Mitigating Embedding and Class Assignment Mismatch in Unsupervised Image Classification

Sungwon Han[1], Sungwon Park[1], Sungkyu Park[1], Sundong Kim[2], and Meeyoung Cha[1,2(✉)]

[1] Korea Advanced Institute of Science and Technology, Daejeon, South Korea
{lion4151,psw0416,shaun.park}@kaist.ac.kr
[2] Data Science Group, Institute for Basic Science, Daejeon, South Korea
{sundong,mcha}@ibs.re.kr

Abstract. Unsupervised image classification is a challenging computer vision task. Deep learning-based algorithms have achieved superb results, where the latest approach adopts unified losses from embedding and class assignment processes. Since these processes inherently have different goals, jointly optimizing them may lead to a suboptimal solution. To address this limitation, we propose a novel two-stage algorithm in which an embedding module for pretraining precedes a refining module that concurrently performs embedding and class assignment. Our model outperforms SOTA when tested with multiple datasets, by substantially high accuracy of 81.0% for the CIFAR-10 dataset (i.e., increased by 19.3 percent points), 35.3% accuracy for CIFAR-100-20 (9.6 pp) and 66.5% accuracy for STL-10 (6.9 pp) in unsupervised tasks.

1 Introduction

Deep learning-based algorithms have led to remarkable advances in various computer vision tasks thanks to their representative power [10,21,24]. However, these models often require active supervision from costly, high-quality labels. In contrast, unsupervised learning reduces labeling costs and, therefore, is more scalable [4,11,38].

Unsupervised image classification aims to determine the membership of each data point as one of the predefined class labels without utilizing any label information [18,39]. Since images are high dimensional objects, most existing methods focus on reducing dimensionality while discovering appropriate decision boundaries. The task of projecting high-dimensional data to lower dimensions is called embedding, and the task of identifying boundaries of dense groupings is called class assignment. Two methods are popularly used: 1) *a sequential method*, which separately trains to embed and assign samples to classes, and 2) *a joint method*, which simultaneously trains the samples in an end-to-end fashion.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-58586-0_45) contains supplementary material, which is available to authorized users.

A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12369, pp. 768–784, 2020.
https://doi.org/10.1007/978-3-030-58586-0_45

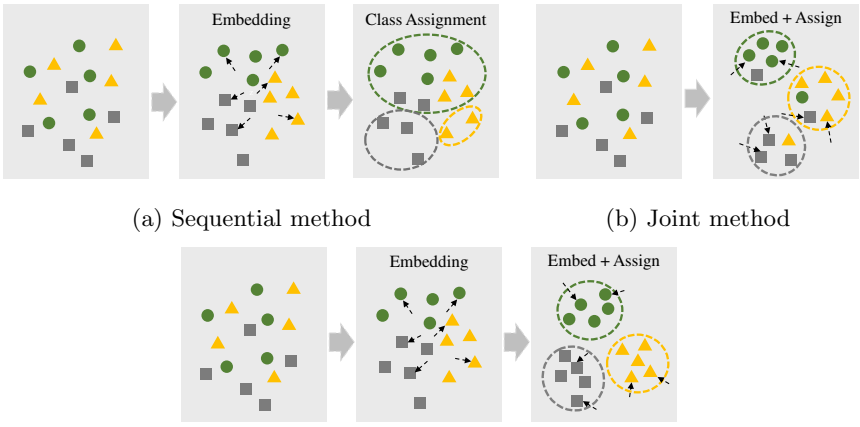


Fig. 1. Unsupervised image classification methods. (a) The sequential method embeds and assigns data points into classes separately, whereas (b) the joint method embeds and groups into classes together. (c) The proposed method first performs embedding learning as a pretraining step to find good initialization, then it jointly optimizes the embedding and class assignment processes. Our two-stage design introduces unique losses upon the pretraining step.

The sequential method, depicted in Fig. 1(a), utilizes embedding learning to represent visual similarity as a distance in the feature space [1,31]. Clustering algorithms such as k-means [25] and DBSCAN [8] are then applied. The embedding stage of this method solely reduces data dimensions, without knowledge of the immediately following class assignment, and hence may find representations that allow little separation between potential clusters.

The joint method, depicted in Fig. 1(b), simultaneously optimizes embedding and class assignment [4,39,41]. Some studies introduce the concept of clustering loss (e.g., k-means loss [40]) that is added to the embedding loss to yield enough separation between decision boundaries. Information Maximizing Self-Augmented Training (IMSAT) [15] and Invariant Information Clustering (IIC) [18] are new methods that show remarkable performance gain over conventional sequential methods. These models effectively extract invariant features against data augmentation by maximizing the mutual information.

Nevertheless, all of these models bear a common drawback that the goals for embedding and class assignment are inherently different. The former encodes high dimensional data points, whereas the latter identifies a proper class label for data points. Hence, jointly minimizing losses for these tasks may lead to what we identify as a "mismatched" result. A good example of a mismatch is when clusters are identified due to trivial traits such as colors rather than object shapes [4,18]. The gradient loss on the class assignment process in the early stage of training affects how images are grouped (i.e., by colors). To overcome

this limitation, some models propose using Sobel-filtered images [4,18] (i.e., black and white versions) and avoiding trivial clustering, yet at the cost of losing crucial information (i.e., colors).

