over stochastically transformed samples. It has been introduced in Π -Model [31] and further developed by MeanTeacher [32], [33]. Recently, advanced methods are introduced. They include MixMatch [19], UDA [34], ReMixMatch [35], FixMatch [20].

b) Self-supervised pre-training-based.: Recent work extends the self-supervised paradigm with more sophisticated fine-tuning techniques. For example, SimCLRv2 [14] proposes to use the third stage of distillation after supervised fine-tuning with few labels. S4L [15] and SelfMatch [16] show that semisupervised fine-tuning further enhance the accuracy and label efficiency.

C. Unsupervised classification / clustering

a) From-scratch approach.: Unsupervised image classification, or image clustering, can be broadly categorized into two: generative and discriminative. Generative approach attempts to learn general representations by using reconstruction or adversarial losses. This approach includes Autoencoder (AE) [36], GAN [37], VAE [38], and ClusterGAN [39]. In contrast, discriminative approach tries to learn general representations by using unsupervised loss that encourages proper cluster assignment in a label space. This approach includes DEC [40], DAC [3], DeepCluster [41], IIC [4], DCCM [42]. However, these two approaches usually suffered from relatively low accuracy since they do not directly optimize representations in an embedding space.

b) Self-supervised pre-training-based.: Most recently, the use of contrastive self-supervised pre-training has been proposed for unsupervised image classification or clustering. Since contrastive self-supervised pre-training aims to learn general representations by directly optimizing them in the embedding space, it has a huge potential to improve the clustering accuracy compared to the previous approaches. This approach includes SCAN [17] and RUC [18].

c) Comparison with other methods.: SCAN [17] proposes to fine-tune the SimCLR [12] pre-trained encoder by using an unsupervised loss that encourages the similarity of cluster assignment probabilities between a sample and k nearest neighborhoods. RUC [18] proposes to further fine-tune



Fig. 3: The model architecture of ContraCluster.

the SCAN model by using *interpolation consistency regularization* (e.g., MixMatch [19]). In contrast to these methods, ContraCluster proposes to fine-tune the pre-trained encoder by using prototype-based semi-supervised fine-tuning (PB-SFT). To achieve this goal, we develop contrastive prototype sampling (CPS) that selects categorically-accurate prototypes to use as noisy labeled data for semi-supervised learning. Similar to ContraCluster, RUC selects clean labeled data by using confidence. However, ContraCluster selects clean labeled data represented by prototypes based on cluster centroids that can be discovered in embedding space.

III. METHOD

In this section, we describe the details of ContraCluster. Figure 3 shows the model architecture. It consists of three stages: (1) contrastive self-supervised pre-training (CPT), (2) contrastive prototype sampling (CPS), and (3) prototype-based semi-supervised fine-tuning (PB-SFT). The weight of the encoder pre-trained in the first stage is transferred in the following stages. In the final stage, the pre-trained encoder is further fine-tuned by using both small noisy labeled data (i.e., prototypes) and large unlabeled data. The model $p_{model}(y|x)$ consists of an encoder $f(\cdot)$ and a head $c(\cdot)$. The learning algorithm of ContraCluster is presented in Algorithm 1.

A. Contrastive self-supervised pre-training

ContraCluster aims to learn a linearly separable embedding space by using only unlabeled data in the first stage. For CPT, it adopts SimCLR [12], one of the most effective methods. SimCLR learns representations by simultaneously encouraging two objectives: (1) maximizing the similarity between different views \tilde{u}_i and \tilde{u}_j from the same sample and (2) minimizing the similarity between different views \tilde{u}_i and \tilde{u}_k ($k \neq i$) from different samples.

As shown in Figure 3, SimCLR consists of four components: (1) data augmentation $\mathcal{T}(\cdot)$, (2) encoder $f(\cdot)$, (3)

Algorithm 1 The ContraCluster 3-stage learning algorithm.

