Pattern Recognition Letters
journal homepage: www.elsevier.com

A Self-Supervised Deep Learning Framework for Unsupervised Few-Shot Learning and Clustering

Hongjing Zhang[a,**], Tianyang Zhan[a], Ian Davidson[a]

[a]*University of California, Davis, 1 Shields Ave, Davis, CA, 95616, USA*

ABSTRACT

The need to learn a good representation is a core problem central to AI. We present a self-supervised representation learning framework and demonstrate its use for few-shot classification and clustering. Our framework can be interpreted as repeatedly discovering new categories from learned embeddings and training a new embedding function with self-supervised signals to differentiate the discovered categories. In our framework, we first discover categories from unlabeled data, next we post-process the previous partition results to remove outliers and derive prototypes of each category, we then construct few-shot learning tasks with previously selected data and augmented virtual data, lastly we propose a iterative training algorithm to learn the final representation. By combining these components, we can considerably outperform previous baselines in unsupervised few-shot classification tasks on *mini*ImageNet and Omniglot data sets. We also validate our learned representation on clustering tasks and demonstrate that our framework further improves upon the recent deep clustering methods.

1. Introduction

Supervised visual representation learning has achieved great success in recent years. However, learning a good representation still requires much-labeled data under a supervised learning setting. Learning a useful visual representations without human supervision is a fundamental, understudied and long-standing problem [6]. Two popular learning paradigms for unsupervised representation learning are deep clustering and self-supervised learning.

Some works have been proposed for deep clustering [28, 14, 30]. This work is limited in that it adopts a multi-stage pipeline that pre-trains the auto-encoders with reconstruction loss and then applies a clustering module on it. Though these approaches yield better clustering results than traditional clustering methods on image data (as they learn a new representation for clustering), they are are not as useful for unseen (novel) classes/clusters. An alternative representative learning paradigm is self-supervised learning methods. Self-supervised learning approaches learn representations using objective functions similar to those used for supervised learning, but train networks to perform tasks where both the inputs and labels are derived from an unlabeled dataset. Many recent works [9, 31, 19, 12] have been proposed, for example, [12] performs data augmentations such as rotating the images with different degrees and then to train the neural networks to predict the corresponding degrees. Although these learned models may not fully match the performance of supervised-learned representations, they have proved to be better than purely unsupervised representation learning approaches. However, these approaches perform self-supervised learning via heuristics with varying assumptions that if not true may harm the generalization ability of the representation.

In this work, we address learning a representation without supervision and demonstrate its use on two popular learning tasks. Our approach (visualized in Figure 1) to representation learning takes several intuitive steps as follows.

Category Discovery. Here we perform unsupervised representation along with clustering to take the unlabeled instances and place them into different categories/groups ($C_1 \dots C_k$).

Post-Processing for Representative Data. Next, using an integer linear programming (ILP) formulation, we post-process the clustering partitions to get representative instances and balance discovered clusters in the previous step.

Construct Tasks for Category Differentiation. Now we

**Corresponding author: Tel.: +0-000-000-0000; fax: +0-000-000-0000;
e-mail: author@author.com (Hongjing Zhang)

use previously selected representative instances and then create augmented versions as our training set. Inspired by the recent success of meta-learning (which aims to learn from limited data to generate to unseen classes), we create few-shot learning tasks which *minimize* the intra-category variation whilst *maximizing* the inter-category embedding distance for better category separation.

Iterative Training. We iterative the above three steps to discover categories based on learned embedding and refining the embedding.

We demonstrate our representation learning scheme on two classic minimal supervision problems: clustering and few-shot classification. The few-shot classification here is a paradigm where the model has been learned for the *base* classes and then is transferred to learn to predict *novel* classes of which there are only a few examples available. We argue that a good visual representation should not only naturally separate images that belong to different semantic groups within the training set but also be applicable for unseen classes. We evaluate our learned embedding functions in two settings: we first study the unsupervised few-shot classification problem with benchmark data sets such as Omniglot and *mini*ImageNet; we then test our learned embedding function on clustering tasks to show whether our pipeline improves upon different initial unsupervised representations. Based on experimental results, we show our approach achieves state-of-the-art performance comparing to recent unsupervised few-shot classification baselines. Moreover, the clustering results demonstrate that our method consistently improves the embedding learned by unsupervised representation learning for clustering.