This paper presents an entirely different, two-stage approach. Stage 1 is embedding learning that extracts data representation without any guidance from human-annotated labels. The goal here is to gather similar data points in the embedding space and find a well-organized initialization. Stage 2 is class assignment and elaboratively minimizes two kinds of losses: (1) class assignment loss that considers assignment of both the original and augmented images and (2) embedding loss that refines embedding and prevents the model from losing its representation power. Our design, depicted in Fig. 1(c), outperforms existing baselines by a large margin. The main highlights are as follows:

- The two-stage process starts with embedding learning as a pretraining step, which produces a good initialization. The second aims to assign a class for each data point by refining its pretrained embedding. Our model successfully optimizes two objectives without falling into the mismatched state.
- Our method outperforms existing baselines substantially. With the CIFAR-10 dataset, we achieve an accuracy of 81.0%, whereas the best performing alternative reaches 61.7%.
- Extensive experiments and ablation studies confirm that both stages are critical to the performance gain. Comparison with the current state-of-the-art (SOTA) shows that our approach's most considerable benefit comes from the embedding learning initialization that gathers similar images nearby in the low-dimensional space.
- Our model can be adopted as a pretraining step for subsequent semi-supervised tasks. We discuss these implications in the experiments section.

Implementation details and codes are made available via GitHub[1] and the supplementary material.

2 Related Work

2.1 Unsupervised Embedding

Advances in unsupervised embedding can be discussed from three aspects: self-supervised learning, sample specificity learning, and generative models. Among them, self-supervised learning relies on auxiliary supervision. For instance, one may extract latent representations from images by expanding or rotating images [9] or solving a jigsaw puzzle made of input images [27]. Next, sample specificity learning considers every instance in the data as a single individual class [3,38]. The idea relies on the observation that deep learning can detect similarities between classes via supervised learning. By separating all data instances into the L2-normalized embedding space, this method gathers similar samples automatically due to its confined space. For example, Anchor Neighborhood

[1] Codes released at https://github.com/dscig/TwoStageUC.

Discovery (AND) [16] progressively discovers sample anchored neighborhoods to learn the underlying class decision boundaries iteratively. More recently, Super-AND [13] unifies some of the key techniques upon the AND model by maintaining invariant knowledge against small deformations [42], and by newly employing entropy-based loss. Finally, generative models learn to reconstruct the hidden data distribution itself without any labels. Thus, the model can generate new samples that likely belong to the input dataset [11,19]. Some research attempts have been made to use the generative model for deep embedding learning [29].

2.2 Unsupervised Classification

Class assignments can sequentially follow embedding or be optimized jointly. An example sequential setting is to apply principal component analysis before performing k-means clustering to relax the curse of dimensionality [7]. Another example is to use matrix factorization, which allows the derived matrix to learn a low-dimensional data representation before clustering [34]. For the face recognition task, one study embedded facial similarity features into Euclidean space and then clustered images based on the derived embedding [31]. Another approach is to utilize autoencoders. For example, one study stacked autoencoders in the deep networks to handle noisy input data [35]. Another study used a Boolean autoencoder as a form of non-linear representation [1]. Both studies showed a considerable improvement in classification tasks at that time.

The joint method is next, which considers embedding and class assignment simultaneously. Deep neural networks are used for the joint optimization of dimensionality reduction and clustering [40]. One study proposed the concept of deep embedded clustering (DEC), which learns latent representations and cluster assignments at the same time [39]. For the image clustering task, another study jointly updated clusters and their latent representations, derived by CNN via a recurrent process [41]. The deep adaptive clustering (DAC) model [5] computes the cosine similarities between pairs of hidden features on images via CNN. This model tackles the variant problem of the binary pairwise classification task, where the goal is to determine whether an image pair belongs to the same cluster. DeepCluster repeatedly learns and updates the features of CNN and the results of k-means clustering [4].

Most recently, a model called IIC (Invariant Information Clustering) has shown superior performance [18]. This model maximizes the mutual information of paired images. Because data augmentation does not deform the critical features of input images, this process can learn invariant features that persist in both the original and augmented data. The framework of the IIC is novel and straightforward. This algorithm is robust against any degeneracy in which one or more clusters have no allocated samples, or a single cluster dominates the others. Because of the entropy term in the mutual information, the loss cannot be minimized if a specific cluster dominates the others.

3 Model

(a) Stage 1: Unsupervised deep embedding finds good initialization.

(b) Stage 2: Unsupervised class assignment with refining pretrained embeddings identifies consistent and dense groupings.

Fig. 2. Model illustration. The encoder projects input images to a lower dimension embedding sphere via deep embedding. The encoder is then trained to gather samples with similar visual contents nearby and separate them otherwise. Next, a multi-head normalized fully-connected layer classifies images by jointly optimizing the class assignment and embedding losses.

Problem Definition. Consider the number of underlying classes n_c and a set of n images $\mathcal{I} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$. The objective of the unsupervised classification task is to learn a mapping f_θ that classifies images into the pre-defined n_c clusters without the use of any labels. Each stage of our method is described below.

