# MICE: MIXTURE OF CONTRASTIVE EXPERTS FOR UNSUPERVISED IMAGE CLUSTERING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

We present Mixture of Contrastive Experts (MiCE), a unified probabilistic clustering framework that simultaneously exploits the discriminative representations learned by contrastive learning and the semantic structures captured by a latent mixture model. Motivated by the mixture of experts, MiCE employs a gating function to partition an unlabeled dataset into subsets according to the latent semantics and multiple experts to discriminate distinct subsets of instances assigned to them in a contrastive learning manner. To solve the nontrivial inference and learning problems caused by the latent variables, we further develop a scalable variant of the Expectation-Maximization (EM) algorithm for MiCE. Empirically, we evaluate the clustering performance of MiCE on four widely adopted natural image datasets. MiCE achieves significantly better results than various previous methods and a strong contrastive learning baseline.

## 1 INTRODUCTION

Unsupervised clustering is a fundamental task that aims to partition data into distinct groups of similar ones without explicit human labels. Deep clustering methods (Xie et al., 2016; Wu et al., 2019) exploit the representations learned by neural networks and have made large progress on high-dimensional data recently. Often, such methods learn the representations for clustering by reconstructing data in a deterministic (Ghasedi Dizaji et al., 2017) or probabilistic manner (Jiang et al., 2016), or maximizing certain mutual information (Hu et al., 2017; Ji et al., 2019) (see Sec. 2 for the related work). Despite the recent advances, the representations learned by existing methods may not be discriminative enough to capture the semantic similarity between images.

The instance discrimination task (Wu et al., 2018; He et al., 2019) in contrastive learning has shown promise in pre-training representations transferable to downstream tasks through fine-tuning. We hypothesize that such representations can improve the clustering performance. A straightforward approach is to learn a classical clustering model, e.g. spherical $k$-means (Dhillon & Modha, 2001), directly on the representations pre-trained by the task. Such a two-stage baseline can achieve excellent clustering results (see empirical evidence in Tab. 2). However, because of the independence of the two stages, the baseline may not fully explore the semantic structures of the data when learning the representations and lead to a sub-optimal solution for clustering.

To this end, we propose Mixture of Contrastive Experts (MiCE), a unified probabilistic clustering method that utilizes the instance discrimination task as a stepping stone to improve clustering. In particular, to capture the semantic structure explicitly, we formulate a mixture of conditional models by introducing latent variables to represent cluster labels of the images, which is inspired by the mixture of experts (MoE) formulation. In MiCE, each of the conditional models, also called an *expert*, learns to discriminate a subset of instances, while an input-dependent *gating function* partitions the dataset into subsets according to the latent semantics by allocating weights among experts. Further, we develop a scalable variant of the Expectation-Maximization (EM) algorithm (Dempster et al., 1977) for the nontrivial inference and learning problems in MiCE. In the E-step, we obtain the approximate inference of the posterior distribution of the latent variables given the observed data. In the M-step, we maximize the evidence lower bound (ELBO) of the log conditional likelihood with respect to all parameters. Moreover, we carefully discuss the algorithmic relation between MiCE and the two-stage baseline and show that the latter is actually a special instance of the former in a certain extreme case.

Compared with existing clustering methods, MiCE has the following advantages. (i) **Methodologically unified**: MiCE conjoins the benefits of both the discriminative representations learned by contrastive learning and the semantic structures captured by a latent mixture model within a unified probabilistic framework. (ii) **Free from regularization**: MiCE trained by EM optimizes a single objective function, which does not require auxiliary loss or regularization terms. (iii) **Empirically effective**: Evaluated on four widely adopted natural image datasets, MiCE achieves significantly better results than a strong contrastive baseline and extensive prior clustering methods on several benchmarks.

## 2 Related Work

**Deep clustering.** Inspired by the success of deep learning, many researchers propose to learn the representations and cluster assignments simultaneously (Xie et al., 2016; Yang et al., 2016; 2017) based on data reconstruction (Xie et al., 2016; Yang et al., 2017), pairwise relationship among instances (Chang et al., 2017; Haeusser et al., 2018; Wu et al., 2019), multi-task learning (Shiran & Weinshall, 2019; Niu et al., 2020), etc. The joint training framework often ends up optimizing a weighted average of multiple loss functions. However, given that the validation dataset is barely provided, tuning the weights between the losses may be difficult in real-world clustering settings (Ghasedi Dizaji et al., 2017).

Recently, several methods also explore probabilistic modeling, and they introduce latent variables to represent the underlying classes. On one hand, deep generative approaches (Jiang et al., 2016; Dilokthanakul et al., 2016; Chongxuan et al., 2018; Mukherjee et al., 2019; Yang et al., 2019) attempt to capture the data generation process with a mixture of Gaussian prior on latent representations. However, the imposed assumptions can be violated in many cases, and capturing the true data distribution is challenging but may not be helpful to the clustering (Krause et al., 2010). On the other hand, discriminative approaches (Hu et al., 2017; Ji et al., 2019; Darlow & Storkey, 2020) directly model the mapping from the inputs to the cluster labels and maximize a form of mutual information, which often yields superior cluster accuracy. Despite the simplicity, the discriminative approaches discard the instance-specific details that can benefit clustering via improving the learned representations.

Besides, MIXAE (Zhang et al., 2017), DAMIC (Chazan et al., 2019), and MoE-Sim-VAE (Kopf et al., 2019) combine the mixture of experts (MoE) formulation (Jacobs et al., 1991) with the data reconstruction task. However, either pre-training, regularization, or an extra clustering loss is required.

**Contrastive learning.** To learn discriminative representations, contrastive learning (Wu et al., 2018; Oord et al., 2018; He et al., 2019; Tian et al., 2019; Chen et al., 2020) incorporates various contrastive loss functions with different pretext tasks such as colorization (Zhang et al., 2016), context auto-encoding (Pathak et al., 2016), and instance discrimination (Dosovitskiy et al., 2015; Wu et al., 2018). The pre-trained representations often achieve promising results on downstream tasks, *e.g.*, depth prediction, object detection (Ren et al., 2015; He et al., 2017), and image classification (Kolesnikov et al., 2019), after fine-tuning with human labels. In particular, InstDisc (Wu et al., 2018) learns from instance-level discrimination using NCE (Gutmann & Hyvärinen, 2010), and maintains a memory bank to compute the loss function efficiently. MoCo replaces the memory bank with a queue and maintains an EMA of the student network as the teacher network to encourage consistent representations. A concurrent work called PCL (Li et al., 2020) also explores the semantic structures in contrastive learning. They add an auxiliary cluster-style objective function on top of the MoCo's original objective, which differs from our method significantly. PCL requires an auxiliary $k$-means (Lloyd, 1982) algorithm to obtain the posterior estimates and the prototypes. Moreover, their aim of clustering is to induce transferable embeddings instead of discovering groups of data that correspond to underlying semantic classes.

## 3 Mixture of Contrastive Experts

We consider the unsupervised problem of partitioning a given unlabeled dataset with $N$ observations, denoted as $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$, into $K$ clusters. To encourage discriminative representations for clustering, we adopt the instance discrimination task from contrastive learning (Wu et al., 2018; He et al., 2019) where each of the datapoint $\mathbf{x}_n$ is assigned with a unique surrogate label $y_n \in \{1, 2, ..., N\}$ such that

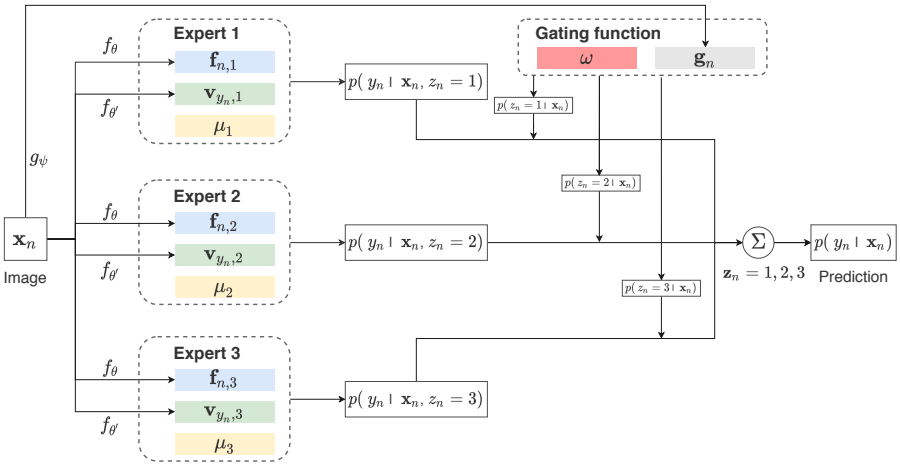


Figure 1: An illustration of MiCE with three experts. Best view in color.