Require: Unlabeled data \mathcal{U} **Require:** Randomly initialize encoder $f(\cdot)$, projection head $g(\cdot)$, and classification head $c(\cdot)$ 1: # Stage 1: Contrastive self-supervised pre-training 2: for n = 1 to E_{CPT} do for k = 1 to B_{CPT} do 3: $u_k \sim \mathcal{U} \\ t \sim \mathcal{T}, t' \sim \mathcal{T}$ ▷ unlabeled batch 4: 5:
$$\begin{split} t \sim \mathcal{T}, t' \sim \mathcal{I} \\ \tilde{u}_i &= t(u_k), \tilde{u}_j = t'(u_k) \quad \triangleright \text{ transformation} \\ h_i &= f(\tilde{u}_i), h_j = f(\tilde{u}_j) \quad \triangleright \text{ encoding} \\ z_i &= g(h_i), z_j = g(h_j) \quad \triangleright \text{ projection} \\ \mathcal{L}_{CPT} &= -\log \frac{\exp(\sin(z_i, z_j)/\tau)}{\sum_{k=1}^{2B_{CPT}} \mathbbm{1}(k \neq i) \exp(\sin(z_i, z_k)/\tau)} \\ & \triangleright \text{ Eq. 1} \end{split}$$
6: 7: 8: 9: $SGD(\eta_{CPT})$ 10: 11: # Stage 2: Contrastive prototype sampling 12: for $\forall u_i \in \mathcal{U}$ do 13: $\mathcal{H}_{high} \leftarrow f(u_i)$ 14: $\mathcal{Z}_{low} = projection(\mathcal{H}_{high}, N_{neigh}, N_{dim})$ 15: $C_k = clustering(\mathcal{Z}_{low}, k_{part})$ 16: $\mathcal{P} = sampleCentroidNeighbors(\mathcal{C}_k, N_{proto})$ 17: # Stage 3: Prototype-based semi-supervised fine-tuning 18: for n = 1 to E_{SFT} do for k = 1 to B_{SFT} do 19: $x_k \sim \mathcal{P}$ ▷ prototype batch 20: $t \sim T$ 21: $\begin{aligned} \tilde{x}_k &= t(x_k) \\ q_k &= c(f(\tilde{x}_k)) \\ \mathcal{L}_{proto} &= \frac{1}{B_{SFT}} \sum_{k=1}^{B_{SFT}} H(\hat{y}_k, q_k) \end{aligned}$ 22: 23: ⊳ Eq. 3 24: for i = 1 to μB_{SFT} do 25: $u_i \sim \mathcal{U}$ ▷ unlabeled batch 26: $\begin{array}{ll} u_i \sim \varkappa \\ t \sim \mathcal{T}_{weak}, t' \sim \mathcal{T}_{strong} \\ \tilde{u}_i^w = t(u_i) \\ \tilde{u}_i^s = t'(u_i) \\ q_i^w = c(f(\tilde{u}_i^w)) \end{array} \qquad \triangleright \mbox{ weak augmentation} \\ \rho \mbox{ strong augmentation} \\ \rho \mbox{ bard prediction} \end{array}$ 27: 28: 29: 30: $\begin{array}{l} q_i = c(f(\tilde{u}_i^s)) & \triangleright \text{ hard} \\ \hat{q}_i^w = argmax(q_i^w) & \triangleright \text{ consist} \\ m = \mathbb{1}(\max(q_i^w) \ge c) \\ \mathcal{L}_{consi} = \frac{1}{\mu B_{SFT}} \sum_{i=1}^{\mu B_{SFT}} mH(\hat{q}_i^w, q_i^s) \end{array}$ ▷ hard prediction 31: ▷ consistency target 32: 33: ⊳ mask 34: ⊳ Eq. 4 $\mathcal{L}_{PB-SFT} = \mathcal{L}_{proto} + \lambda_u \mathcal{L}_{consi}$ ⊳ Eq. 2 35: $SGD(\eta_{SFT})$ 36: **return** encoder $f(\cdot)$, classification head $c(\cdot)$

projection head $g(\cdot)$, and (4) contrastive loss \mathcal{L}_c . The data augmentation uses random crop and color distortion. The encoder is ResNet-50 [1]. The projection head is a two-layer MLP with dropout [43] and ReLU activation. Finally, the contrastive loss is formulated as follows:

$$\mathcal{L}_{CPT} = -\log \frac{\exp(\operatorname{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2B_{CPT}} \mathbb{1}(k \neq i) \exp(\operatorname{sim}(z_i, z_k)/\tau)} \quad (1)$$

where B_{CPT} is a batch size, $\mathbb{1}(k \neq i)$ is an indicator function

evaluating to 1 only if $k \neq i$, $sim(\cdot)$ is a similarity function, and τ is a temperature parameter scaling the similarity. The contrastive learning hyperparameters are listed in Table II.