We summarize the contributions of our work as follows:

- We propose a framework to improve the unsupervised representation learning (see section 3).
- We validate our proposed model in unsupervised-few-shot learning settings and show our proposed model achieves state-of-the-art results for benchmark datasets. Experimental results on clustering analysis also show that our work can further improve the embedding generated via popular deep clustering baselines (see section 4).
- Our ablation study shows the contributions of different technical components. We find out both the integer linear programming based post-processing algorithm and iterative training strategy improves the quality of our learned representations.

We begin this paper by next briefly overviewing related research, after which we discuss our approach in section 3. We experimentally compare our methods to many different baselines in section 4 and finally conclude.

2. Related Work

Image Representation Learning. Deep unsupervised feature learning has been explored to learn informative representations of images. One line of this direction is learning various auto-encoders that learn a reduced feature representation that can reconstruct the inputs. For example, the auto-encoder [4], denoising auto-encoder [25], variational auto-encoder [17], and adversarial auto-encoder [3]. Additionally, deep generative models also learn the underlying latent information about images and many generative adversarial networks [13, 7, 8] have been proposed to encode visual information from an adversarial learning strategy.

Recent deep clustering works take the advantages of these unsupervised visual representation techniques to learn clustering favored representations. For example, [28] (DEC) fine-tune the embedding learned from stacked-denoising auto-encoder via a self-supervised signal to form tight clusters. Unlike DEC, [11] guides agglomerative clustering and feature learning jointly based on the over-clustering initialized by KNN.[5] does discriminative clustering by alternating between clustering the features of a convolutional neural network and using the clusters as labels to optimize the network weights via back-propagating a standard classification loss. Our work is similar in style to previous deep clustering works, which learn a representation that is general for clustering is useful. Differently, our framework further improves the representation learned by these works via clustering and self-supervised few-shot learning.

Unsupervised Meta-Learning. Recent work studies the unsupervised meta-learning problem, which focuses on how to generate tasks for meta-learning approaches and pre-train the few-shot learner to achieve comparable performance with supervised meta-learning. For example, [15] constructed tasks from unlabeled data via clustering from different views and runs meta-learning models over the constructed tasks. This paper's core idea is to leverage the unsupervised embeddings to propose tasks for a meta-learning algorithm to pre-train the model for potential supervised few-shot classification tasks. [16] allows unsupervised, model-agnostic meta-learning for few-shot classification problems by generating synthetic data with artificial labels. [2] uses a similar way to generate tasks with synthetic data and pseudo labels but train on a more advanced meta learner as [1]. Our work differs from these works as our model gets trained over unsupervised few-shot classification tasks as an intermediate learning step. Moreover, our work aims to learn a good representation that works for clustering and few-shot learning by iteratively fine-tune the embedding function learned with carefully designed few-shot learning tasks.

3. Methodology

In this section, we present details of the proposed representation learning scheme in which the model discovers different concepts first and then gradually learns to discriminate both the representative instances and augmented instances within different concepts. We introduce a new method to learn an unsupervised embedding function that is useful for both unsupervised few-shot classification tasks and image clustering.

3.1. Overview

The pipeline of our proposed framework is illustrated in Figure 1. Firstly, we leverage the existing image representation methods and embed all training data into one latent space and