3.1 Stage 1: Unsupervised Deep Embedding

The goal of this stage is to extract visually essential features without the use of any labels. An ideal embedding scenario should discriminate images of different classes and place them far apart in the embedding space while gathering similar images near each other. Since the model does not know beforehand which images are in the same class, the unsupervised embedding task is inherently challenging. Several advances have been made in this domain, including self-supervised learning, sample specificity learning, and generative models. Among them, we adopt Super-AND [13] to initialize the encoder. This model achieves high performance for unsupervised deep embedding tasks.

Super-AND extends AND [16], a powerful sample specificity learning method, by employing (1) data augmentation and (2) entropy-based loss. Total three losses are used for training: AND-loss (L_{and}), UE-loss (unification entropy loss, L_{ue}), and AUG-loss (augmentation loss, L_{aug}). Following the original AND algorithm, Super-AND considers every data occurrence as an individual class and separates the data in the L2-normalized embedding space. Then, the model groups the data points into small clusters by discovering the nearest neighborhood pairs, which is depicted in Fig. 2(a). The model runs iteratively to increase the number of identified subclasses by focusing on local clusters in each round. The subclass information is used for a self-supervised learning task to distinguish every local cluster. AND-loss considers each discovered neighborhood pair

or remaining data instance as a single class to separate. This cross-entropy loss is written as:

$$L_{and} = -\sum_{i \in \mathcal{N}} \log\Big(\sum_{j \in \tilde{\mathcal{N}}(\mathbf{x}_i) \cup \{i\}} \mathbf{p}_i^j\Big) - \sum_{i \in \mathcal{N}^c} \log \mathbf{p}_i^i, \tag{1}$$

where $\mathcal{N}$ is the selected part of the neighborhood pair sets with its complement $\mathcal{N}^c$, $\tilde{\mathcal{N}}(\mathbf{x}_i)$ is the neighboring image i, and $\mathbf{p}_i^j$ represents the probability of i-th image being identified as j-th class.

UE-loss intensifies the concentration effect of AND-loss. UE-loss is defined as the entropy of the probability vector $\tilde{\mathbf{p}}_i$ except for instance itself. $\tilde{\mathbf{p}}_i$, which is computed from the softmax function, represents the similarity between instance i and the others in a probabilistic manner. The superscript j in $\tilde{\mathbf{p}}_i^j$ denotes the j-th component value of a given vector $\tilde{\mathbf{p}}_i$. By excluding the class of one's own, minimizing UE-loss makes nearby data occurrences attract each other—a concept that is contrary to the sample specificity learning. Jointly employing UE-loss with the AND-loss will enforce the overall neighborhoods to be separated while keeping similar neighborhoods to be placed closely. The UE-loss is calculated as follows:

$$L_{ue} = -\sum_{i} \sum_{j \neq i} \tilde{\mathbf{p}}_i^j \log \tilde{\mathbf{p}}_i^j. \tag{2}$$

Lastly, AUG-loss is defined to learn invariant image features against data augmentation. Since augmentation does not deform the underlying data characteristics, invariant features learned from the augmented data will still contain the class-related information. Model regards every augmentation instance as a positive sample and reduces the discrepancy between the original and augmented pair in embedding space. In Eq. 3, $\bar{\mathbf{p}}_i^j$ denotes the probability of wrong identification to class-j for the original i-th image, when $\bar{\mathbf{p}}_i^i$ describes that of correct identification to class-i for an augmented version of i-th image. Then, AUG-loss is defined as a cross entropy to minimize misclassification over batch instances.

$$L_{aug} = -\sum_{i} \sum_{j \neq i} \log(1 - \bar{\mathbf{p}}_i^j) - \sum_{i} \log \bar{\mathbf{p}}_i^i \tag{3}$$

The three losses are combined by weight manipulation on the UE-loss (Eq. 4). Weights for UE-loss $w(t)$ are initialized from 0 and increased gradually. The Super-AND model is trained by optimizing the total loss, and finally, the trained encoder becomes an initial point for the classification model in the next stage.

$$L_{stage1} = L_{and} + w(t) \times L_{ue} + L_{aug} \tag{4}$$

3.2 Stage 2: Unsupervised Class Assignment with Refining Pretrained Embeddings

The goal of this stage is to identify appropriate boundaries among classes. Unlike the first stage, an ideal class assignment requires not only ideal embedding, but

also requires dense grouping to form decision boundaries with sufficient margins. This stage handles the given requirements by refining the initialized embeddings from the previous stage. Here, two kinds of losses are defined and used: class assignment loss and consistency preserving loss.