$y_n \neq y_j, \forall j \neq n$[1]. Let the latent variable $z_n \in \{1, 2, ..., K\}$ be the cluster label of the image $\mathbf{x}_n$. We consider the model of the form:

$$p(\mathbf{Y}, \mathbf{Z}|\mathbf{X}) = \prod_{n=1}^{N} \prod_{k=1}^{K} p(y_n, z_n = k|\mathbf{x}_n) = \prod_{n=1}^{N} \prod_{k=1}^{K} p(z_n = k|\mathbf{x}_n) p(y_n|\mathbf{x}_n, z_n = k). \quad (1)$$

The formulation explicitly introduces a mixture model to capture the latent semantic structures, which is inspired by the mixture of experts (MoE) framework (Jacobs et al., 1991). In Eq. (1), $p(y_n|\mathbf{x}_n, z_n)$ is one of the *experts* that learn to discriminate a subset of instances and $p(z_n|\mathbf{x}_n)$ is a *gating function* that partitions the dataset into subsets according to the latent semantics by routing the given input to one or a few experts. With a divide-and-conquer principle, the experts are often highly specialized in particular images that share similar semantics, which improves the learning efficiency. Also see Fig. 1 for an illustration of MiCE with three experts.

In contrast to the original MoE used in the supervised settings (Jacobs et al., 1991), our experts learn from instance-wise discrimination instead of human labels. In addition, both gating and expert parts of MiCE are based on deep neural networks (DNNs) to fit the high-dimensional data. In the following, we will elaborate on how we parameterize the gating function and the experts to fit the clustering task. For simplicity, we omit the parameters in all probability distributions in this section.

**Gating function.** The gating function organizes the instance discrimination task into $K$ simpler subtasks by weighting the experts based on the semantics of the input image. We define $g_\psi$ as a feature embedding network that outputs an embedding for each input image. We denote the output vector for image $\mathbf{x}_n$ as $\mathbf{g}_n \in \mathbb{R}^d$. The gating function is then parameterized as:

$$p(z_n|\mathbf{x}_n) = \frac{\exp(\boldsymbol{\omega}_{z_n}^\top \mathbf{g}_n / \kappa)}{\sum_{k=1}^{K} \exp(\boldsymbol{\omega}_k^\top \mathbf{g}_n / \kappa)}, \quad (2)$$

where $\kappa$ is the temperature hyper-parameter that controls the level of concentration (Hinton et al., 2015), and $\boldsymbol{\omega} = \{\boldsymbol{\omega}_k\}_{k=1}^{K}$ represent the gating prototypes. All prototypes and image embeddings are $\ell_2$-normalized in the $\mathbb{R}^d$ space. Hence, the gating function performs a soft partitioning of the dataset based on the cosine similarity between the image embeddings and the gating prototypes. We can view it as a prototype-based discriminative clustering module, whereas we obtain the cluster labels using posterior inference to consider additional information in the experts.

**Experts.** In MiCE, every expert learns to solve the instance discrimination subtask arranged by the gating function. We define the expert in terms of the unnormalized model $\Phi(\cdot)$ following Wu et al. (2018); He et al. (2019). Specifically, the probability of the image $\mathbf{x}_n$ being recognized as the $y_n$-th

[1]The value of the surrogate label can be regarded as the index of the image.

one by the $z_n$-th expert is formulated as follows:

$$p(y_n|\mathbf{x}_n, z_n) = \frac{\Phi(\mathbf{x}_n, y_n, z_n)}{Z(\mathbf{x}_n, z_n)}, \tag{3}$$

where $Z(\mathbf{x}_n, z_n) = \sum_{l=1}^{N} \Phi(\mathbf{x}_n, y_l, z_n)$ is a normalization constant that is often computationally intractable. Let $\mathbf{f}_{n,z_n} \in \mathbb{R}^d$ and $\mathbf{v}_{y_n,z_n} \in \mathbb{R}^d$ be the student embedding and the teacher embedding for images $\mathbf{x}_n$ in the $z_n$-th expert respectively. We parameterize the unnormalized model as:

$$\Phi(\mathbf{x}_n, y_n, z_n) = \exp\left(\mathbf{v}_{y_n,z_n}^\top \left(\mathbf{f}_{n,z_n} + \boldsymbol{\mu}_{z_n}\right)/\tau\right), \tag{4}$$

where $\tau$ and $\boldsymbol{\mu} = \{\boldsymbol{\mu}_k\}_{k=1}^K$ represent the temperature hyper-parameter and the cluster prototypes for the experts, respectively. In Eq. (4), the first instance-wise dot product explores the *instance-level* information to induce discriminative representations within each expert. The second instance-prototype dot product incorporates the *class-level* information in representation learning, encouraging a clear cluster structure around the prototype. Overall, the learned embeddings are therefore encoded with semantic structures while being discriminative enough to represent the instances. Eq. (4) is built upon MoCo (He et al., 2019), the recent state-of-the-art contrastive learning method. In principle, many other potential solutions exist to define the experts, which are left for future studies.

**Details on DNNs.** For the experts, the image embeddings come from two feature embedding networks $f_{\boldsymbol{\theta}}$ and $f_{\boldsymbol{\theta}'}$ in which the parameters are shown in the subscripts. We use $f_{\boldsymbol{\theta}}$ to denote the student network that maps the image $\mathbf{x}_n$ into $K$ continuous embeddings $\mathbf{f}_n = \{\mathbf{f}_{n,k}\}_{k=1}^K \in \mathbb{R}^{K\times d}$. Similarly, the teacher network $f_{\boldsymbol{\theta}'}$ produces $\mathbf{v}_{y_n} = \{\mathbf{v}_{y_n,k}\}_{k=1}^K \in \mathbb{R}^{K\times d}$ given $\mathbf{x}_n$. In order to produce high quality and consistent embeddings to guide the learning process, the teacher network $f_{\boldsymbol{\theta}'}$ is set to be the exponential moving average (EMA) version of the student network (Tarvainen & Valpola, 2017; He et al., 2019). The student network $f_{\boldsymbol{\theta}}$ is implemented as a DNN with $K$ output layers on top of a shared feature extraction module with the expert network $g_{\boldsymbol{\psi}}$ which has a single output layer. All the output layers generate a $d$-dimensional embedding that is normalized by its $\ell_2$-norm. Please refer to the Appendix C for the full specification.

## 4 Inference and Learning

We first discuss the evidence lower bound (ELBO), the single objective used in MiCE, in Sec. 4.1. Then, we present a scalable variant of the Expectation-Maximization (EM) algorithm (Dempster et al., 1977) to deal with the non-trivial inference and learning of MiCE in Sec. 4.2. Lastly, in Sec. 4.3, we show that a naïve two-stage baseline, in which we run a spherical $k$-means algorithm on the embeddings learned by MoCo (He et al., 2019), is a special case of MiCE.

### 4.1 Evidence lower bound (ELBO)

The parameters to update include the parameters $\boldsymbol{\theta}, \boldsymbol{\psi}$ of the student and gating network respectively, and the expert prototypes $\boldsymbol{\mu} = \{\boldsymbol{\mu}\}_{k=1}^K$. The learning objective of MiCE is to maximize the evidence lower bound (ELBO) of the log conditional likelihood of the entire dataset. The ELBO of the datapoint $n$ is given by:

$$\begin{aligned}\log p(y_n|\mathbf{x}_n) &\geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\mu}; \mathbf{x}_n, y_n) \\ &:= \mathbb{E}_{q(z_n|\mathbf{x}_n, y_n)}[\log p(y_n|\mathbf{x}_n, z_n; \boldsymbol{\theta}, \boldsymbol{\mu})] - D_{\mathrm{KL}}(q(z_n|\mathbf{x}_n, y_n)\|p(z_n|\mathbf{x}_n; \boldsymbol{\psi})),\end{aligned} \tag{5}$$

where $q(z_n|\mathbf{x}_n, y_n)$ is a variational distribution to infer the latent cluster label given the observed data. The first term in Eq. (5) encourages $q(z_n|\mathbf{x}_n, y_n)$ to be high for the experts that are good at discriminating the input images. Intuitively, it can relief the potential *degeneracy* issue (Caron et al., 2018; Ji et al., 2019), where all images are assigned to the same cluster. This is because a degenerated posterior puts the pressure of discriminating all images on a single expert, which may result in a looser ELBO. The second term in Eq. (5) is the Kullback–Leibler divergence between the variational distribution and the distribution defined by the gating function. With this term, the gating function is refined during training and considers the capability of the experts when partitioning data. Notably, MiCE does not rely on auxiliary loss or regularization terms as many prior methods (Haeusser et al., 2018; Shiran & Weinshall, 2019; Wu et al., 2019; Niu et al., 2020) do.