B. Contrastive prototype sampling

CPS is to select highly accurate prototypical images from the embedding space of the first stage. For prototypes, CPS simply chooses n nearest neighbors from the cluster centroids. To do so, CPS first reduces the dimension of space $(\mathcal{H}_{high} \rightarrow \mathcal{Z}_{low})$ by using a non-linear dimensionality reduction algorithm such as UMAP [44] and t-SNE [45]. Then, kmeans clustering [46] (C_k) applies on the projected embedding space (e.g., a 2-dimensional space) to find cluster centroids, assuming the groud-truth number of clusters is given, holding the same condition with previous works [17], [18] to compare. Finally, CPS selects n nearest neighbors (\mathcal{P}) from the cluster centroids. We have empirically found out that non-linear dimensionality reduction is essential to increase the accuracy (see Table IV). We conjecture that this is mainly due to the fact that classic clustering algorithms such as k-means suffer from the low accuracy problem because of the curse of dimensionality. Note that DBSCAN [47], an alternative to k-means, cannot be applied by lack of cluster centroids.

To determine the proper hyperparameter values for UMAP (e.g., # of neighbors N_{neigh} , # of dimension N_{dim} , etc.), we use Silhouette Coefficient [48] (see Figure 4). The Silhouette Coefficient (y-x)/max(x,y) is evaluated on the mean intracluster distance (x) and the mean nearest-cluster distance (y) for each sample. Since Silhouette Coefficient is calculated with clustering results (i.e., no ground-truths required), we can approximately choose the best hyperparameters of UMAP and k-means by using only unlabeled data. Hyperparameters are summarized in Table II.

C. Prototype-based semi-supervised fine-tuning

PB-SFT can further increase the accuracy by leveraging both small noisy labeled prototypes and (large) unlabeled data. We adopts FixMatch [20] that exploits augmentation-based consistency regularization for unlabeled data. It encourages the consistent prediction between weakly and strongly augmented examples. More specifically, its objective function is to minimize the cross entropy H(p,q) between the prediction \hat{q}_i^w of a weakly augmented input \tilde{u}_i^w and the class probability distribution q_i^s of a strongly augmented input \tilde{u}_i^s (see Figure 3). The weak augmentation uses random crops and horizontal flips. The strong one adopts RandAugment (RA) [49], an effective automated method. The classification head $c(\cdot)$ is a MLP of two layers with dropout [43] and ReLU activation.

The loss function consists of prototype-based cross-entropy loss and consistency regularization loss:

$$\mathcal{L}_{PB-SFT} = \mathcal{L}_{proto} + \lambda_u \mathcal{L}_{consi} \tag{2}$$

The prototype-based supervised loss \mathcal{L}_{proto} is formulated as follows:

$$\mathcal{L}_{proto} = \frac{1}{B_{SFT}} \sum_{k=1}^{B_{SFT}} H(\hat{y}_k, q_k)$$
(3)

where B_{SFT} is a fine-tuning batch size, \hat{y}_i is a noisy label of a prototype.

The augmentation-based consistency regularization loss \mathcal{L}_{consi} is formulated as follow:

$$\mathcal{L}_{consi} = \frac{1}{\mu B_{SFT}} \sum_{i=1}^{\mu B_{SFT}} \mathbb{1}(\max(q_i^w) \ge c) H(\hat{q}_i^w, q_i^s) \quad (4)$$

where c is a confidence threshold, $\mathbb{1}(\max(q_i^w) \ge c)$ is an indicator function, B_{SFT} is a fine-tuning batch size, and μ is the ratio of prototypes and unlabeled samples in a batch.

For training, it utilizes Exponential Moving Average (EMA) [32] with a weight decay for stable training and inference. The hyperparameters are described in Table II. Many of them follows SimCLR [12] and FixMatch [20], to make a fair comparison with the other methods.

IV. EXPERIMENTS

A. Datasets

We empirically validate ContraCluster using standard benchmark datasets: CIFAR-10 [21], STL-10 [22], and ImageNet-10 [23], as in Table I.

| Dataset | CIFAR10 | STL10 | ImageNet10 |
|---------------|---------|-------------|------------|
| Size | 32x32 | 96x96 | 224x224 |
| Classes | 10 | 10 | 10 |
| Train split | train | train+test | train |
| Test split | test | train+test | train |
| Train samples | 50,000 | 5,000+8,000 | 13,000 |
| Test samples | 10,000 | 5,000+8,000 | 13,000 |

TABLE I: Summary of datasets.