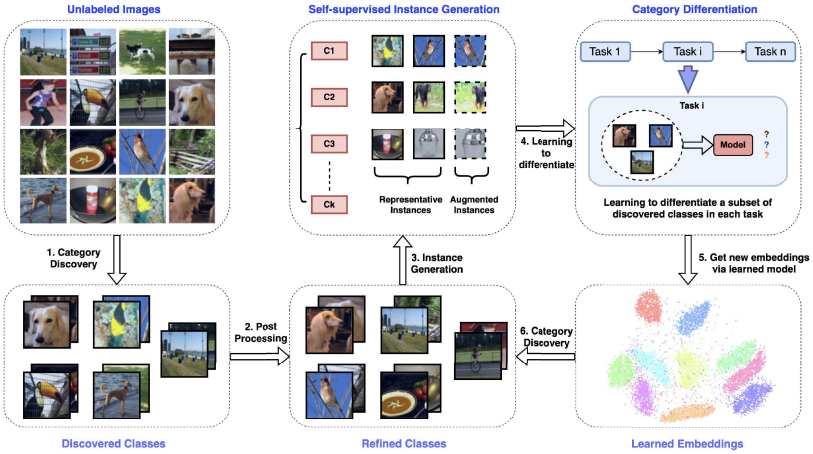


Fig. 1: Pipeline of the proposed unsupervised representation learning framework. The inputs are unlabeled images and the output is the learned embedding function. The learning process mainly contains fours steps: 1) category discovery, 2) category post-processing, 3) generate virtual instances to construct learning tasks, 4) learning to differentiate created categories.

conduct clustering. Secondly, given the clustered instances, we propose an integer linear programming based formulation to identify some unlabeled instances for representative and balanced clusters. Thirdly, we augment the selected representative instances of each discovered category to enrich the training data. Finally, we train our embedding function with Prototype-based networks using self-generated few-shot learning tasks. We repeat the previous learning steps until we learn a stable representation.

3.2. Step 1: Category Discovery

Clustering is a natural way to gather data that has similar features or close semantic meanings. Recent works on deep clustering that simultaneously conduct representation learning and clustering have been shown to outperform traditional clustering approaches. We take advantage of new unsupervised representation learning methods and choose one of them as our initialization function $\phi(x)$. Given all the unlabeled points $\{x_1, \dots x_n\}$ we can calculate their embeddings as $\{\phi(x_1), \dots \phi(x_n)\}$.

We assume that each discovered category/concept should have one centroid, and all the instances belonging to this category/concept should be closely clustered around the centroid. Thus we use the K-means objective below to look for concepts:

$$\arg\min_{C} \sum_{i=1}^{n} [\arg\min_{z_i} \|Cz_i - \phi(x_i)\|] \tag{1}$$

C is a $d \times k$ matrix where each column corresponds to a centroid, k is the number of centroids, and z_i is a binary assignment vector with length d. By solving this objective function, we will get k clusters of unlabeled instances, which will be used as discovered categories for the next learning stage.

3.3. Step 2: Post-Processing for Representative Data

We wish to use the discovered categories to generate pseudo-labels and create pre-tasks for our next step's self-supervised learning. However, directly using these partitions to generate supervised learning tasks will incur several challenges. For example, 1) the size of each cluster may vary greatly so that large clusters can dominate; 2) some unlabeled instances are hard assigned to a category so that the instances lying on the cluster decision boundaries are prone to be falsely clustered. To counter the previous issues, we propose selecting a group of instances that best resembles the centroids of each discovered category and form balanced clusters.

Given the n unlabeled instances we aim to find a $n \times 1$ binary allocation vector $T = \{t_1, \dots t_n\}$ that selects valuable unlabeled instances. $t_i = 1$ means instance i is selected and $t_i = 0$ means we ignore instance i. Given the learned latent embeddings $\{\phi(x_1), \dots \phi(x_n)\}$ and the centroids for k clusters as $C = \{c_1, \dots c_k\}$, we calculate the cluster probability distribution of instance x_i belonging to cluster j as w_{ij}:

$$w_{ij} = \frac{\exp\{-\|\phi(x_i) - c_j\|^2\}}{\sum_p \exp\{-\|\phi(x_i) - c_p\|^2\}} \tag{2}$$

We look for instances which lie close to their category centroids, to measure the closeness we calculate the entropy of each instance as $Q_i = H(w_i) = -\sum_{j=1}^{k} w_{ij} \log w_{ij}$ and use $n \times 1$ vector Q to represent them. Now we solve for the $n \times 1$ binary vector T. One objective function then is simply find the most valuable instances:

$$\arg\min_{t} \sum_{i} t_i Q_i \tag{3}$$

The aim of the constraints are two-folded: to balance the size of each discovered category whilst also keep the most valuable instances. Our first basic constraint includes the total number of selected instances to be $m < n$ so that uncertain unlabeled instances will be removed. Our next constraint requires that each cluster should have a reasonable number of instances so that some large clusters won't dominate. Thus we have