### 4.2 EM ALGORITHM

**E-step.** Inferring the posterior distribution of latent variables given the observations is an important step to apply MiCE to clustering. According to Bayes' theorem, the posterior distribution given the current estimate of the model parameters is:

$$p(z_n|\mathbf{x}_n, y_n; \boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\mu}) = \frac{p(z_n|\mathbf{x}_n; \boldsymbol{\psi}) p(y_n|\mathbf{x}_n, z_n; \boldsymbol{\theta}, \boldsymbol{\mu})}{\sum_{k=1}^{K} p(k|\mathbf{x}_n; \boldsymbol{\psi}) p(y_n|\mathbf{x}_n, k; \boldsymbol{\theta}, \boldsymbol{\mu})}. \tag{6}$$

Comparing with the gating function $p(z_n|\mathbf{x}_n; \boldsymbol{\psi})$, the posterior provides better estimates of the latent variables by incorporating the supplementary information of the experts. However, we cannot tractably compute the posterior distribution because of the normalization constants $Z(\mathbf{x}_n, z_n; \boldsymbol{\theta}, \boldsymbol{\mu})$. In fact, given the image $\mathbf{x}_n$ and the cluster label $z_n$, $Z(\mathbf{x}_n, z_n; \boldsymbol{\theta}, \boldsymbol{\mu})$ sums over the entire dataset, which is prohibitive for large-scale image dataset. We present a simple and analytically tractable estimator to approximate them. Specifically, we maintain a queue $\mathbf{q} \in \mathbb{R}^{\nu \times K \times D}$ that stores $\nu$ previous outputs of the teacher network, following MoCo closely. Formally, the estimator $\hat{Z}(\cdot)$ is:

$$\hat{Z}(\mathbf{x}_n, z_n; \boldsymbol{\theta}, \boldsymbol{\mu}) = \exp\left(\mathbf{v}_{y_{n,z_n}}^{\top} \left(\mathbf{f}_{n,z_n} + \boldsymbol{\mu}_{z_n}\right)/\tau\right) + \sum_{i=1}^{\nu} \exp\left(\mathbf{q}_{i,z_n}^{\top} \left(\mathbf{f}_{n,z_n} + \boldsymbol{\mu}_{z_n}\right)/\tau\right). \tag{7}$$

The estimator is biased, while its bias decreases as $\nu \to \infty$ and we can get a sufficient amount of embeddings from the queue $\mathbf{q}$ efficiently[2]. With Eq. (7), we approximate the posterior as:

$$q(z_n|\mathbf{x}_n, y_n; \boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\mu}) = \frac{p(z_n|\mathbf{x}_n; \boldsymbol{\psi}) \Phi\left(\mathbf{x}_n, y_n, z_n; \boldsymbol{\theta}, \boldsymbol{\mu}\right) / \hat{Z}(\mathbf{x}_n, z_n; \boldsymbol{\theta}, \boldsymbol{\mu})}{\sum_{k=1}^{K} p(k|\mathbf{x}_n; \boldsymbol{\psi}) \Phi\left(\mathbf{x}_n, y_n, k; \boldsymbol{\theta}, \boldsymbol{\mu}\right) / \hat{Z}(\mathbf{x}_n, k; \boldsymbol{\theta}, \boldsymbol{\mu})}. \tag{8}$$

**M-step.** We leverage the stochastic gradient ascent to optimize ELBO with respect to the network parameters $\boldsymbol{\theta}, \boldsymbol{\psi}$ and the expert prototypes $\boldsymbol{\mu}$. We approximate the normalization constants appear in ELBO in analogy to the E-step, formulated as follows:

$$\begin{aligned} \widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\mu}; \mathbf{x}_n, y_n) = {} & \mathbb{E}_{q(z_n|\mathbf{x}_n, y_n; \boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\mu})} \left[\log \frac{\Phi\left(\mathbf{x}_n, y_n, z_n; \boldsymbol{\theta}, \boldsymbol{\mu}\right)}{\hat{Z}(\mathbf{x}_n, z_n; \boldsymbol{\theta}, \boldsymbol{\mu})}\right] \\ & - D_{\mathrm{KL}}(q(z_n|\mathbf{x}_n, y_n; \boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\mu}) \| p(z_n|\mathbf{x}_n; \boldsymbol{\psi})). \end{aligned} \tag{9}$$

Sampling a mini-batch $B$ of datapoints, we can construct an efficient stochastic estimator of ELBO over the full dataset to learn $\boldsymbol{\theta}, \boldsymbol{\psi}$ and $\boldsymbol{\mu}$:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\mu}; \mathbf{X}, \mathbf{Y}) \approx \frac{N}{|B|} \sum_{n \in B} \widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\mu}; \mathbf{x}_n, y_n). \tag{10}$$

It requires additional care on the update of the prototypes, as discussed in many clustering methods (Sculley, 2010; Xie et al., 2016; Yang et al., 2017; Shiran & Weinshall, 2019). Some of them carefully adjust the learning rate of each prototype separately (Sculley, 2010; Yang et al., 2017), which can be very different from the one used for the network parameters. Since evaluating different learning rate schemes on the validation dataset is often infeasible in unsupervised clustering, we employ alternative strategies which are free from using per-prototype learning rates in MiCE.

As for the expert prototypes, we observe that using only the stochastic update can lead to bad local optima. Therefore, at the end of each training epoch, we apply an additional analytical update derived from the ELBO as follows:

$$\hat{\boldsymbol{\mu}}_k = \sum_{n:\hat{z}_n = k} \mathbf{v}_{y_n,k}, \quad \boldsymbol{\mu}_k = \frac{\hat{\boldsymbol{\mu}}_k}{\|\hat{\boldsymbol{\mu}}_k\|_2}, \quad \forall k, \tag{11}$$

where $\forall n$, $\hat{z}_n = \arg\max_k \; q(k|\mathbf{x}_n, y_n; \boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\mu})$ is the hard assignment of the cluster label. Please refer to Appendix A.2 for the detailed derivation. Intuitively, the analytical update in Eq. (11) considers all the teacher embeddings assigned to the $k$-th cluster, instead of only the ones in a mini-batch, to avoid bad local optima.

[2]Even though the bias does not vanish, we find that the approximation works well empirically.

Beside, we fix the gating prototypes $\boldsymbol{\omega}$ to a set of pre-defined embeddings to stabilize the training process. However, using randomly initialized prototypes may cause unnecessary difficulties in partitioning the dataset if some of them are crowded together. We address the potential issue by using the means of a Max-Mahalanobis distribution (MMD) (Pang et al., 2018) which is a special case of the mixture of Gaussian distribution. The untrainable means in MMD provide the optimal inter-cluster dispersion (Pang et al., 2019) that stabilizes the gating outputs. We provide a systematical ablation study in Tab. 3 to investigate the effect of the updates on $\boldsymbol{\omega}$ and $\boldsymbol{\mu}$.

### 4.3 Relations to a two-stage baseline

The combination of a contrastive learning method and a clustering method is a natural baseline of MiCE. Our analysis reveals that MiCE is the general form of the two-stage baseline in which we learn the image embeddings with MoCo (He et al., 2019) and subsequently run a spherical $k$-means algorithm (Dhillon & Modha, 2001) to obtain the cluster labels.