Mutual Information-Based Class Assignment. Mutual information quantifies mutual dependencies of two random variables [28], and measures how much two variables share the same kind of information. For example, if two variables are highly correlated or come from the same underlying distribution, their mutual information is significant, i.e., higher than zero. We can regard mutual information as the KL-divergence between the joint distribution and the product of its marginal distribution as follows:

$$I(x, y) = D_{KL}(p(x, y)||p(x)p(y)) \tag{5}$$

$$= \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \tag{6}$$

$$= H(x) - H(x|y). \tag{7}$$

The IIC (Invariant Information Clustering) model has been proposed for unsupervised classification to maximize the mutual information between the samples and the augmented set of samples [18]. This unique method trains the classifier with invariant features from data augmentation as follows: Suppose we have an image set $\mathbf{x}$ and a corresponding augmented image set $g(\mathbf{x})$ with a function g that geometrically transforms input images. The mapping f_θ classifies the images and generates the probability vector $y = f_\theta(\mathbf{x}), \hat{y} = f_\theta(g(\mathbf{x}))$ of all classes. Then, the model tries to find an optimal f_θ that maximizes the following terms:

$$\max_\theta I(y, \hat{y}) = \max_\theta (H(y) - H(y|\hat{y})). \tag{8}$$

By maximizing mutual information, we can prevent the clustering degeneracy, i.e., some clusters dominate the others, or there are no instances in a certain cluster. Since mutual information can be decomposed into two terms, maximizing $I(y, \hat{y})$ leads to maximizing $H(y)$ and minimizing $H(y|\hat{y})$. More specifically, $H(y)$ is maximized when every data sample is evenly assigned to every cluster; we can avoid degenerated solutions after optimization while consistent clusters are made with minimized $H(y|\hat{y})$.

We denote the joint probability distribution of y and $\hat{y}$ over the batch $\mathcal{B}$ to matrix $\mathbf{P}$ (i.e., $\mathbf{P} = \frac{1}{n} \sum_{i \in \mathcal{B}} f_\theta(x_i) \cdot f_\theta(g(x_i))^T$), where n is the size of batch $\mathcal{B}$). Using this matrix, we can easily define the objective function targeted to maximize the mutual information. By changing its sign, we define a class assignment loss L_{assign} (Eq. 9), where c and c' denote the class indices of the original and its augmented version, respectively. In the equation, $\mathbf{P}_{cc'}$ denotes the element at c-th row and c'-th column, where $\mathbf{P}_c$ and $\mathbf{P}_{c'}$ are the marginals over the rows and columns of the matrix.

$$L_{assign} = -\sum_c \sum_{c'} \mathbf{P}_{cc'} \cdot \log \frac{\mathbf{P}_{cc'}}{\mathbf{P}_{c'} \cdot \mathbf{P}_c} \tag{9}$$

Consistency Preserving on Embedding. We added an extra loss term, consistency preserving loss L_{cp}, to penalize any mismatch between original and augmented images. If the model only focuses on assigning class, during the process of dense grouping, embedding results can be easily modified just to match the number of final classes. By concurrently minimizing L_{cp}, our model can refine its embedding and avoid hasty optimization.

Assume that the $\mathbf{v}_i$ is the representation of an image $\mathbf{x}_i$ produced by the encoder in Stage 1. This $\mathbf{v}_i$ is projected into the normalized sphere, where the dot product can calculate the similarity of instances. Since augmented images have the same contents as the original ones, the similarity distance between them should be closer than other instances. We calculate $\hat{\mathbf{p}}_i^j$ $(i \neq j)$, the probability of given instance i classified as j-th instance, and $\hat{\mathbf{p}}_i^i$, the probability of being classified as its own i-th augmented instance (Eq. 10). The temperature value, τ, ensures that the label entropy distribution remains low [14]. Consistency preserving loss L_{cp} finally minimizes any misclassified cases over the batches (Eq. 11).

$$\hat{\mathbf{p}}_i^j = \frac{\exp(\mathbf{v}_j^\top \mathbf{v}_i/\tau)}{\sum_{k=1}^n \exp(\mathbf{v}_k^\top \mathbf{v}_i/\tau)}, \quad \hat{\mathbf{p}}_i^i = \frac{\exp(\mathbf{v}_i^\top \hat{\mathbf{v}}_i/\tau)}{\sum_{k=1}^n \exp(\mathbf{v}_k^\top \hat{\mathbf{v}}_i/\tau)} \tag{10}$$

$$L_{cp} = -\sum_i \sum_{j \neq i} \log(1 - \hat{\mathbf{p}}_i^j) - \sum_i \log \hat{\mathbf{p}}_i^i \tag{11}$$

The total unsupervised classification loss for the second stage is defined as follows (Eq. 12). λ is a hyper-parameter used for manipulating the weight of the consistency preserving loss term.

$$L_{stage2} = L_{assign} + \lambda \cdot L_{cp} \tag{12}$$

Normalized Fully-Connected Classifier. The commonly used fully-connected layer computes a weighted sum over the input. However, for feature vectors projected to a unit sphere space, as in the case of our encoder model, the conventional fully-connected layer with a bias term does not fit well. This is because the scale of the weights and bias obtained during training for each class can become pronounced to make drastic decisions after the softmax function. These variable sizes of weight vectors can lead to internal covariate shifts [17,30].