$$s.t. \sum_{i} t_i = m \tag{4}$$

$$s.t. \sum_{i} t_i z_{ij} \geq L \quad \forall j \tag{5}$$

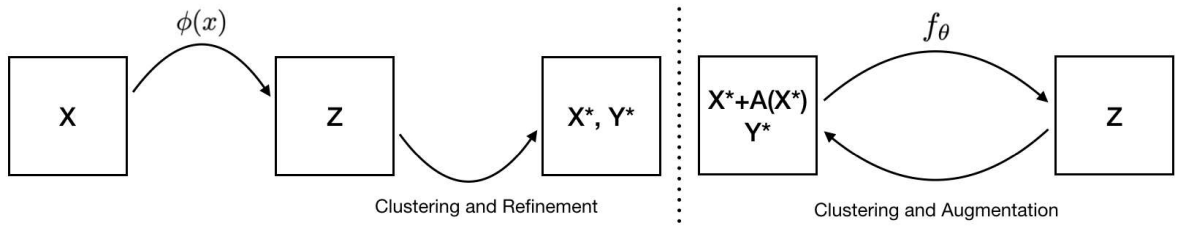


Fig. 2: The whole learning process of our framework (left hand side shows model initialization and right hand side shows iterative learning). Left: The unlabeled points set X is first encoded to Z via existing unsupervised representation learning function $\phi(x)$. After that we conduct clustering on Z and post-process the clustering assignments to get pseudo-labeled data set X^*, Y^*. Right: Given the pseudo-labeled data and corresponding composed augmentations we train ProtoNet f_θ via episodic learning and get learned embeddings Z, clustering on Z and re-label X^* and generate new augmented instances.

$$s.t. \sum_i t_i z_{ij} \leq U \quad \forall j \tag{6}$$

Note that U and L serves as the upper bound and lower bound of each cluster's size. Details of how we set up the constraint hyper-parameters will be described in experimental section 4.

3.4. Virtual Instance Generation

Given the m selected instances which belong to k categories, the instances belonging to the same category tend to be similar to each other and are not diverse enough. Training a supervised few-shot learning model over these instances is prone to overfitting due to the low-diversity and cannot generalize well. Moreover, the learned embedding will largely resemble the embedding which we initialized.

Increasing the number of training examples by data augmentation is a popular approach in supervised learning and semi-supervised learning to improve generalization. Before splitting the selected instances into support sets and query sets for meta training, we apply traditional image augmentation approaches to augment the m chosen instances. We pick the standard image augmentation strategies to enrich the diversity while maintaining the class membership of the augmented instances. To be specific, we compose random sized crop, image jitter, and random horizontal flip as our augmentation strategy.

3.5. Iterative Training

In this section, we introduce how we train our representation learning framework thoroughly. Our work aims to learn an embedding function f_θ that naturally splits the discovered categories of all the unlabeled instances and form tight and diverse clusters. To achieve this goal, we propose to use a prototype-based network as f_θ and train it via few-shot classification tasks based on discovered labeled instances.

We first introduce notations and recap the few-shot learning setting. In few-shot learning tasks, we divide the dataset into three disjoint sets which are meta-training set $\mathcal{S}^{tr}$, meta-validation set $\mathcal{S}^{val}$ and meta-testing set $\mathcal{S}^{test}$. $\mathcal{S}^{tr}$ contains all the base classes, $\mathcal{S}^{val}$ and $\mathcal{S}^{test}$ contains disjoint novel classes. The few-shot learning models are trained in an episodic paradigm [26] that each episode contains one support set $\mathcal{D}^S$ and one query set $\mathcal{D}^Q$. Here, the support set $\mathcal{D}^S$ serves as the labeled training set on which the model is trained to minimize the loss of its predictions for query set $\mathcal{D}^Q$.