On one hand, in the extreme case where $\kappa \to \infty$ (Assumption A3), the student embeddings $\mathbf{f}_{n,k}$ and teacher embeddings $\mathbf{v}_{y_n,k}$ are identical for different $k$ (Assumption A4), and the class-level information in Eq. (4) is omitted (Assumption A5), we arrive at the same Softmax classifier and objective function used by MoCo (He et al., 2019) as a special case of our method. On the other hand, of particular relevance to the analytical update on expert prototypes (Eq. (11)) is the spherical $k$-means algorithm (Dhillon & Modha, 2001) that leverages the cosine similarity to cluster $\ell_2$-normalized data (Hornik et al., 2012). In addition to Assumptions A3 and A4, if we assume the unnormalized model is perfectly self-normalized (Assumption A2), using the hard assignment to get the cluster labels together with the analytical update turns out to be a single-iteration spherical $k$-means algorithm on the teacher embeddings. Please refer to the Appendix B for a detailed derivation.

The performance of the baseline is limited by the independence of the representation learning stage and the clustering stage. In contrast, MiCE provides a unified framework to align the representation learning and clustering objectives in a principled manner. See a comprehensive comparison in Tab. 2.

## 5 Experiments

In this section, we present experimental results to demonstrate the effectiveness of MiCE. We compare MiCE with extensive prior clustering methods and the contrastive learning based two-stage baseline on four widely adopted benchmarking datasets for clustering, including STL-10 (Coates et al., 2011), CIFAR-10 (Krizhevsky & Hinton, 2009), CIFAR-100 (Krizhevsky & Hinton, 2009), and ImageNet-Dog (Chang et al., 2017). The evaluation protocol follows the literature closely (Chang et al., 2017; Wu et al., 2019; Ji et al., 2019; Shiran & Weinshall, 2019; Darlow & Storkey, 2020). For CIFAR-10, CIFAR-100, and STL-10, all the training and test images are jointly utilized, and the 20 superclasses of CIFAR-100 are used instead of the fine labels. The 15 classes of dog images are selected from the ILSVRC2012 1K (Deng et al., 2009) dataset and resized to $96 \times 96 \times 3$ to form the ImageNet-Dog dataset. Note that the numbers of the clusters are known in advance as in Chang et al. (2017); Ji et al. (2019); Wu et al. (2019); Shiran & Weinshall (2019). The statistics of the datasets are summarized in Tab. 1. We adopt three common metrics to evaluate the clustering performance, namely normalized mutual information (NMI), cluster accuracy (ACC), and adjusted rand index (ARI).

Table 1: Statistics of the datasets.

| Dataset | Images | Clusters | Image Size |
|---|---|---|---|
| CIFAR-10 | 60,000 | 10 | $32 \times 32 \times 3$ |
| CIFAR-100 | 60,000 | 20 | $32 \times 32 \times 3$ |
| STL-10 | 13,000 | 10 | $96 \times 96 \times 3$ |
| ImageNet-Dog | 19,500 | 15 | $96 \times 96 \times 3$ |

Regarding the network architecture, we use a 34-layer ResNet (ResNet-34) (He et al., 2016) backbone following the recent methods (Ji et al., 2019; Shiran & Weinshall, 2019) for fair comparisons. For the student network, the final output layer is replaced by $K$ fully connected layers. Each of them generates a 128-dimensional embedding with a unit norm. The gating network is a ResNet-34 model with a single output layer. The parameters of the student and gating networks are shared except for the output layers. The architecture of the teacher network follows the student network, while

Table 2: Unsupervised clustering performance of different methods on three datasets. The first sector presents the results from the literature, the later ones display the results of the baseline and the proposed MiCE. In the last two sectors, the bold results indicating the one with the highest values. Methods with the legend† are the ones that required post-processing by $k$-means to obtain the clusters since they do not learn the clustering function directly, except that we use spherical $k$-means for MoCo. We calculate the mean and standard deviation (Std.) of MiCE and MoCo based on five runs.

| Datasets | CIFAR-10 | | | CIFAR-100 | | | STL-10 | | | ImageNet-Dog | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Methods/Metrics | NMI | ACC | ARI | NMI | ACC | ARI | NMI | ACC | ARI | NMI | ACC | ARI |
| $k$-means (Lloyd, 1982) | 8.7 | 22.9 | 4.9 | 8.40 | 13.0 | 2.8 | 12.5 | 19.2 | 6.1 | 5.5 | 10.5 | 2.0 |
| SC (Zelnikmanor & Perona, 2004) | 10.3 | 24.7 | 8.5 | 9.0 | 13.6 | 2.2 | 9.8 | 15.9 | 4.8 | 3.8 | 11.1 | 1.3 |
| AC (Gowda & Krishna, 1978) | 10.5 | 22.8 | 6.5 | 9.8 | 13.8 | 3.4 | 23.9 | 33.2 | 14.0 | 3.7 | 13.9 | 2.1 |
| NMF (Cai et al., 2009) | 8.1 | 19.0 | 3.4 | 7.9 | 11.8 | 2.6 | 9.6 | 18.0 | 4.6 | 4.4 | 11.8 | 1.6 |
| AE† (Bengio et al., 2006) | 23.9 | 31.4 | 16.9 | 10.0 | 16.5 | 4.8 | 25.0 | 30.3 | 16.1 | 10.4 | 18.5 | 7.3 |
| DAE† (Vincent et al., 2010) | 25.1 | 29.7 | 16.3 | 11.1 | 15.1 | 4.6 | 22.4 | 30.2 | 15.2 | 10.4 | 19.0 | 7.8 |
| SWWAE† (zhao et al., 2015) | 23.3 | 28.4 | 16.4 | 10.3 | 14.7 | 3.9 | 19.6 | 27.0 | 13.6 | 9.4 | 15.9 | 7.6 |
| GAN† (radford et al., 2015) | 26.5 | 31.5 | 17.6 | 12.0 | 15.3 | 4.5 | 21.0 | 29.8 | 13.9 | 12.1 | 17.4 | 7.8 |
| VAE† (Kingma & Welling, 2013) | 24.5 | 29.1 | 16.7 | 10.8 | 15.2 | 4.0 | 20.0 | 28.2 | 14.6 | 10.7 | 17.9 | 7.9 |
| JULE (Yang et al., 2016) | 19.2 | 27.2 | 13.8 | 10.3 | 13.7 | 3.3 | 18.2 | 27.7 | 16.4 | 5.4 | 13.8 | 2.8 |
| DEC (Xie et al., 2016) | 25.7 | 30.1 | 16.1 | 13.6 | 18.5 | 5.0 | 27.6 | 35.9 | 18.6 | 12.2 | 19.5 | 7.9 |
| DAC (Chang et al., 2017) | 39.6 | 52.2 | 30.6 | 18.5 | 23.8 | 8.8 | 36.6 | 47.0 | 25.7 | 21.9 | 27.5 | 11.1 |
| DCCM (Wu et al., 2019) | 49.6 | 62.3 | 40.8 | 28.5 | 32.7 | 17.3 | 37.6 | 48.2 | 26.2 | 32.1 | 38.3 | 18.2 |
| IIC (Ji et al., 2019) | - | 61.7 | - | - | 25.7 | - | - | 49.9 | - | - | - | - |
| DHOG (Darlow & Storkey, 2020) | 58.5 | 66.6 | 49.2 | 25.8 | 26.1 | 11.8 | 41.3 | 48.3 | 27.2 | - | - | - |
| AttentionCluster (Niu et al., 2020) | 47.5 | 61.0 | 40.2 | 21.5 | 28.1 | 11.6 | 44.6 | 58.3 | 36.3 | 28.1 | 32.2 | 16.3 |
| MMDC (Shiran & Weinshall, 2019) | 57.2 | 70.0 | - | 25.9 | 31.2 | - | 49.8 | 61.1 | - | - | - | - |
| MoCo (Mean)† (He et al., 2019) | 66.0 | 74.7 | 59.3 | 38.8 | 39.5 | 24.0 | 60.5 | 70.7 | 53.0 | 34.2 | 30.8 | 18.4 |
| MoCo (Std.)† (He et al., 2019) | 0.6 | 1.7 | 0.9 | 0.2 | 0.1 | 0.4 | 0.9 | 2.0 | 0.8 | 0.3 | 1.7 | 0.9 |
| MiCE (Mean, **Ours**) | **73.5** | **83.4** | **69.5** | **43.0** | **42.2** | **27.7** | **61.3** | **72.0** | **53.2** | **39.4** | **39.0** | **24.7** |
| MiCE (Std., **Ours**) | 0.2 | 0.2 | 0.3 | 0.5 | 1.4 | 0.4 | 1.2 | 1.8 | 2.4 | 1.8 | 3.0 | 2.4 |
| MoCo (Best)† (He et al., 2019) | 66.9 | 77.6 | 60.8 | 39.0 | 39.7 | 24.2 | 61.5 | 72.8 | 52.4 | 34.7 | 33.8 | 19.7 |
| MiCE (Best, **Ours**) | **73.7** | **83.5** | **69.8** | **43.6** | **44.0** | **28.0** | **63.5** | **75.2** | **57.5** | **42.3** | **43.9** | **28.6** |

the parameters are updated using the exponential moving average (EMA) of the parameters in the student network. As it is often infeasible to tune the hyper-parameters with a validation dataset in real-world clustering tasks (Ghasedi Dizaji et al., 2017), we set both temperatures $\tau$ and $\kappa$ as 1.0. For a fair comparison, MoCo in the two-stage baseline also uses a ResNet-34 backbone. The network, hyper-parameters, and training settings are discussed detailedly in Appendix C.[3]