We introduce the normalized fully-connected layer (Norm-FC) without any bias term for classification. Each weight in Norm-FC becomes a prototype vector of each class, and images are classified by evaluating similarity in comparison with the class prototypes. Compared to the original classifier with the softmax function, Norm-FC is the layer whose weights $\mathbf{w}$ are L2-normalized (Eq. 13). To improve prediction confidence (i.e., reduce entropy in the distribution of model prediction), we additionally divide temperature $\tau_c < 1$ before the softmax function, as in Eq. 10. The temperature τ_c is critical for adjusting the concentration of feature vectors projected in unit sphere [36,38]. In the following equation, y_i^j is the j-th element of the classification probability vector for image $\mathbf{x}_i$, and $\mathbf{w}_j$

is the weight vector for class j in the fully-connected layer. Encoder with five Norm-FC classification heads is used for the second stage classifier.

$$y_i^j = \frac{\exp(\frac{\mathbf{w}_j}{||\mathbf{w}_j||} \cdot \mathbf{v}_i/\tau_c)}{\sum_k \exp(\frac{\mathbf{w}_k}{||\mathbf{w}_k||} \cdot \mathbf{v}_i/\tau_c)} \tag{13}$$

4 Experiments

We conducted extensive experiments and compared the model's performance against other baselines. We also examined how each of the two stages contributes to the performance gain. Last, we evaluated the performance after training with a dataset with scarce labels to analyze the relevance of the model to semi-supervised models. Implementation details such as model architecture and the reasoning behind hyper-parameter values such as τ_c, λ, and $w(t)$, can be found in supplementary material.

4.1 Image Classification Task

Datasets. Three datasets are used. (1) *CIFAR-10* [20] consists of a total of 60,000 images of 32×32 pixels and 10 classes, including airplanes, birds, and cats. (2) *CIFAR-100(20)* [20] is similar to CIFAR-10, but with 100 classes. Each class contains 600 images, which are then grouped into 20 superclasses. We use both CIFAR-100 and CIFAR-20, a version using 20 superclasses. (3) *STL-10* [6] contains 10 classes of 96×96 pixel images, based on 13,000 labeled images and 100,000 unlabeled images.

Evaluation. The Hungarian method [22] was used to achieve the best bijection permutation mapping between the unsupervised results and ground-truth labels. Then, the mapping was finally evaluated with top-1 classification accuracy. Given a network with five classification heads, we select the head with the lowest training loss for our final model.

Results. Table 1 compares the performances against the baselines. Across the three datasets, CIFAR-10, CIFAR-20, and STL-10, our model outperforms baselines substantially in classification accuracy, even when compared to the best performing sequential and joint classification algorithms. In the case of STL-10, we evaluated the model trained with both unlabeled and labeled datasets for a fair comparison with IIC. For CIFAR-100, whose results are omitted in the table, our model reports 28.1% for the lowest loss head accuracy as opposed to 20.3% for the IIC model (i.e., the current SOTA model).

4.2 Component Analyses

We conducted an ablation study by repeatedly testing the performance after removing or modifying each module. Such a step can confirm the efficacy of each component. We then interpret the model's predictions via qualitative analysis.

Table 1. Comparison of different unsupervised classification models. We report the accuracy (%) of our model from the head with having the lowest training loss. Baseline results are excerpted from the IIC paper [18].

Network	CIFAR-10	CIFAR-20	STL-10
Random network	13.1	5.9	13.5
k-means [37]	22.9	13.0	19.2
Autoencoder (AE) [2]	31.4	16.5	30.3
SWWAE [43]	28.4	14.7	27.0
GAN [29]	31.5	15.1	29.8
JULE [41]	27.2	13.7	27.7
DEC [39]	30.1	18.5	35.9
DAC [5]	52.2	23.8	47.0
DeepCluster [4]	37.4	18.9	33.4
ADC [12]	32.5	16.0	53.0
IIC [18]	61.7	25.7	59.6
Our Model	**81.0**	**35.3**	**66.5**

First Stage Ablations. We modify the first stage unsupervised embedding algorithm to various alternatives including state-of-the-art embedding methods [16,38], and measure the changes in the embedding quality and the final classification accuracy. Embedding quality is measured by the weighted k-NN classifier considering that the training labels are known [16]. For each test sample, we retrieve 200 nearest training samples based on the similarities in the embedding space and perform weighted voting for measuring its embedding quality. Table 2 describes the embedding quality and corresponding classification accuracy on CIFAR-10, which shows that the Super-AND algorithm outperforms other initialization alternatives in terms of the embedding quality and classification accuracy. This finding is evidence that the quality of pretrained embedding contributes to the final classification accuracy. We note that the Lemniscate [38] algorithm performs poorly in classification, although its embedding accuracy is reasonable. We speculate that this trend is due to the encoder's ambiguous knowledge learned only by separating every instance over the embedding space.

Second Stage Ablations. We modify (1) the network that trains the classifier and (2) the kind of classifier algorithm. For the former, we evaluated the classifier performance without the consistency preserving loss L_{cp}, without weight normalization, and without temperature in the softmax function. All experiments had an equal setting of a pretrained encoder from the first stage. Table 3 displays the comparison results, which demonstrate the significance of each component. We find temperature τ_c to be critical for correctly training the classifier, which can be explained by the topological characteristics of the normalized vector, i.e., it has a confined space, and the temperature is critical for amplifying certain signals.

Table 2. Comparison of deep embedding algorithms on CIFAR-10. The better encoder our model has, the more precise prediction the model can produce.