Secondly, we introduce how to train our few-shot classification learner. Denote the augmented representative training set as $\mathcal{S}^{tr}$, now we randomly split $\mathcal{S}^{tr}$ into support set $\mathcal{D}^S$ and query set $\mathcal{D}^Q$. We choose Prototypical Networks as our learner to classify different discovered classes. Prototypical Networks compute an M dimensional representation $c_k \in \mathcal{R}^M$, or prototype, of each class through an embedding function f_θ with learnable parameters θ. Each prototype c_k is the mean vector of the embedded support points belonging to class k:

$$c_k = \frac{1}{\mathcal{D}_k^S} \sum_{(x_i, y_i) \in \mathcal{D}_k^S} f_\theta(x_i) \tag{7}$$

Given a Euclidean distance function d, Prototypical Networks produce a distribution over classes for a query point x based on a softmax over Euclidean distances to the prototypes in the embedding space:

$$p_\theta(y = k|x) = \frac{\exp\left(-d(f_\theta(x), c_k)\right)}{\sum_{k'} \exp\left(-d(f_\theta(x), c_{k'})\right)} \tag{8}$$

Learning proceeds by minimizing the negative log probability of the true class k via SGD: $J(\theta) = -\log p_\theta(y = k|x)$. Training episodes are generated from meta-training set $\mathcal{S}^{tr}$.

Now we connect all the technical components and visualize the training process in Figure 2. Note the embedding function f_θ is initialized in each iteration to learn a new embedding of unlabeled instances that can discriminate against the previously discovered classes and augmented instances. We repeat the whole training process for f_θ until the clustering results remain stable. We empirically observe that the clustering results of the learned embeddings Z tend to converge after a few rounds of iterative learning loop. Moreover, the iterative learning consistently improves both the clustering results and few-shot classification performance which can be seen in the experimental section 4.

4. Experiments

The proposed framework is evaluated in two learning settings, which are image clustering and unsupervised few-shot classification. Our experimental results aim to answer the following questions:

- How does the proposed framework perform on unsupervised few-shot classification tasks compared to recent baselines?
- Compared to the original features we used to discover the categories, can our proposed framework provide a further enhancement in terms of clustering results?
- How does each step within the framework contributes to the final results?

4.1. Datasets

Omniglot [20] is a dataset of handwritten characters frequently used to compare few-shot learning algorithms. It comprises 1623 characters from 50 different alphabets. Every character in Omniglot has 20 different instances. The *mini*ImageNetis a collection of ImageNet for few-shot image recognition. It is composed of 100 classes randomly selected from ImageNet, with each class containing 600 examples.

We also experiment on some traditional image clustering data sets to study whether our framework can improve further on the image clustering tasks: 1) MNIST [18], which consists of 70000 handwritten digits of 28-by-28 pixel size. The digits are centered and size-normalized in our experiments; 2) FASHION-MNIST [27]: a Zalando's article images-consisting of a training set of 60000 examples and a test set of 10000 examples. Each example is a 28-by-28 grayscale image, associated with a label from 10 classes; 3) USPS, which contains 9298 handwritten digit images with the size of 16 by 16 pixels.

4.2. Implementation Details

We have explored different unsupervised representation learning algorithms to initialize our pipeline. For the Omniglot data set, we choose adversarially constrained autoencoder interpolation (ACAI) [3], which is a convolutional autoencoder regularized with a term encouraging meaningful interpolations in the latent space. For *mini*ImageNet, we leverage the recent large-scale unsupervised visual representation work (DeepCluster [5]). For traditional deep clustering data sets such as MNIST, Fashion-MNIST, and USPS, IDEC [22] is an auto-encoder based deep clustering algorithm that we adopted for testing whether our proposed pipeline improves upon traditional deep clustering approaches.

For k-means clustering, we initialize the clustering methods with 100 random trials and use the best trial for initialization; the number of clusters for unsupervised few-shot learning is set to 100 empirically for both Omniglot and *mini*ImageNet.

To select the valuable instances from the clustering results, we set m to be 80% of the original unlabeled data size. The rationale for selecting a relatively large portion is because we only observe a small group of points lies between different clusters across different data sets. The size of selected instances is not sensitive within the range between 70% to 90% of all unlabeled data. Denote the average number of instances per cluster as $\mathcal{S}$; we set the upper bound U and lower bound L as $2\mathcal{S}$ and $0.5\mathcal{S}$. We solve this integer programming problem by linear programming relaxation and then round the results for faster computation on complex datasets such as *mini*ImageNet and Omniglot.