### 5.1 Main clustering results

**Comparison with existing deep clustering methods.** As shown in Tab. 2, MiCE outperforms the previous clustering approaches by a significant margin on all datasets. The comparison highlights the importance of exploring the discriminative representations and the semantic structures of the dataset.

**Comparison with the two-stage baseline.** Compared to the straightforward combination of MoCo and spherical $k$-means, MiCE explores the semantic structures of the dataset that improve the clustering performance. From Tab. 2, we can see that MiCE consistently outperforms the baseline in terms of the mean performance, which agrees with the analysis in Sec. 4.3. Specifically, regarding ACC, we improve upon the strong baseline by 8.7%, 2.7%, and 8.2% on CIFAR-10, CIFAR-100, and ImageNet-Dog, respectively. Taking the measurement variance into consideration, our performance overlaps with MoCo only on STL-10. We conjecture that the small data size may limit the performance as each expert learns from a subset of data. Nevertheless, the comparison manifests the significance of aligning the representation learning and clustering objectives in a unified framework. Together with the strong empirical evidence, we believe that MiCE points out a promising direction for future studies in clustering.

[3]We submit our source code (in PyTorch) to the system and will release it online after the blind review.

Table 3: Ablations of MiCE on the probabilistic model (left) and different ways of learning the gating and expert prototypes (right). Each row shows the cluster accuracy (ACC) on CIFAR-100 when applying the single change to MiCE. The assumptions are detailed in the Appendix B.

| | CIFAR-100 |
|---|---|
| (1) A3 ( $p(z_n\|\mathbf{x}_n) = 1/K$) | 40.7 |
| (2) A4 (Single output layer) | 39.3 |
| (3) A3 + A4 | 33.6 |
| (4) A5 (Discard class-level information) | 17.0 |
| MiCE (Ours) | 42.2 |

| | CIFAR-100 |
|---|---|
| (a) No analytical update on $\boldsymbol{\mu}$ in Eq. (11) | 21.3 |
| (b) No gradient update on $\boldsymbol{\mu}$ | 41.0 |
| (c) Initialize $\boldsymbol{\omega}$ with a uniform distribution | 41.0 |
| (d) Optimize $\boldsymbol{\omega}$ with gradient | 42.0 |
| MiCE (Ours) | 42.2 |

### 5.2 Ablation Studies

**Simplified model (Tab. 3 (left)).** We investigate the gating function and the unnormalized model to understand the contributions of different components. Using a simpler latent variable model often deteriorates the performance. (1) With a uniform prior, the experts would take extra efforts to become specialized in a set of images with shared semantics. (2 & 3) The teacher embedding $\mathbf{v}_{y_n}$ is pushed to be close to all expert prototypes at the same time. It may be difficult for the simplified expert to encode the latent semantics while being discriminative. (4) The performance drop shows that the class-level information is essential for the image embeddings to capture the semantic structures of the dataset, despite the learned representations are still discriminative between instances. Without the term, the learned embeddings are mixed up and scattered over the embedding space without a clear cluster structure.

**Prototypes update rules (Tab. 3 (right)).** We also conduct ablation studies to gain insights into the different ways of handling the gating and expert prototypes. We see that (a) without Eq. (11), we may be stuck in bad local optima. As mentioned in Sec. 4.2, a possible reason is that we are using the same learning rate for all network parameters and prototypes (Sculley, 2010; Yang et al., 2017), but tuning separate learning rates for each prototype is impractical for unsupervised clustering. Hence, we derive the analytical update to tackle the issue. As for (b), it shows that the current gradient update rule avoids the potential inconsistency between the expert prototypes and the teacher embeddings during the mini-batch training. Lastly, as discussed in Sec. 4.2, comparing to using (c) uniformly initiated gating prototypes projected onto the unit sphere, utilizing the means of MMD slightly improves performance. This also bypasses the potential learning rate issue that may appear in (d).

## 6 Conclusion

We present a principled probabilistic clustering method that conjoins the benefits of the discriminative representations learned by contrastive learning and the semantic structures introduced by the latent mixture model in a unified framework. With a divide-and-conquer principle, MiCE comprises an input-dependent gating function that distributes subtasks to one or a few specialized experts, and $K$ experts that discriminate the subset of images based on instance-level and class-level information. To address the challenging inference and learning problems, we present a scalable variant of Expectation-Maximization (EM) algorithm, which maximizes the ELBO and is free from any other loss or regularization terms. Moreover, we show that MoCo with spherical $k$-means, one of the two-stage baselines, is a special case of MiCE under various assumptions. Empirically, MiCE outperforms extensive prior methods and the strong two-stage baseline by a significant margin on several benchmarking datasets.

For future work, one may explore different learning pretext tasks that potentially fit the clustering task, other than the instance discrimination one. Besides, being able to obtain semantically meaningful clusters could be beneficial to weakly supervised settings (Zhou, 2018) where quality labels are scarce.

## References

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pp.

15535–15545, 2019.

Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. *NIPS*, pp. 153–160, 2006.

Deng Cai, Xiaofei He, Xuanhui Wang, Hujun Bao, and Jiawei Han. Locality preserving nonnegative matrix factorization. *IJCAI*, pp. 1010–1015, 2009.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 132–149, 2018.

Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep adaptive image clustering. In *Proceedings of the IEEE international conference on computer vision*, pp. 5879–5887, 2017.

Shlomo E Chazan, Sharon Gannot, and Jacob Goldberger. Deep clustering based on a mixture of autoencoders. pp. 1–6, 2019.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

LI Chongxuan, Max Welling, Jun Zhu, and Bo Zhang. Graphical generative adversarial networks. In *Advances in neural information processing systems*, pp. 6069–6080, 2018.

Adam Coates, Y. Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. *AISTATS*, pp. 215–223, 2011.

Luke Nicholas Darlow and Amos Storkey. Dhog: Deep hierarchical object grouping. *arXiv preprint arXiv:2003.08821*, 2020.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1): 1–22, 1977.

Jia Deng, Wei Dong, Richard Socher, Li-jia Li, Kai Li, and Fei-fei Li. Imagenet: A large-scale hierarchical image database. *CVPR*, pp. 248–255, 2009.

S. Inderjit Dhillon and S. Dharmendra Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, pp. 143–175, 2001.

Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.

Alexey Dosovitskiy, Philipp Fischer, Tobias Jost Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1734–1747, 2015.

Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *Proceedings of the IEEE international conference on computer vision*, pp. 5736–5745, 2017.

K Chidananda Gowda and G Krishna. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern recognition*, 10(2):105–112, 1978.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 297–304, 2010.

Philip Haeusser, Johannes Plapp, Vladimir Golkov, Elie Aljalbout, and Daniel Cremers. Associative deep clustering: Training a classification network with no labels. In *German Conference on Pattern Recognition*, pp. 18–32. Springer, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

E. Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, 2015.

Kurt Hornik, Ingo Feinerer, and Martin Kober. Spherical k-means clustering. *Journal of Statistical Software*, 050(1):1–22, 2012.

Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1558–1567. JMLR. org, 2017.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9865–9874, 2019.

Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. *arXiv preprint arXiv:1611.05148*, 2016.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. *arXiv preprint arXiv:1901.09005*, 2019.

Andreas Kopf, Vincent Fortuin, Vignesh Ram Somnath, and Manfred Claassen. Mixture-of-experts variational autoencoder for clustering and generating from similarity-based representations. *arXiv: Learning*, 2019.

