# Infinite Mixture Prototypes for Few-Shot Learning

**Kelsey R. Allen** [1] **Evan Shelhamer** [*] [2] **Hanul Shin** [*] [1] **Joshua B. Tenenbaum** [1]

## Abstract

We propose infinite mixture prototypes to adaptively represent both simple and complex data distributions for few-shot learning. Infinite mixture prototypes combine deep representation learning with Bayesian nonparametrics, representing each class by a set of clusters, unlike existing prototypical methods that represent each class by a single cluster. By inferring the number of clusters, infinite mixture prototypes interpolate between nearest neighbor and prototypical representations in a learned feature space, which improves accuracy and robustness in the few-shot regime. We show the importance of adaptive capacity for capturing complex data distributions such as super-classes (like alphabets in character recognition), with 10-25% absolute accuracy improvements over prototypical networks, while still maintaining or improving accuracy on standard few-shot learning benchmarks. By clustering labeled and unlabeled data with the same rule, infinite mixture prototypes achieve state-of-the-art semi-supervised accuracy, and can perform purely unsupervised clustering, unlike existing fully- and semi-supervised prototypical methods.

## 1. Introduction

Few-shot classification is the problem of learning to recognize new classes from only a few examples of each class (Lake et al., 2015; Fei-Fei et al., 2006; Miller et al., 2000). This requires careful attention to generalization, since overfitting or underfitting to the sparsely available data is more likely. Nonparametric methods are well suited to this task, as they can model decision boundaries that more closely reflect the data distribution by using the data itself.

Two popular classes of nonparametric methods are nearest neighbor methods and prototypical methods. Nearest neighbor methods represent a class by storing all of its examples, and are high-capacity models that can capture complex distributions. Prototypical methods, such as Gaussian mixture models, represent a class by the mean of its examples, and are low-capacity models that can robustly fit simple distributions. Neighbors and prototypes are thus two ends of a spectrum from complex to simple decision boundaries, and the choice of which to apply generally requires knowledge about the complexity of the distribution.

Recent work has looked at combining nonparametric methods with deep representation learning, such that the learned features are directly optimized for nonparametric inference (Snell et al., 2017; Vinyals et al., 2016). However, these approaches have fixed model capacity: the complexity of the task is handled by the representation learning, rather than the inference procedure, and cannot adapt.

Here we investigate adaptively modulating model capacity during inference. This is especially useful in few-shot learning where the complexity of individual tasks can differ, and so should not necessarily be handled by the representation learning component alone. Several approaches exist to tackle this, such as choosing $k$ for $k$-nearest neighbours, selecting the number of mixture components for Gaussian mixture models, or adjusting the bandwidth (Jones et al., 1996) for kernel density estimation (Parzen, 1962).

Infinite mixture modeling (Hjort et al., 2010) represents one way of unifying these approaches for adaptively setting capacity. By inferring the number of mixture components for a given class from the data, it is possible to span the spectrum from nearest neighbors to prototypical representations.

This is particularly important in few-shot learning, where both underfitting and overfitting are common problems, because current models are fixed in their capacity.

To give an example, consider the problems of character and alphabet recognition. Recognizing characters is fairly straightforward: each character looks alike, and can be represented as a single prototype (a uni-modal Gaussian distribution). Recognizing alphabets is more complex: the uni-modal distribution assumption could be violated, and a multi-modal approach could better capture the complexity

[*] Equal contribution [1] Department of Brain and Cognitive Sciences, Center for Brains, Minds, and Machines (CBMM), CSAIL, MIT, Cambridge, MA [2] Computer Science, UC Berkeley, Berkeley, CA. Correspondence to: Kelsey R. Allen <krallen@mit.edu>.

*Proceedings of the $36^{th}$ International Conference on Machine Learning*, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

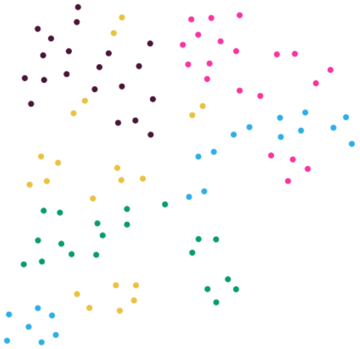

*Figure 1.* t-SNE visualization of the deep embedding from a prototypical network trained for alphabet recognition on Omniglot. Each point is a character colored by its alphabet label. The data distribution of each class is clearly not uni-modal, in violation of the modeling assumption for existing prototypical methods, causing errors. Our infinite mixture prototypes represent each class by a *set* of clusters, and infer their number, to better fit such distributions.

of the distribution. Figure 1 shows a prototypical network embedding for alphabets with this very issue. Even though the embedding was optimized for uni-modality, the unimodal assumption is not guaranteed on held-out data.