For fair comparison with other few-shot classification methods, we adopt a widely-used CNN architecture [26, 10, 21, 24] as the feature embedding function f_θ. Following [21], we adopt the episodic training procedure, i.e., we sample a set of N-way k-shot training tasks to mimic the N-way k-shot test problems. In all settings, the query set's size of each novel class is set as 15, and the performance is averaged over 1000 randomly generated episodes from the test set. All our models are trained with Adam optimizer and an initial learning rate of 10^{-3}. We trained 240000 tasks for *mini*ImageNet and 100000 tasks for other data sets. We have included [15, 16, 2] as our unsupervised few-shot learning baselines and also [14] as deep clustering baseline.

Table 1: Comparison to prior works on *Omniglot*. We report the few-shot classification mean accuracies

	Omniglot			
Model	(5, 1)	(5, 5)	(20, 1)	(20, 5)
CACTUs-MAML	68.84%	87.78%	48.09%	73.36%
CACTUs-ProtoNet	68.12%	83.58%	47.75%	66.27%
UMTRA	77.80%	92.74%	62.20%	77.50%
AAL-MAML++	88.40%	97.96%	70.21%	88.32%
AAL-ProtoNet	84.66%	89.14%	68.79%	74.28%
Ours	**94.09%**	**98.43%**	**87.56%**	**96.15%**
Supervised-MAML	98.70%	98.90%	95.80%	98.90%
Supervised-ProtoNet	98.80%	99.70%	96.00%	98.90%

Table 2: Comparison to prior works on *mini*ImageNet. We report the few-shot classification mean accuracies. AAL [2] didn't report the results for 20 shot and 50 shot setting so we leave them as blank entries

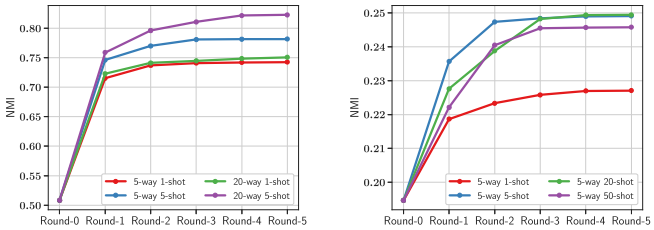
	***mini*ImageNet**			
Model	(5, 1)	(5, 5)	(5, 20)	(5, 50)
CACTUs-MAML	39.90%	53.97%	63.84%	69.64%
CACTUs-ProtoNet	39.18%	53.36%	61.54%	63.55%
UMTRA	39.93%	50.73%	61.11%	63.55%
AAL-MAML++	33.30%	49.18%	–	–
AAL-ProtoNet	37.67%	40.29%	–	–
Ours	**41.16%**	**57.03%**	**68.85%**	**70.64%**
Supervised-MAML	46.81%	64.13%	71.03%	75.54%
Supervised-ProtoNet	50.16%	65.56%	70.05%	72.04%

4.3. Unsupervised Few-shot Classification on Omniglot

We compare our model's unsupervised few-shot classification mean accuracy with recent baselines on the Omniglot dataset in Table 1. We can find our approach significantly improves upon the previous baselines in both 5 way few-shot classification and 20 way few-shot classification settings. This shows that our framework consistently improves the learned embeddings under multiple way classification tasks. We also report the representative supervised few-shot classification results and find out our approach largely minimizes the gap between unsupervised and supervised few-shot classification in Omniglot, with 1.27% difference in 5-way 5-shot learning and 1.75% difference in 20-way 5-shot setting.

4.4. Unsupervised Few-shot Classification on mini*ImageNet*

We further evaluate the unsupervised few-shot classification on *mini*ImageNet data set which is much more challenging than Omniglot. Results are reported in Table 2. As shown, our approach still achieves the best results for 5 way classification with a different number of shots. This suggests that our framework is general across different data sets and behaves consistently under different few-shot classification settings. Moreover, comparing our results with the supervised baselines, we can find the gap is larger compared to the gap in Omniglot



(a) Omniglot (b) *mini*ImageNet

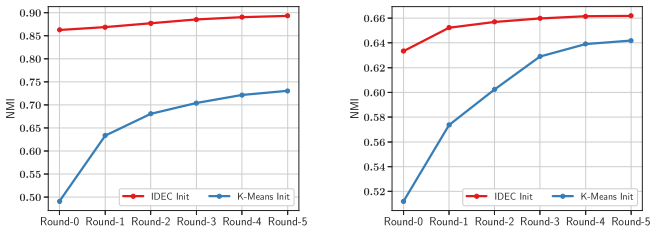
Fig. 3: We evaluate the clustering NMI for our learned embeddings on Omniglot and *mini*ImageNet for each training iteration.

dataset. This is acceptable as images within *mini*ImageNet vary and are harder to recognize.