Initialization	Embedding quality (k-NN accuracy)	Class assignment quality (Top-1 classification accuracy)
Random	–	58.6
Lemniscate [38]	80.8	63.8
AND [16]	86.3	73.9
Super-AND [13]	**89.2**	**81.0**

Table 3. Test of alternative algorithms on CIFAR-10. Every component of the model contributes to a performance increase, and the two-stage design is superior than the sequential or joint versions.

Alternatives	Accuracy
Without L_{cp} in Eq. 12	58.2
Without weight normalization	56.0
Without temperature (i.e., $\tau_c = 1$)	19.5
Sequential: first stage + k-means	38.3
Sequential: first stage + Hierarchical clustering	49.9
Joint: first & second stage	68.4
Two-stage model with full features	**81.0**

We next test alternative algorithms, including sequential and joint versions of our model. For the sequential method, k-means and hierarchical clustering are adopted for the class assignment, while maintaining the pretrained encoder of the proposed model. The joint approach combines the losses from both stages and optimize concurrently. The comparison results in the table report a substantial drop in the accuracy of these variants, implying that the proposed two-stage paradigm contributes to a performance gain.

Qualitative Analysis. Figure 3 illustrates the visual interpretation of our model's prediction by Grad-CAM [32]. The blue framed images are the success cases and imply that our model can capture unique visual traits for each class, such as legs in the horse class, funnels in the ship class, wings in the airplane class, and horns in the deer class. The red framed images are failure cases, in contrast, and show that these cases are unable to detect key visual traits for the given class.

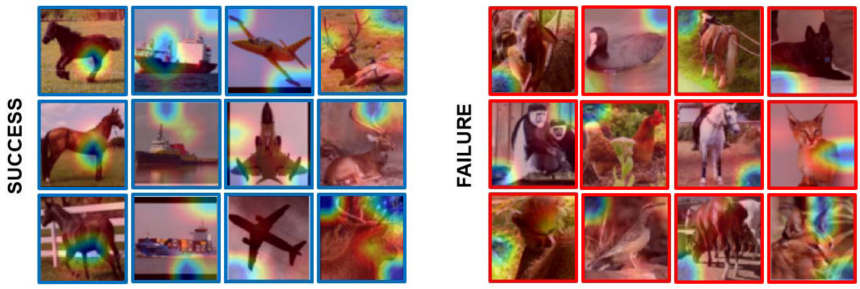


Fig. 3. Successes and failure cases from STL-10. Blue images are the success cases, where the highlighted part indicates how the model interpreted class traits based on the Grad-CAM visualization. Red images are the failure cases, where the model finds periphery areas to contain significant class traits. (Color figure online)

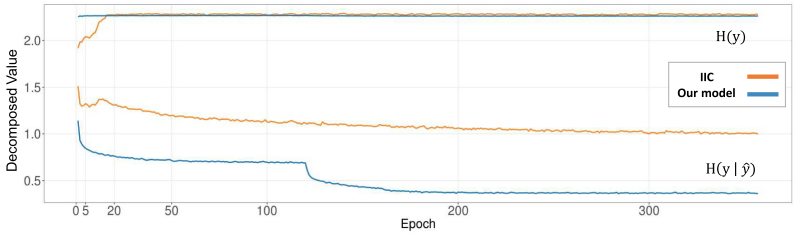
4.3 Improvement over SOTA

So far, we have demonstrated that the algorithm's superior performance. To understand what attributes to such novelty, we analyze the intermediate states of the proposed model against IIC, the current state-of-the-art. Figure 4(a) tracks the temporal changes in mutual information $I(y, \hat{y})$ in Eq. 7 by its two components: $H(y)$ and $H(y|\hat{y})$.

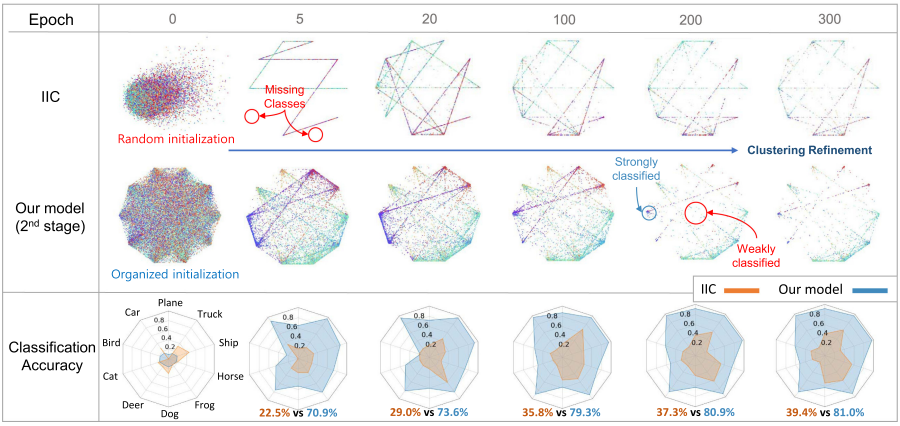
The figure shows that IIC gradually achieves higher $H(y)$, whereas our method starts with a high-value, thanks to the competitive advantage of the pretraining step. Note that a higher value indicates that data points are well divided across clusters. Next, IIC starts with a much larger $H(y|\hat{y})$ value compared to our model at epoch 0. Furthermore, while IIC gradually finds model instances with smaller $H(y|\hat{y})$ values, our method is more aggressive in identifying good clusters. Those drastic decreases are contributed by the consistency preserving loss, which enhances embedding refinement. Note that a lower value for this second term indicates that the cluster assignments of the original image and its augmentation version are closely related.