We therefore propose infinite mixture prototypes (IMP) to represent a class as a set of clusters, with the number of clusters determined directly from the data. IMP learns a deep embedding while also adapting the model capacity based on the complexity of the embedded data. As a further benefit, the infinite mixture modeling approach can naturally incorporate unlabeled data. We accordingly extend IMP to semi-supervised few-shot learning, and even to fully-unsupervised clustering inference.

An alternative approach to IMP would be to learn a parametric model. The decision boundary would then be linear in the embedding, which is more complex than uni-modal prototypes, but less complex than nearest neighbors. However, it may not be possible to find a *nonlinear* embedding that yields an effective *linear* decision boundary. In practice, either a parametric method or uni-modal mixture model is sensitive to the choice of model capacity, and may not successfully learn complex classes such as Omniglot (Lake et al., 2015) alphabets. Instead, a higher-capacity nonparametric method like nearest neighbors can work better. For simpler classes such as characters, a parametric model from a meta-learned initialization (Finn et al., 2017) or a prototypical network that assumes uni-modal data (Snell et al., 2017) suffice. Infinite mixture prototypes span these extremes, learning to adapt to both simple and complex classes.

In this paper, we extend prototypical networks from unimodal to multi-modal clustering through infinite mixture modeling to give 25% improvement in accuracy for alphabet recognition (complex classes) while preserving accuracy on character recognition (simple classes) on Omniglot. In the semi-supervised setting infinite mixture prototypes are more accurate than semi-supervised prototypical networks. Infinite mixture modeling also allows for fully unsupervised clustering unlike existing prototypical methods. We demonstrate that the DP-means algorithm (Kulis & Jordan, 2012) is suitable for instantiating new clusters and that our novel extensions are necessary for best results in the few-shot regime. By end-to-end learning with infinite mixture modeling, IMP adapts its model capacity to simple or complex data distributions, shown by equal or better accuracy compared to neighbors and uni-modal prototypes in all experiments.

## 2. Background

For nonparametric representation learning methods, the model parameters are for the embedding function $h_\phi : \mathbb{R}^D \to \mathbb{R}^M$ that map an input point $x$ into a feature. The embedding of point $x$ is the $M$-dimensional feature vector from the embedding function. In deep models the parameters $\phi$ are the weights of a deep network, and the embedding is the output of the last layer of this network. (Such methods are still nonparametric because they represent decisions by the embedding of the data, and not parameters alone.)

### 2.1. Few-shot Classification

In few-shot classification we are given a *support* set $S = \{(x_1, y_1), \ldots, (x_K, y_K)\}$ of $K$ labeled points and a *query* set $Q = \{(x'_1, y'_1), \ldots, (x'_{K'}, y'_{K'})\}$ of $K'$ labeled points where each $x_i, x'_i \in \mathbb{R}^D$ is a $D$-dimensional feature vector and $y_i, y'_i \in \{1, \ldots, N\}$ is the corresponding label. In the semi-supervised setting, $y_i$ may not be provided for every point $x_i$. The support set is for learning while the query set is for inference: the few-shot classification problem is to recognize the class of the queries given the labeled supports.

Few-shot classification is commonly learned by constructing few-shot tasks from a large dataset and optimizing the model parameters on these tasks. Each task, comprised of the support and query sets, is called an *episode*. Episodes are drawn from a dataset by randomly sampling a subset of classes, sampling points from these classes, and then partitioning the points into supports and queries. The number of classes in the support is referred to as the “way” of the episode, and the number of examples of each class is referred to as the “shot” of the episode. Episodic optimization (Vinyals et al., 2016) iteratively trains the model by making one episode and taking one update at a time. The update to the model parameters is defined by the task loss, which for classification could be the softmax cross-entropy loss.