Andreas Krause, Pietro Perona, and Ryan G Gomes. Discriminative clustering by regularized information maximization. In *Advances in neural information processing systems*, pp. 775–783, 2010.

A Krizhevsky and G Hinton. Learning multiple layers of features from tiny images. 2009.

Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020.

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

Sudipto Mukherjee, Himanshu Asnani, Eugene Lin, and Sreeram Kannan. Clustergan: Latent space clustering in generative adversarial networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 4610–4617, 2019.

Chuang Niu, Jun Zhang, Ge Wang, and Jimin Liang. Gatcluster: Self-supervised gaussian-attention network for image clustering. *arXiv preprint arXiv:2002.11863*, 2020.

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

Tianyu Pang, Chao Du, and Jun Zhu. Max-mahalanobis linear discriminant analysis networks. *arXiv preprint arXiv:1802.09308*, 2018.

Tianyu Pang, Kun Xu, Yinpeng Dong, Chao Du, Ning Chen, and Jun Zhu. Rethinking softmax cross-entropy loss for adversarial robustness. *arXiv preprint arXiv:1905.10626*, 2019.

Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and A. Alexei Efros. Context encoders: Feature learning by inpainting. *CVPR*, pp. 2536–2544, 2016.

alec radford, luke metz, and soumith chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *international conference on learning representations*, 2015.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pp. 91–99, 2015.

D. Sculley. Web-scale k-means clustering. *WWW*, pp. 1177–1178, 2010.

Guy Shiran and Daphna Weinshall. Multi-modal deep clustering: Unsupervised partitioning of images. *arXiv preprint arXiv:1912.02678*, 2019.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pp. 1195–1204, 2017.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *CoRR*, 2019.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine 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.

Jianlong Wu, Keyu Long, Fei Wang, Chen Qian, Cheng Li, Zhouchen Lin, and Hongbin Zha. Deep comprehensive correlation mining for image clustering. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 8150–8159, 2019.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3733–3742, 2018.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pp. 478–487, 2016.

Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3861–3870. JMLR. org, 2017.

Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5147–5156, 2016.

Linxiao Yang, Ngai-Man Cheung, Jiaying Li, and Jun Fang. Deep clustering by gaussian mixture variational autoencoders with graph embedding. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6440–6449, 2019.

Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6210–6219, 2019.

Lihi Zelnikmanor and Pietro Perona. Self-tuning spectral clustering. pp. 1601–1608, 2004.

Dejiao Zhang, Yifan Sun, Brian Eriksson, and Laura Balzano. Deep unsupervised clustering using mixture of autoencoders. *arXiv: Learning*, 2017.

Richard Zhang, Phillip Isola, and A. Alexei Efros. Colorful image colorization. *ECCV*, pp. 649–666, 2016.

junbo zhao, michael mathieu, ross goroshin, and yann lecun. Stacked what-where auto-encoders. *CoRR*, 2015.

Zhihua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1): 44–53, 2018.

# A INFERENCE AND LEARNING

## A.1 DERIVATION OF ELBO

$$
\begin{aligned}
&\log p(\mathbf{Y}|\mathbf{X};\boldsymbol{\theta},\boldsymbol{\psi},\boldsymbol{\mu}) \\
&= \mathbb{E}_{q(\mathbf{Z}|\mathbf{X},\mathbf{Y})}\left[\log \frac{p(\mathbf{Y},\mathbf{Z}|\mathbf{X};\boldsymbol{\theta},\boldsymbol{\psi},\boldsymbol{\mu})}{q(\mathbf{Z}|\mathbf{X},\mathbf{Y})}\right] + D_{\mathrm{KL}}(q(\mathbf{Z}|\mathbf{X},\mathbf{Y})\|p(\mathbf{Z}|\mathbf{X},\mathbf{Y};\boldsymbol{\theta},\boldsymbol{\psi},\boldsymbol{\mu}) \\
&\geq \mathcal{L}(\boldsymbol{\theta},\boldsymbol{\psi},\boldsymbol{\mu};\mathbf{X},\mathbf{Y}) \\
&:= \mathbb{E}_{q(\mathbf{Z}|\mathbf{X},\mathbf{Y})}\left[\log \frac{p(\mathbf{Y},\mathbf{Z}|\mathbf{X};\boldsymbol{\theta},\boldsymbol{\psi},\boldsymbol{\mu})}{q(\mathbf{Z}|\mathbf{X},\mathbf{Y})}\right] \\
&= \mathbb{E}_{q(\mathbf{Z}|\mathbf{X},\mathbf{Y})}\left[\log p(\mathbf{Y},\mathbf{Z}|\mathbf{X};\boldsymbol{\theta},\boldsymbol{\psi},\boldsymbol{\mu})\right] - \mathbb{E}_{q(\mathbf{Z}|\mathbf{X},\mathbf{Y})}\left[\log q(\mathbf{Z}|\mathbf{X},\mathbf{Y})\right] \\
&= \sum_{n=1}^{N}\sum_{k=1}^{K} q(k|\mathbf{x}_n,y_n)\left[\log p(k|\mathbf{x}_n;\psi) + \log p(y_n|\mathbf{x}_n,k;\boldsymbol{\theta},\boldsymbol{\mu}) - \log q(k|\mathbf{x}_n,y_n)\right].
\end{aligned}
$$

## A.2 DERIVATION OF EQ. (11)

**Assumption A1.** *Under the hard assignment, the variational distribution is given by:*

$$
\hat{q}(z_n|\mathbf{x}_n,y_n) = \begin{cases} 1, & \text{if } z_n = \operatorname*{argmax}_k \quad q(k|\mathbf{x}_n,y_n), \\ 0, & \text{otherwise.} \end{cases}
$$

**Assumption A2.** *For all possible inputs, the unnormalized model $\Phi(\mathbf{x}_n,y_n,z_n)$ is self-normalized similar to Wu et al. (2018), i.e., $Z(\mathbf{x}_n,z_n)$ equals to a constant $c$, such that there is no need to approximate the normalization constant as well.*

Here, we derive the analytical update of the expert prototypes based on ELBO. Under the Assumptions A1 and A2, we can update $\boldsymbol{\mu}_k$ by solving the following problem:

$$
\begin{aligned}
\boldsymbol{\mu}_k &\leftarrow \operatorname*{argmax}_{\boldsymbol{\mu}_k} \mathcal{L}(\boldsymbol{\mu}_k;\mathbf{X},\mathbf{Y}) \\
&= \operatorname*{argmax}_{\boldsymbol{\mu}_k} \sum_{n=1}^{N} \hat{q}(k|\mathbf{x}_n,y_n)\left[\log p(k|\mathbf{x}_n) + \log p(y_n|\mathbf{x}_n,k;\boldsymbol{\mu}_k) - \log \hat{q}(k|\mathbf{x}_n,y_n)\right] \\
&= \operatorname*{argmax}_{\boldsymbol{\mu}_k} \sum_{n=1}^{N} \hat{q}(k|\mathbf{x}_n,y_n)\left[\log p(y_n|\mathbf{x}_n,k;\boldsymbol{\mu}_k)\right] \\
&= \operatorname*{argmax}_{\boldsymbol{\mu}_k} \sum_{n=1}^{N} \hat{q}(k|\mathbf{x}_n,y_n)\left[\log \frac{\Phi(\mathbf{x}_n,y_n,k)}{Z(\mathbf{x}_n,k)}\right] \\
&= \operatorname*{argmax}_{\boldsymbol{\mu}_k} \sum_{n=1}^{N} \hat{q}(k|\mathbf{x}_n,y_n)\left[\log \Phi(\mathbf{x}_n,y_n,k)\right] \qquad \text{(Assumption A2)} \\
&= \operatorname*{argmax}_{\boldsymbol{\mu}_k} \sum_{n=1}^{N} \hat{q}(k|\mathbf{x}_n,y_n)\left[\log \exp\left(\mathbf{v}_{y_n,k}^{\top}\mathbf{f}_{n,k}/\tau + \mathbf{v}_{y_n,k}^{\top}\boldsymbol{\mu}_k/\tau\right)\right] \\
&= \operatorname*{argmax}_{\boldsymbol{\mu}_k} \sum_{n=1}^{N} \hat{q}(k|\mathbf{x}_n,y_n)\left(\mathbf{v}_{y_n,k}^{\top}\mathbf{f}_{n,k}/\tau + \mathbf{v}_{y_n,k}^{\top}\boldsymbol{\mu}_k/\tau\right) \\
&= \operatorname*{argmax}_{\boldsymbol{\mu}_k} \sum_{n=1}^{N} \hat{q}(k|\mathbf{x}_n,y_n)\mathbf{v}_{y_n,k}^{\top}\boldsymbol{\mu}_k/\tau,
\end{aligned}
$$

subjected to the constraint that $\|\boldsymbol{\mu}_k\| = 1$. Introducing a Lagrange multiplier $\lambda$, we then solve the Lagrangian of the objective function:

$$\underset{\boldsymbol{\mu}_k}{\operatorname{argmax}} \quad \lambda(1 - \boldsymbol{\mu}_k^\top \boldsymbol{\mu}_k) + \sum_{n=1}^{N} \hat{q}(k|\mathbf{x}_n, y_n) \mathbf{v}_{y_n,k}^\top \boldsymbol{\mu}_k / \tau \tag{12}$$