The visualization in Fig. 4(b) represents the corresponding clusters at different training epochs. Images appear well dispersed for our model at epoch 0. However, the lower $H(y)$ value for IIC is due to missing clusters, or those with zero matched images. This result implies that pretraining alleviates the mismatch between embedding and class assignments more effectively than IIC. The bottom row shows the per-class accuracy. The proposed model far exceeds IIC in the classification of images for every single class in all examined epochs.

We examined the confusion matrix between the ground truth labels and classification results in Fig. 5. The results are shown separately for our model trained after 300 epochs and IIC trained after 2,000 epochs. A perfect classification would only place items on the diagonal line. The figure shows that the proposed model eventually finds the right cluster for most images, although cluster assignments for some classes such as birds, cats, and dogs are error-prone.



(a) Change of mutual information (i.e., $H(y) - H(y|\hat{y})$) across the epoch.



(b) Changes in classification results and the accuracy of each class. The color of each dot represents its original label.

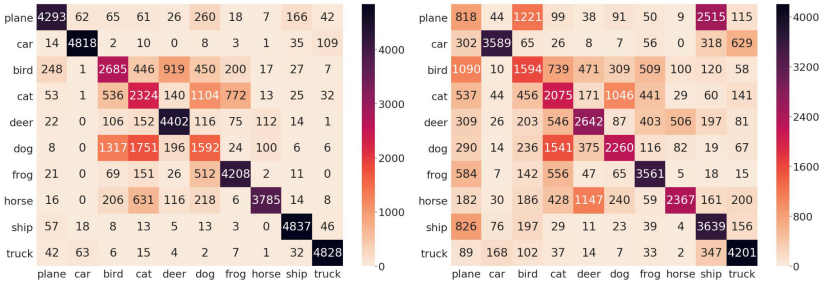
Fig. 4. Performance difference between the two models, IIC and ours, as they are trained on CIFAR-10. (a) Mutual information $I(y, \hat{y})$ of IIC is smaller than that of our model across all epochs as it is associated with the quality of classification depicted in (b) the form of clustering and classification accuracy.

Nonetheless, the IIC model, even after six times longer training epochs, achieves less accurate performance compared to our model.

4.4 Implication on Semi-supervised Learning

Can unsupervised learning methods bring new insights into well-known tasks? As a practical implication, we next discuss the potential of the proposed approach to assist in classical image classification tasks with a small amount of labeled data. As an empirical test, we compare the classification accuracy of (semi-) supervised models against their variants that utilize our method in the pretraining step.

Datasets. Two datasets are chosen for evaluating semi-supervised learning tasks. (1) the *CIFAR-10* dataset and (2) the *SVHN* dataset [26], which is a real-world digit dataset with 10 classes of 32 × 32 pixel images. We used only 4,000 labels in CIFAR-10 and 1,000 labels in SVHN for training. The unlabeled



(a) Our Model at 300 epochs (b) The IIC model at 2,000 epochs

Fig. 5. Comparison of confusion matrices. The final confusion matrices on CIFAR-10 are given, where columns are the predicted class, and rows are the actual class. Classifications on the diagonal line show that our model produces superior results than IIC.

Table 4. Classification accuracy on partially labeled datasets. Applying our model as a pretraining to the existing semi-supervised methods such as Π-Model and Mean Teacher yields enhancing the classification accuracy.

Algorithms	CIFAR-10	SVHN
Supervised	79.7	87.7
Π-Model [23]	83.8	92.2
Mean Teacher [33]	84.2	93.5
Our Model + Supervised	**85.4**	**93.4**
Our Model + Π-Model	**85.8**	**93.6**
Our Model + Mean Teacher	**88.0**	**94.2**

dataset is also used by semi-supervised learning models, excluding the supervised model.

Results. For both CIFAR-10 and SVHN, our model surpasses the fully-supervised baselines. Table 4 shows that the proposed model can even match some of the semi-supervised learning algorithms. By applying a simple Π-model [23] or the mean teacher model [33] to our pretrained network, a semi-supervised model can obtain meaningful starting points that contribute to performance improvement.

5 Conclusion

Unsupervised image classification has endless potential. This study presented a new two-stage algorithm for unsupervised image classification, where an embedding module precedes the refining module that concurrently performs embedding

and class assignment. The pretraining module in the first stage initializes data points and relaxes any mismatches between embedding and class assignment. The next stage uniquely introduces the L_{cp} loss term on the mutual information-based algorithm. Combinations of these stages led to massive gain over existing baselines across multiple datasets. These improvements have implications across a broad set of domains, including semi-supervised learning tasks.

Acknowledgement. We thank Cheng-Te Li and Yizhan Xu for their insights and discussions. This work was supported by the Institute for Basic Science (IBS-R029-C2) and the Basic Science Research Program through the National Research Foundation funded by the Ministry of Science and ICT in Korea (No. NRF-2017R1E1A1A01076400).