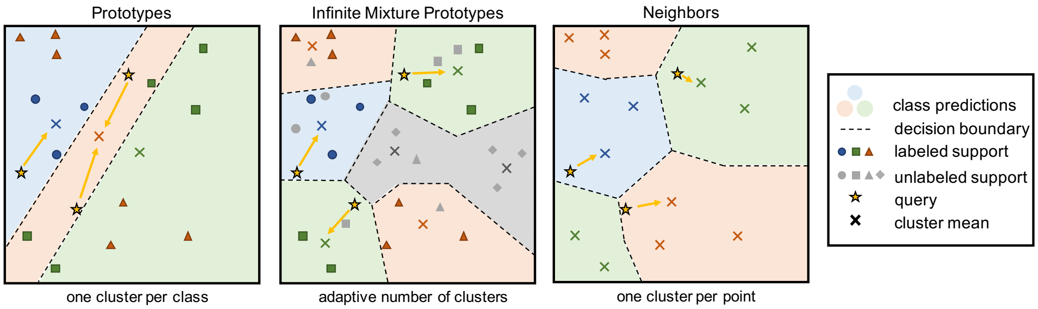

*Figure 2.* Our infinite mixture prototypes (IMP) method combines deep representation learning with nonparametric clustering to represent each class by a set of clusters in a learned feature embedding. The number of clusters is inferred from the data to adjust modeling capacity. IMP is optimized end-to-end to cluster labeled and unlabeled data into multi-modal prototypes.

### 2.2. Neighbors & Prototypes

**Neighbors** Nearest neighbors classification (Cover & Hart, 1967) assigns each query the label of the closest support. Neighbor methods are extremely simple but remarkably effective, because the classification is local and so they can fit complex data distributions. This generality comes at a computational cost, as the entire training set has to be stored and searched for inference. More fundamentally, there is a modeling issue: how should the distance metric to determine the "nearest" neighbor be defined?

Neighborhood component analysis (Goldberger et al., 2004) learns the distance metric by defining stochastic neighbors to make the classification decision differentiable. The metric is parameterized as a linear embedding $A$, and the probability of a point $x_i$ having neighbor $x_j$ is given by the softmax over Euclidean distances in the embedding:

$$p_{ij} = \frac{\exp(\|Ax_i - Ax_j\|^2)}{\sum_{k\neq j}\exp(\|Ax_i - Ax_k\|^2)}. \tag{1}$$

The probability that a point $x_i$ is in class $n$ is given by the sum of probabilities of neighbors in the class:

$$p_A(y = n \,|\, x_i) = \sum_{j:y_j=n} p_{ij}. \tag{2}$$

Stochastic neighbors naturally extend to a non-linear embedding trained by episodic optimization. Deep nearest neighbors classification therefore serves as a high-capacity nonparametric method for few-shot learning.

**Prototypes** Prototypical networks (Snell et al., 2017) form *prototypes* as the mean of the embedded support points in each class:

$$\mu_n = \frac{1}{|S_n|}\sum_{(x_i,y_i)\in S_n} h_\phi(x_i), \tag{3}$$

with $S_n$ denoting the set of support points in class $n$. Paired with a distance $d(x_i, x_j)$, the prototypes classify a query point $x'$ by the softmax over distances to the prototypes:

$$p_\phi(y' = n \,|\, x') = \frac{\exp(-d(h_\phi(x'), \mu_n))}{\sum_{n'}\exp(-d(h_\phi(x'), \mu_{n'}))}. \tag{4}$$

For the standard choice of the Euclidean distance function, the prototypes are equivalent to a Gaussian mixture model in the embedding with an identity covariance matrix.

$\phi$ is optimized by minimizing the negative log-probability of the true class of each query point by stochastic gradient descent over episodes. Prototypical networks therefore learn to create *uni-modal* class distributions for *fully-labeled* supports by representing each class by one cluster.

### 2.3. Infinite Mixture Modeling

Infinite mixture models (Hjort et al., 2010) do not require the number of mixture components to be known and finite. Instead, the number of components is inferred from data through Bayesian nonparametric methods (West et al., 1994; Rasmussen, 2000). In this way infinite mixture models adapt their capacity to steer between overfitting with high capacity and underfitting with low capacity.