We can get the estimates of $\boldsymbol{\mu}_k$ and $\lambda$ by differentiating the Lagrangian with regards to them and setting the derivatives to zero. To be specific, for $\boldsymbol{\mu}_k$, we have:

$$\begin{aligned}
\nabla_{\boldsymbol{\mu}_k} \quad & \lambda(1 - \boldsymbol{\mu}_k^\top \boldsymbol{\mu}_k) + \sum_{n=1}^{N} \hat{q}(k|\mathbf{x}_n, y_n) \mathbf{v}_{y_n,k}^\top \boldsymbol{\mu}_k / \tau \\
&= -2\lambda \boldsymbol{\mu}_k + \sum_{n=1}^{N} \hat{q}(k|\mathbf{x}_n, y_n) \mathbf{v}_{y_n,k} / \tau = 0,
\end{aligned}$$

such that

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} \hat{q}(k|\mathbf{x}_n, y_n) \mathbf{v}_{y_n,k}}{2\lambda\tau} := \frac{\hat{\boldsymbol{\mu}}_k}{2\lambda\tau}, \tag{13}$$

where we define $\hat{\boldsymbol{\mu}}_k := \sum_{n=1}^{N} \hat{q}(k|\mathbf{x}_n, y_n) \mathbf{v}_{y_n,k} := \sum_{n:\hat{z}_n=k} \mathbf{v}_{y_n,k}$ for simplicity.

Similarly, by differentiating with respect to $\lambda$, we get:

$$\boldsymbol{\mu}_k{}^\top \boldsymbol{\mu}_k = 1. \tag{14}$$

Substituting Eq. (13) in Eq. (14), we have:

$$1 = \frac{1}{4\tau^2\lambda^2} \hat{\boldsymbol{\mu}}_k^\top \hat{\boldsymbol{\mu}}_k = \frac{1}{4\tau^2\lambda^2} \|\hat{\boldsymbol{\mu}}_k\|^2.$$

Therefore, we obtain the analytical solutions:

$$\begin{aligned}
\lambda &= \frac{\|\hat{\boldsymbol{\mu}}_k\|}{2\tau}, \\
\boldsymbol{\mu}_k &= \frac{\hat{\boldsymbol{\mu}}_k}{2\tau\lambda} = \frac{\hat{\boldsymbol{\mu}}_k}{\|\hat{\boldsymbol{\mu}}_k\|}.
\end{aligned}$$

The last expression is essentially the update for the prototypes of the experts, as presented in Eq. (11) in the main text.

# B Relations to the two-stage baseline

## B.1 Contrastive learning

**Assumption A3.** *The gating temperature $\kappa \to \infty$ such that for all $k$, the prior*

$$p(z_n|\mathbf{x}_n) = \frac{1}{K}, \quad \forall n.$$

**Assumption A4.** *There is only a single output layer for both teacher network $f_{\boldsymbol{\theta}}$ and student network $f_{\boldsymbol{\theta}'}$ respectively. The resulting embeddings are used across $K$ expert, to be specific, for all possible cases,*

$$\Phi(\mathbf{x}_n, y_n, k) = \exp\left(\mathbf{v}_{y_n}^\top (\mathbf{f}_n + \boldsymbol{\mu}_k)/\tau\right),$$

*where $\mathbf{f}_n = f_{\boldsymbol{\theta}}(\mathbf{x}_n) \in \mathbb{R}^d$, $\mathbf{v}_{y_n}^\top = f_{\boldsymbol{\theta}'}(\mathbf{x}_n) \in \mathbb{R}^d$.*

**Assumption A5.** *The unnormalized model $\Phi(\cdot)$ considers only the instance-level information such that for all possible cases,*

$$\Phi(\mathbf{x}_n, y_n, z_n) = \exp\left(\mathbf{v}_{y_n,z_n}^\top \mathbf{f}_{n,z_n}/\tau\right).$$

**Theorem 1.** *If Assumptions A3-A5 holds, the overall output of MiCE become*

$$p(y_n|\mathbf{x}_n) = \frac{\exp(\mathbf{v}_{y_n}^\top \mathbf{f}_n/\tau)}{\sum_{i=1}^{N} \exp(\mathbf{v}_i^\top \mathbf{f}_n/\tau)}.$$

*Proof.* Since we the expert

$$p(y_n|\mathbf{x}_n, z_n) = \frac{\Phi(\mathbf{x}_n, y_n, z_n)}{Z(\mathbf{x}_n, z_n)} = \frac{\exp\left(\mathbf{v}_{y_n}^\top(\mathbf{f}_n + \boldsymbol{\mu}_{z_n})/\tau\right)}{\sum_{i=1}^N \exp\left(\mathbf{v}_i^\top(\mathbf{f}_n + \boldsymbol{\mu}_{z_n})/\tau\right)} \quad \text{(Assumption A4)}$$

$$= \frac{\exp(\mathbf{v}_{y_n}^\top \mathbf{f}_n/\tau)}{\sum_{i=1}^N \exp(\mathbf{v}_i^\top \mathbf{f}_n/\tau)}. \quad \text{(Assumption A5)}$$

The overall output of MiCE is then

$$\begin{aligned} p(y_n|\mathbf{x}_n) &= \sum_{k=1}^K p(z_n = k|\mathbf{x}_n) p(y_n|\mathbf{x}_n, z_n = k) \\ &= \sum_{k=1}^K \frac{1}{K} p(y_n|\mathbf{x}_n, z_n = k) \quad \text{(Assumption A3)} \\ &= \frac{\exp(\mathbf{v}_{y_n}^\top \mathbf{f}_n/\tau)}{\sum_{i=1}^N \exp(\mathbf{v}_i^\top \mathbf{f}_n/\tau)}. \end{aligned}$$

□

The simplified model shown in Theorem 1 is essentially the non-parametric Softmax classifier used by MoCo and InstDisc, which is also related to some recent contrastive learning methods (Ye et al., 2019; Bachman et al., 2019). Note that InstDisc adopts a slightly different implementation in which the teacher network is identical to the student network and the loss function is based on NCE (Gutmann & Hyvärinen, 2010). For detailed comparisons, please refer to He et al. (2019).