References

1. Baldi, P.: Autoencoders, unsupervised learning, and deep architectures. In: Proceedings of the ICML Workshop on Unsupervised and Transfer Learning, pp. 37–49 (2012)
2. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. In: Advances in Neural Information Processing Systems (NeurIPS), pp. 153–160 (2007)
3. Bojanowski, P., Joulin, A.: Unsupervised learning by predicting noise. In: Proceedings of the International Conference on Machine Learning (ICML), pp. 517–526 (2017)
4. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 132–149 (2018)
5. Chang, J., Wang, L., Meng, G., Xiang, S., Pan, C.: Deep adaptive image clustering. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 5879–5887 (2017)
6. Coates, A., Ng, A., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS), pp. 215–223 (2011)
7. Ding, C., He, X.: K-means clustering via principal component analysis. In: Proceedings of the International Conference on Machine Learning (ICML), p. 29 (2004)
8. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. of the International Conference on Knowledge Discovery and Data Mining (SIGKDD), pp. 226–231 (1997)
9. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. arXiv preprint arXiv:1803.07728 (2018)
10. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Hoboken (2016)
11. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems (NeurIPS), pp. 2672–2680 (2014)

12. Haeusser, P., Plapp, J., Golkov, V., Aljalbout, E., Cremers, D.: Associative deep clustering: training a classification network with no labels. In: Brox, T., Bruhn, A., Fritz, M. (eds.) GCPR 2018. LNCS, vol. 11269, pp. 18–32. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-12939-2_2
13. Han, S., Xu, Y., Park, S., Cha, M., Li, C.T.: A Comprehensive Approach to Unsupervised Embedding Learning based on AND Algorithm. arXiv preprint arXiv:2002.12158 (2020)
14. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
15. Hu, W., Miyato, T., Tokui, S., Matsumoto, E., Sugiyama, M.: Learning discrete representations via information maximizing self-augmented training. In: Proceedings of the International Conference on Machine Learning (ICML), pp. 1558–1567 (2017)
16. Huang, J., Dong, Q., Gong, S., Zhu, X.: Unsupervised deep learning by neighbourhood discovery. In: Proceedings of the International Conference on Machine Learning (ICML), pp. 2849–2858 (2019)
17. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
18. Ji, X., Henriques, J.F., Vedaldi, A.: Invariant information clustering for unsupervised image classification and segmentation. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 9865–9874 (2019)
19. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
20. Krizhevsky, A.: Learning multiple layers of features from tiny images. Technical report. Citeseer (2009)
21. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems (NeurIPS), pp. 1097–1105 (2012)
22. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Res. Logist. Q. **2**(1–2), 83–97 (1955)
23. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. arXiv preprint arXiv:1610.02242 (2016)
24. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436 (2015)
25. Lloyd, S.: Least squares quantization in PCM. IEEE Trans. Inf. Theory **28**(2), 129–137 (1982)
26. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning (2011)
27. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 69–84. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_5
28. Paninski, L.: Estimation of entropy and mutual information. Neural Comput. **15**(6), 1191–1253 (2003)
29. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
30. Salimans, T., Kingma, D.P.: Weight normalization: a simple reparameterization to accelerate training of deep neural networks. In: Advances in Neural Information Processing Systems (NeurIPS), pp. 901–909 (2016)

31. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 815–823 (2015)
32. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 618–626 (2017)
33. Tarvainen, A., Valpola, H.: Mean teachers are better role models: weight-averaged consistency targets improve semi-supervised deep learning results. In: Advances in Neural Information Processing Systems (NeurIPS), pp. 1195–1204 (2017)
34. Trigeorgis, G., Bousmalis, K., Zafeiriou, S., Schuller, B.: A deep semi-NMF model for learning hidden representations. In: Proceedings of the International Conference on Machine Learning (ICML), pp. 1692–1700 (2014)
35. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research **11**, 3371–3408 (2010)
36. Wang, F., Xiang, X., Cheng, J., Yuille, A.L.: Normface: L2 hypersphere embedding for face verification. In: Proceedings of the ACM International Multimedia Conference, pp. 1041–1049 (2017)
37. Wang, J., Wang, J., Song, J., Xu, X.S., Shen, H.T., Li, S.: Optimized cartesian k-means. IEEE Trans. Knowl. Data Eng. **27**(1), 180–192 (2014)
38. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3733–3742 (2018)
39. Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: Proceedings of the International Conference on Machine Learning (ICML), pp. 478–487 (2016)
40. Yang, B., Fu, X., Sidiropoulos, N.D., Hong, M.: Towards k-means-friendly spaces: simultaneous deep learning and clustering. In: Proceedings of the International Conference on Machine Learning (ICML), pp. 3861–3870 (2017)
41. Yang, J., Parikh, D., Batra, D.: Joint unsupervised learning of deep representations and image clusters. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5147–5156 (2016)
42. Ye, M., Zhang, X., Yuen, P.C., Chang, S.F.: Unsupervised embedding learning via invariant and spreading instance feature. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6210–6219 (2019)
43. Zhao, J., Mathieu, M., Goroshin, R., Lecun, Y.: Stacked what-where auto-encoders. arXiv preprint arXiv:1506.02351 (2015)