4.5. *How our framework improves upon the initial embeddings in terms of clustering performance.*

Our proposed framework takes advantage of the initialization embeddings learned from recent unsupervised learning methods. To show how our model improves upon them, we test the clustering performance of both initial embeddings and learned embeddings and plot the results in Figure 3. Note we have used ACAI's [3] embedding algorithm to initialize Omniglot data and DeepCluster's [5] embedding approach to initialize the *mini*ImageNet data. We adopt standard metrics for evaluating clustering performance which measures how close the clustering found is to the ground truth result. Specifically, we employ the normalized mutual information (NMI) [23, 29].

As can be seen from part (a) of Figure 3, we train our framework for five iterations with 5-way and 20-way few-shot classification tasks. The embeddings' clustering performance consistently improves as the number of rounds increasing and tend to converge after five iterations. The first iteration of learning provides the largest improvement, which shows that our framework further improved the initialized embedding. Moreover, we discover that the embeddings learned with five shots are better for clustering than the embeddings learned with only one shot. We have observed the similar results in *mini*ImageNet experiment that shown in part (b) of Figure 3. Our framework can provide consistent improvements across different data sets with different initializations.



(a) Clustering results on MNIST (b) Clustering results on Fashion

Fig. 4: We evaluate the clustering NMI for our learned embeddings on MNIST, FASHION-MNIST for each training iteration.

Moreover, we have tested our approach on traditional clustering datasets such as MNIST, FASHION-MNIST, and USPS. We apply two approaches to initialize all these three datasets. We first test on whether our framework can improve upon the existing deep clustering approach (IDEC). We also use a naive K-Means clustering on the raw images to test whether our framework is sensitive to the initial partition results. We fix the few-shot classification tasks as 5-way 5 shot classification problem for our training process. The clustering results for these three data sets are shown in Figure 4.

Firstly, we can find our framework always improves the clustering performance no matter which initialization strategy we use. Secondly, we observe that even if we start our framework with K-Means partitions over the raw images, our framework can still largely improve the clustering NMI for MNIST and Fashion-MNIST. Thirdly we find that our approach can achieve high NMI over USPS data set by using K-Means initialization. We hypothesis this could happen when the initial NMI is relatively high. Overall speaking, our proposed approach is general for learning a representation that is suitable for clustering; the initialization approach serves as a performance lower-bound for our framework.

4.6. *Benefits of data augmentations*

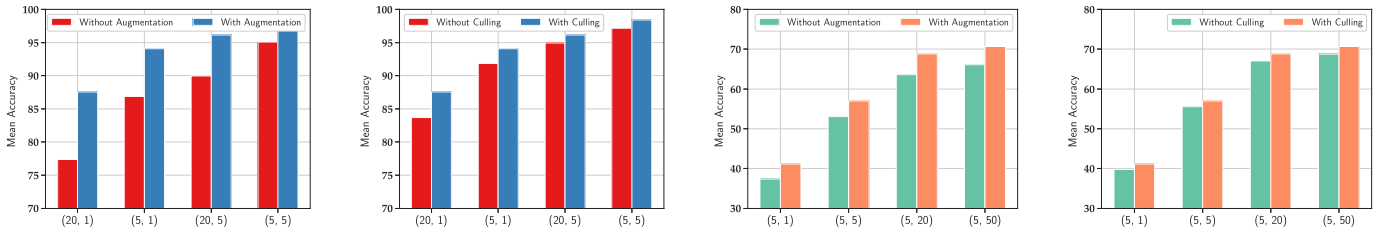
Figure 5 (a) and (c) shows the impact of adding composed data augmentations when models are trained for different few-shot classification settings. In the Omniglot data set, we find that when we are doing 1-shot learning, data augmentation contributes significantly to the final performance. With more labeled points, the gaps between using or not use augmentation decrease. However, we still observe a large improvement in terms of classification accuracy. In *mini*ImageNet, we find data augmentation consistently improves the final performance with a different number of shots. Overall speaking, the composition of data augmentations plays a critical role in unsupervised few-shot classification.