**Lemma 1.** *If Assumptions A3-A5 hold, the posterior is uniformly distributed.*

*Proof.*

$$\begin{aligned} p(z_n|\mathbf{x}_n, y_n) &= \frac{p(z_n|\mathbf{x}_n)\Phi\left(\mathbf{x}_n, y_n, z_n\right)/Z(\mathbf{x}_n, z_n)}{\sum_{k=1}^K p(k|\mathbf{x}_n)\Phi\left(\mathbf{x}_n, y_n, k\right)/Z(\mathbf{x}_n, k)} \\ &= \frac{\Phi\left(\mathbf{x}_n, y_n, z_n\right)/Z(\mathbf{x}_n, z_n)}{\sum_{k=1}^K \Phi\left(\mathbf{x}_n, y_n, k\right)/Z(\mathbf{x}_n, k)} \quad \text{(Assumption A3)} \\ &= \frac{\frac{\exp(\mathbf{v}_{y_n}^\top \mathbf{f}_n/\tau)}{\sum_{i=1}^N \exp(\mathbf{v}_i^\top \mathbf{f}_n/\tau)}}{\sum_{k=1}^K \frac{\exp(\mathbf{v}_{y_n}^\top \mathbf{f}_n/\tau)}{\sum_{i=1}^N \exp(\mathbf{v}_i^\top \mathbf{f}_n/\tau)}} \quad \text{(Theorem 1)} \\ &= \frac{1}{K}. \end{aligned}$$

□

**Assumption A6.** *The normalization constant is computationally tractable such that we can have the variational distribution being identical to the posterior distribution.*

**Theorem 2.** *Given that A3-A6 hold,* $p(y_n|\mathbf{x}_n) = \exp(\mathbf{v}_{y_n}^\top \mathbf{f}_n/\tau)/\sum_{i=1}^N \exp(\mathbf{v}_i^\top \mathbf{f}_n/\tau)$, *and the tractable version of ELBO is identical to the form of InfoNCE (Oord et al., 2018) loss used by MoCo.*

*Proof.*

$$\begin{aligned} \widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\mu}; \mathbf{x}_n, y_n) &= \mathbb{E}_{q(z_n|\mathbf{x}_n, y_n)}[\log \frac{\Phi\left(\mathbf{x}_n, y_n, z_n\right)}{\hat{Z}(\mathbf{x}_n, z_n)}] - D_{\mathrm{KL}}(q(z_n|\mathbf{x}_n, y_n) \| p(z_n|\mathbf{x}_n) \\ &= \log \frac{\Phi\left(\mathbf{x}_n, y_n, z_n\right)}{\hat{Z}(\mathbf{x}_n, z_n)} \quad \text{(Lemma 1)} \\ &= \log \frac{\exp\left(\mathbf{v}_{y_n}^\top \mathbf{f}_n/\tau\right)}{\exp\left(\mathbf{v}_{y_n}^\top \mathbf{f}_n/\tau\right) + \sum_{i=1}^\nu \exp\left(\mathbf{q}_i^\top \mathbf{f}_n/\tau\right)}. \quad \text{(Theorem 1)} \end{aligned}$$

□

According to Theorem 1 and 2, we see that MoCo can be viewed as a special case of the proposed MiCE. In other words, they are able to learn the same representations under the same experimental setting if the above assumptions are made.

### B.2 SPHERICAL $k$-MEANS

**Theorem 3.** *If Assumptions A1-A4 hold, the analytical update on the expert prototypes is equivalent to a single-iteration spherical $k$-means algorithm on the teacher embeddings.*

*Proof.* Under the assumptions, the variational distribution reduces to

$$\begin{aligned}
q(z_n|\mathbf{x}_n, y_n) &= \frac{\Phi\left(\mathbf{x}_n, y_n, z_n\right)}{\sum_{k=1}^{K} \Phi\left(\mathbf{x}_n, y_n, k\right)} && \text{(Assumptions A2 and A3)} \\
&= \frac{\exp\left(\mathbf{v}_{y_{n,z_n}}^\top \mathbf{f}_{n,z_n}/\tau + \mathbf{v}_{y_{n,z_n}}^\top \boldsymbol{\mu}_{z_n}/\tau\right)}{\sum_{k=1}^{K} \exp\left(\mathbf{v}_{y_{n,k}}^\top \mathbf{f}_{n,k}/\tau + \mathbf{v}_{y_{n,k}}^\top \boldsymbol{\mu}_{k}/\tau\right)} \\
&= \frac{\exp\left(\mathbf{v}_{y_n}^\top \mathbf{f}_{n}/\tau + \mathbf{v}_{y_n}^\top \boldsymbol{\mu}_{z_n}/\tau\right)}{\sum_{k=1}^{K} \exp\left(\mathbf{v}_{y_n}^\top \mathbf{f}_{n}/\tau + \mathbf{v}_{y_n}^\top \boldsymbol{\mu}_{k}/\tau\right)} && \text{(Assumption A4)} \\
&= \frac{\exp\left(\mathbf{v}_{y_n}^\top \boldsymbol{\mu}_{z_n}/\tau\right)}{\sum_{k=1}^{K} \exp\left(\mathbf{v}_{y_n}^\top \boldsymbol{\mu}_{k}/\tau\right)},
\end{aligned}$$

such that the hard cluster assignment is determined bt the cosine similarity between the teacher embedding and the expert prototypes:

$$\hat{q}(z_n|\mathbf{x}_n, y_n) = \begin{cases} 1, & \text{if } z_n = \operatorname*{argmax}_k \quad q(k|\mathbf{x}_n, y_n) = \operatorname*{argmax}_k \quad \mathbf{v}_{y_n}^\top \boldsymbol{\mu}_k, \\ 0, & \text{otherwise.} \end{cases}$$

With the simplified expert model, we follow the previous discussion on Eq. (12) and get the Lagrangian of the objective function:

$$\operatorname*{argmax}_{\boldsymbol{\mu}_k} \quad \lambda(1 - \boldsymbol{\mu}_k^\top \boldsymbol{\mu}_k) + \sum_{n=1}^{N} \hat{q}(k|\mathbf{x}_n, y_n) \mathbf{v}_{y_n}^\top \boldsymbol{\mu}_k/\tau.$$

The analytical solution is therefore:

$$\begin{aligned}
\hat{\boldsymbol{\mu}}_k &:= \sum_{n=1}^{N} \hat{q}(k|\mathbf{x}_n, y_n) \mathbf{v}_{y_n}, \\
\boldsymbol{\mu}_k &= \frac{\hat{\boldsymbol{\mu}}_k}{\|\hat{\boldsymbol{\mu}}_k\|},
\end{aligned}$$

where the cluster assignment step and prototype update rule are the same as the spherical $k$-means.

□

## C EXPERIEMENT SETTINGS

Regarding the network architecture, MiCE mainly use a ResNet-34 (He et al., 2016) as the backbone. For the gating network, the output layer is replaced by a single fully connected layer that generates a $\ell_2$-normalized embeddings in $\mathbb{R}^{128}$. As for the student network, it uses the same ResNet-34 backbone as the gating network and includes $K$ more fully connected layers which map the images to $K$ embeddings. Therefore, the parameters of student and gating networks are shared except for the output layers. The teacher network $f_{\boldsymbol{\theta}'}$ is the exponential moving average (EMA) version of the

student network $f_{\boldsymbol{\theta}}$, which stabilizes the learning process (He et al., 2019; Tarvainen & Valpola, 2017). The update rule follows $\boldsymbol{\theta}' \leftarrow m\boldsymbol{\theta}' + (1-m)\boldsymbol{\theta}$ with $m \in [0, 1)$ being the smoothing coefficient. In practice, we let $m = 0.999$ following MoCo. Since the images of CIFAR-10 and CIFAR-100 are smaller than ImageNet images, following (Chen et al., 2020), we replace the first 7x7 Conv of stride 2 with a 3x3 Conv of stride 1 for all experiments on CIFAR-10 and CIFAR-100. The first max-pooling operation is removed as well. For fair comparisons, MoCo also uses a ResNet-34 with a $128$-dimensional output and follows the same hyper-parameter settings as MiCE.

Besides, the queue size $\nu$ is set to $12800$ for STL-10 because of the smaller data size and $16384$ for the other three datasets. The data augmentation follows MoCo closely. Before passing into any of the embedding networks, images are randomly resized and cropped to the same size, followed by random gray-scale, random color jittering, and random horizontal flip. Given that the validation dataset is often unavailable in the real-world scenarios, both temperatures $\tau$ and $\kappa$ are set as $1.0$.

In terms of the optimization details, we use stochastic gradient descent (SGD) as our optimizer on the negative ELBO. We set the SGD weight decay as $0.0001$ and the SGD momentum as $0.9$ (He et al., 2019). The learning rate is initiated as $1.0$ and is multiplied by $0.1$ at three different epochs. For different datasets, the number of training epochs is different to accommodate the data size to have a similar and reasonable training time. For CIFAR-10/100, we train for $1000$ epochs in total and multiply the learning rate by $0.1$ at $480$, $640$, and $800$ epochs. For STL-10, the total epochs are $6000$ and the learning rate is multiplied by $0.1$ at $3000$, $4000$, and $5000$ epochs. Lastly, for ImageNet-Dog, the total epochs are $3000$ and the learning rate is multiplied by $0.1$ at $1500$, $2000$, and $2500$ epochs. Also, the learning rate for expert prototypes is the same as the one for network parameters. All the experiments are trained on a single GPU.