B. Hyperparameter setting

| Hyperparameters | CIFAR10 | STL10 | ImageNet10 |
|-------------------------------|---------|---------|------------|
| Temperature τ | 0.1 | 0.1 | 0.1 |
| Batch size B_{CPT} | 512 | 256 | 64 |
| Optimizer | SGD | SGD | SGD |
| Learning rate η_{CPT} | 0.6 | 0.3 | 0.075 |
| Max epoch E_{CPT} | 1024 | 1024 | 1024 |
| # of neigh. N_{neigh} | 20 | 50 | 50 |
| Projection dim. N_{dim} | 2 | 2 | 2 |
| Min. distance D_{min} | 0.5 | 0.0 | 0.0 |
| Similarity metric | correl. | correl. | correl. |
| # of proto. N_{proto} | 250 | 1000 | 1000 |
| Batch size B_{SFT} | 64 | 64 | 64 |
| Unlab. batch ratio μ | 7 | 7 | 7 |
| Unlab. loss ratio λ_l | 1 | 1 | 1 |
| Confidence thre. c | 0.95 | 0.95 | 0.95 |
| Optimizer | SGD | SGD | SGD |
| Learning rate η_{SFT} | 0.03 | 0.03 | 0.03 |
| Max epoch E_{SFT} | 400 | 400 | 400 |

TABLE II: Hyperparameters of ContraCluster.

Table II presents a complete list of the hyperparameters. Each partition of the table shows the values used in the stage one to three respectively. They are empirically determined. C. Hyperparameter selection for contrastive prototype sampling



Fig. 4: Hyperparameter selection with Silhouette Coefficient for contrastive prototype sampling and prorotype accuracy of ContraCluster. (Left) w.r.t the projection dimension. (Center) w.r.t the number of neighbors. (Right) w.r.t prototype accuracy.

To choose the proper hyperparamter values for CPS, we use Silhouette Coefficient. Figure 4 shows the variation of it with respect to # of neighbors N_{neigh} and the projected dimension N_{dim} of UMAP. For CIFAR-10, we choose 2 for N_{dim} and 20 for N_{neigh} , where it is the highest (see Table II).

D. Unsupervised image classification accuracy

Table III shows a comparison of unsupervised image classification performance measured in accuracy (%) and NMI (normalized mutual information) [52]. Asterisked (*) results come from the SCAN [17] paper, and the others from respective original publications. We provide both the mean and maximum accuracy of ContraCluster. The mean is computed by averaging five evaluations with different random seed numbers. It achieves state-of-the-art results for CIFAR-10, STL-10, and ImageNet-10.

a) CIFAR-10.: ContraCluster achieves a 90.8% classification accuracy that outperforms DAC (52.2%), IIC (61.7%), and SCAN (87.6%) by significant margins. Note that, without any labels, it is comparable with the accuracy of supervised learning with full labels (95.8%).

b) STL-10.: ContraCluster achieves a 87.5% accuracy that also outperforms DAC (47.0%), IIC (59.6%), and SCAN (76.7%) significantly.

c) ImageNet-10.: ContraCluster achieves a 90.5% accuracy, which corresponds to outperform notable existing methods such as DCCM (71.0%) and CC (89.3%).

E. Prototype accuracy

Figure 4 shows the variation of prototype accuracy with respect to the number of them. This figure shows that Contra-Cluster can select highly accurate prototypes (about 95.0%) that can be used as noisy labeled data for PB-SFT. We choose 250 prototypes for CIFAR-10 (about 96.5%), 1,000 for STL-10 (about 96.2%), and 1,000 for ImageNet-10 (about 94.0%), all based on empirical trials. For CIFAR-10 and STL-10, although 40 prototypes provide the highest accuracy