4.7. *Benefits of category post-processing*

Figure 5 (b) and (d) shows that the culling of unlabeled instances via our post-processing objective improves the performance on both data sets. This suggests that directly using the naive K-Means clustering results to design pre-task may not be appropriate, picking instances which are representative for each discovered class is beneficial for creating useful few-shot classification tasks.

5. Conclusions

We have presented a self-supervised learning framework to learn representation on images for better downstream tasks such as unsupervised few-shot learning and clustering, where no human annotation is provided. We first discover categories from initialized embeddings and then propose integer linear programming to select valuable instances that form balance and representative concepts. Moreover, we augment the selected valuable instances to create pre-tasks and iterative train



(a) Omniglot (b) Omniglot (c) *mini*ImageNet (d) *mini*ImageNet

Fig. 5: Plots of ablation study. Note plot (a) and (c) test on how much our framework benefit from composition of data augmentations and plot (b) and (d) test on how much our framework benefit from culling the unlabeled instances for better pre-task design

our network to fine-tune gradually, which resembles the human learning process, such as discovering concepts and learning to separate them. We show by experiments that the proposed framework has further improved the embedding initialized by existing representation learning approaches in terms of downstream tasks. For unsupervised few-shot classification, our proposed approach achieved state-of-the-art performance on the Omniglot and *mini*ImageNet benchmarks. In the image clustering setting, we also find our approach further improves recent deep clustering approaches under traditional clustering data sets. The ablation study demonstrates the importance of each technical component in our framework.

References

[1] A. Antoniou, H. Edwards, and A. Storkey. How to train your maml. *arXiv preprint arXiv:1810.09502*, 2018.

[2] A. Antoniou and A. Storkey. Assume, augment and learn: Unsupervised few-shot meta-learning via random labels and data augmentation. *arXiv preprint arXiv:1902.09884*, 2019.

[3] D. Berthelot, C. Raffel, A. Roy, and I. Goodfellow. Understanding and improving interpolation in autoencoders via an adversarial regularizer. *arXiv preprint arXiv:1807.07543*, 2018.

[4] H. Bourlard and Y. Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 59(4-5):291–294, 1988.

[5] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *European conference on computer vision*, pages 132–149, 2018.

[6] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[7] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.

[8] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[9] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in neural information processing systems*, pages 766–774, 2014.

[10] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. JMLR. org, 2017.

[11] K. Ghasedi Dizaji, A. Herandi, C. Deng, W. Cai, and H. Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *Proceedings of the IEEE international conference on computer vision*, pages 5736–5745, 2017.

[12] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[14] X. Guo, L. Gao, X. Liu, and J. Yin. Improved deep embedded clustering with local structure preservation. In *IJCAI*, pages 1753–1759, 2017.

[15] K. Hsu, S. Levine, and C. Finn. Unsupervised learning via meta-learning. *arXiv preprint arXiv:1810.02334*, 2018.

[16] S. Khodadadeh, L. Boloni, and M. Shah. Unsupervised meta-learning for few-shot image classification. In *Advances in neural information processing systems*, pages 10132–10142, 2019.

[17] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[19] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016.

[20] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850, 2016.

[21] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017.

[22] K. Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in neural information processing systems*, pages 1857–1865, 2016.

[23] A. Strehl, J. Ghosh, and R. Mooney. Impact of similarity measures on web-page clustering. In *Workshop on AAAI*, volume 58, page 64, 2000.

[24] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, pages 1199–1208, 2018.

[25] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(Dec):3371–3408, 2010.

[26] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

[27] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[28] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487, 2016.

[29] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *SIGIR*, pages 267–273, 2003.

[30] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *International conference on machine learning*, pages 3861–3870. JMLR. org, 2017.

[31] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.