| | CIF | CIFAR-10 STL-10 (tra | | (train+test) | ain+test) ImageNet-10 | |
|---|---------------------------------|----------------------|---------------------------------|----------------|---------------------------------|----------------|
| Method | Acc.(%) | NMI | Acc.(%) | NMI | Acc.(%) | NMI |
| Supervised (full labels) [24] | 95.8 | - | | - | 91.4 | - |
| k-means* [46] | 22.9 | 0.087 | 19.2 | 0.125 | - | - |
| Spectral clustering* [50] | 24.7 | 0.103 | 15.9 | 0.098 | | - |
| Autoencoder (AE)* [36] | 31.4 | 0.234 | 30.3 | 0.250 | - | - |
| DCGAN* [37] | 31.5 | 0.265 | 29.8 | 0.210 | - | - |
| ClusterGAN [39] | 41.2 | 0.323 | 42.3 | 0.335 | - | - |
| DEC* [40] | 30.1 | 0.257 | 35.9 | 0.276 | - | - |
| DAC* [3] | 52.2 | 0.400 | 47.0 | 0.366 | - | - |
| DeepCluster* [41] | 37.4 | - | 33.4 | - | - | - |
| DCCM [42] (only train set) | 62.3 | 0.496 | 48.2 | 0.376 | 71.0 | - |
| IIC* [4] | 61.7 | 0.511 | 59.6 | 0.496 | - | - |
| CC [51] | 79.0 | 0.705 | 85.0 | 0.764 | 89.3 | 0.859 |
| SCAN* [17] (only train set) | $87.6 {\pm} 0.4$ | 0.787 | 76.7±1.9 | 0.680 | - | - |
| RUC (Conf.) [18] | 90.3 | - | 86.7 | - | - | - |
| ContraCluster (avg.) ContraCluster (max) | 90.8 ±0.5 91.7 | 0.837 0.857 | 87.5 ±0.3 87.9 | 0.784 0.787 | 90.2 ±0.4 90.5 | 0.804 0.809 |

TABLE III: Comparison of unsupervised image classification accuracy.

(more than 95.0%), sufficient number of prototypes (i.e., more than 100) is required for PB-SFT to achieve high clustering accuracy.

F. Ablation study

| Method | Accuracy(%) |
|----------------------------|--------------|
| ContraCluster w/o SimCLR | 29.0 (-61.8) |
| ContraCluster w/o UMAP | 82.4 (-8.4) |
| ContraCluster w/o FixMatch | 84.4 (-6.4) |
| ContraCluster | 90.8 |

TABLE IV: Ablation study of ContraCluster for CIFAR-10.

Table IV shows an ablation study result of ContraCluster for CIFAR-10. It proves that the each stage is essential for achieving the stage-of-the-art results. ContaCluster w/o SimCLR means applying UMAP and k-means on raw pixels (i.e., a sample space) without performing CPT. Since it is very difficult to capture semantic information from high-dimensional raw pixels, it shows significant performance degradation. Without UMAP, ContraCluster does not provide the state-of-the-art accuracy because UMAP can effectively help find cluster centroids in a low-dimensional space. Note that it is one reason why trivial application of the projection head of SimCLR as prototype sampler is suboptimal. Finally, without FixMatch, we could not have achieved the state-of-the-art either. This shows that PB-SFT is effective to further increase the accuracy.

G. Example results

a) Embedding space.: Figure 5 shows an example of the embedding space learned by ContraCluster for CIFAR-10. The left side shows an projected embedding space of UMAP. The right side shows a clustered embedding space by k-means. We present the more examples in the appendix.



Fig. 5: Visualization of the embedding space. It is learned by ContraCluster for CIFAR-10. (Left) a projected embedding space by UMAP. (Right) a clustered embedding space by k-means.

b) Clustering results.: Figure 1 shows an example of the final clustering result by ContraCluster. The example shows 90.8% class-accurate clustering (see Table III). We present more examples in the appendix.

V. CONCLUSION

We have presented ContraCluster, an unsupervised image classification method based on contrastive self-supervised learning. Combining the three stages, (1) contrastive self-supervised pre-training (CPT), (2) contrastive prototype sampling (CPS), and (3) prototype-based semi-supervised fine-tuning (PB-SFT), it build a high-performance classification pipeline without relying on labeled data. Our experimental evaluation indicates that it achieves new state-of-the-art results on CIFAR-10, STL-10, and ImageNet-10.

REFERENCES

 K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

- [2] K. He, G. Gkioxari, P. Dollr, and R. Girshick, "Mask rcnn," in IEEE/CVF International Conference on Computer Vision (ICCV), 2017.
- [3] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, "Deep adaptive image clustering," in *Proceedings of the IEEE International Conference* on Computer Vision (ICCV), 2017, pp. 5879–5887.
- [4] X. Ji, J. F. Henriques, and A. Vedaldi, "Invariant information clustering for unsupervised image classification and segmentation," in *Proceedings* of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019.
- [5] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015.
- [6] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *European Conference on Computer Vision (ECCV)*. Springer, 2016.
- [7] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *European Conference on Computer Vision (ECCV)*, 2016.
- [8] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *International Conference on Learning Representations (ICLR)*, 2018.
- [9] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," arXiv preprint arXiv:1807.03748, 2018.
- [10] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," in *International Conference* on Learning Representations (ICLR), 2019.
- [11] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *IEEE/CVF Conference* on Computer Vision and Pattern Recognition (CVPR), 2020.
- [12] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning (ICML)*, 2020.
- [13] M. Tschannen, J. Djolonga, P. K. Rubenstein, S. Gelly, and M. Lucic, "On mutual information maximization for representation learning," in *International Conference on Learning Representations (ICLR)*, 2020.
- [14] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton, "Big self-supervised models are strong semi-supervised learners," in Advances in Neural Information Processing Systems (NeurIPS), 2020.
- [15] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, "S4I: Self-supervised semi-supervised learning," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.
- [16] B. Kim, J. Choo, Y.-D. Kwon, S. Joe, S. Min, and Y. Gwon, "Selfmatch: Combining contrastive self-supervision and consistency for semisupervised learning," in Advances in Neural Information Processing Systems (NeurIPS) Workshop, 2020.
- [17] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. Van Gool, "Scan: Learning to classify images without labels," in *European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 268–285.
- [18] S. Park, S. Han, S. Kim, D. Kim, S. Park, S. Hong, and M. Cha, "Improving unsupervised image clustering with robust learning," *arXiv* preprint arXiv:2012.11150, 2020.
- [19] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," in Advances in Neural Information Processing Systems (NeurIPS), 2019.
- [20] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, "Fixmatch: Simplifying semisupervised learning with consistency and confidence," in Advances in Neural Information Processing Systems (NeurIPS), 2020.
- [21] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.
- [22] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *International Conference on Artificial Intelligence and Statistics*, 2011.
- [23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [24] H. Pham and Q. V. Le, "Autodropout: Learning dropout patterns to regularize deep networks," arXiv preprint arXiv:2101.01761, 2021.
- [25] J. Huang, S. Gong, and X. Zhu, "Deep semantic clustering by partition confidence maximisation," in *Proceedings of the IEEE/CVF Conference* on Computer Vision and Pattern Recognition (CVPR), 2020.

- [26] J. R. Regatti, A. A. Deshmukh, E. Manavoglu, and U. Dogan, "Consensus clustering with unsupervised representation learning," in 2021 International Joint Conference on Neural Networks (IJCNN), 2021.
- [27] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, and X. Peng, "Contrastive clustering," in 2021 AAAI Conference on Artificial Intelligence (AAAI), 2021.
- [28] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," in Advances in Neural Information Processing Systems (NeurIPS), 2019.
- [29] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *ICML Workshop on Challenges in Representation Learning*, 2013.
- [30] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in Advances in Neural Information Processing Systems (NeurIPS), 2005.
- [31] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," in Advances in Neural Information Processing Systems (NeurIPS), 2016.
- [32] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in Advances in Neural Information Processing Systems (NeurIPS), 2017.
- [33] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," in *International Conference on Learning Representations (ICLR)*, 2016.
- [34] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, "Unsupervised data augmentation for consistency training," in *International Conference on Learning Representations (ICLR)*, 2020.
- [35] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel, "Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring," in *International Conference on Learning Representations (ICLR)*, 2020.
- [36] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle et al., "Greedy layerwise training of deep networks," in Advances in Neural Information Processing Systems (NeurIPS), 2007.
- [37] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *International Conference on Learning Representations (ICLR)*, 2016.
- [38] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in International Conference on Learning Representations (ICLR), 2014.
- [39] K. Ghasedi, X. Wang, C. Deng, and H. Huang, "Balanced self-paced learning for generative adversarial clustering network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4391–4400.
- [40] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *International Conference on Machine Learning* (*ICML*). PMLR, 2016, pp. 478–487.
- [41] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 132–149.
- [42] J. Wu, K. Long, F. Wang, C. Qian, C. Li, Z. Lin, and H. Zha, "Deep comprehensive correlation mining for image clustering," in *Proceedings* of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019.
- [43] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [44] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," arXiv preprint arXiv:1802.03426, 2018.
- [45] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." Journal of machine learning research, vol. 9, no. 11, 2008.
- [46] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 881–892, 2002.
- [47] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.
- [48] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

- [49] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition* Workshops, 2020.
- [50] A. Y. Ng, M. I. Jordan, Y. Weiss et al., "On spectral clustering: Analysis [50] X. F. Ng, M. F. Jordan, T. Weiss et al., On spectral clustering. Analysis and an algorithm," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2002, pp. 849–856.
 [51] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, and X. Peng, "Contrastive clustering," *arXiv preprint arXiv:2009.09687*, 2020.
- [52] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *The Journal of Machine Learning Research*, vol. 11, pp. 2837–2854, 2010.