The advantage of adaptivity is countered by the implementation and computational difficulties of Gibbs sampling and variational inference for infinite mixtures. To ease these issues, DP-means (Kulis & Jordan, 2012) is a deterministic, hard clustering algorithm derived via Bayesian nonparametrics for the Dirichlet process. DP-means iterates over the data points, computing each point's minimum distance to all existing cluster means. If this distance is greater than a threshold $\lambda$, a new cluster is created with mean equal to the point. It optimizes a $k$-means-like objective for reconstruction error plus a penalty for making clusters.

$\lambda$, the distance threshold for creating a new cluster, is the sole hyperparameter for the algorithm. In deriving DP-means, Kulis & Jordan (2012) relate $\alpha$, the concentration parameter for the Chinese restaurant process (CRP) (Aldous, 1985), to $\lambda$:

$$\lambda = 2\sigma\log\left(\frac{\alpha}{(1+\frac{\rho}{\sigma})^{d/2}}\right) \tag{5}$$

where $\rho$ is a measure of the standard deviation for the base distribution from which clusters are assumed to be drawn in the CRP. They then derive DP-means by connection to a Gibbs sampling procedure in the limit as $\sigma$ approaches 0.

## 3. Infinite Mixture Prototypes (IMP)

Our infinite mixture prototypes (IMP) method pursues two approaches for adapting capacity: learning cluster variance to scale assignments, and multi-modal clustering to interpolate between neighbor and prototypical representations. This capability lets our model adapt its capacity to avoid underfitting, unlike existing prototypical models with fixed capacity. Figure 2 gives a schematic view of our multi-modal representation and how it differs from existing prototype and neighbor representations. Algorithm 1 expresses infinite mixture prototypes inference in pseudocode.

Within an episode, we initially cluster the support into class-wise means. Inference proceeds by iterating through all support points and computing their minimum distance to all existing clusters. If this distance exceeds a threshold $\lambda$, a new cluster is made with mean equal to that point. IMP then updates soft cluster assignments $z_{i,c}$ as the normalized Gaussian density for cluster membership. Finally, cluster means $\mu_c$ are computed by the weighted mean of their members. Since each class can have multiple clusters, we classify a query point $x'$ by the softmax over distances to the closest cluster in each class $n$:

$$p_\phi(y' = n \,|\, x') = \frac{\exp(-d(h_\phi(x'),\ \mu_{c^*_n}))}{\sum_{n'} \exp(-d(h_\phi(x'),\ \mu_{c^*_{n'}}))} \tag{6}$$

with $c^*_n = \arg\min_{c:l_c=n} d(h_\phi(x'), \mu_c)$ indexing the clusters, where each cluster $c$ has label $l_c$.

IMP optimizes the embedding parameters $\phi$ and cluster variances $\sigma$ by stochastic gradient descent across episodes, using the loss $J$ (equation 7).

### 3.1. Adapting capacity by learning cluster variance $\sigma$

We learn the cluster variance $\sigma$ to scale the assignment of support points to clusters. When $\sigma$ is small, the effective distance is large and the closest points dominate, and when $\sigma$ is large, the effective distance is small so farther points are more included. $\sigma$ is differentiable, and therefore learned jointly with the embedding parameters $\phi$. In practice, learning $\sigma$ can improve the accuracy of prototypical networks, which we demonstrate by ablation in Table 1. For IMP, $\sigma$ has a further role in creating new clusters.

### 3.2. Adapting capacity by multi-modal clustering

To create multi-modal prototypes, we extend the clustering algorithm DP-means (Kulis & Jordan, 2012) for compatibility with classification and end-to-end optimization. For classification, we distinguish labeled and unlabeled clusters, and incorporate labels into the point-cluster distance calculation. For end-to-end optimization, we soften cluster assignment, propose a scheme to select $\lambda$, and mask the classification loss to encourage multi-modality.

**Algorithm 1** IMP: support prototypes and query inference

```
Require: supports $(x_1, y_1)..., (x_K, y_K)$ and queries $x'_1, ..., x'_{K'}$
Return: clusters $(\mu_c, l_c, \sigma_c)$ and query classifications $p(y'|x')$
  1. Init. each cluster $\mu_c$ with label $l_c$ and $\sigma_c = \sigma_l$ as class-
     wise means of the supports, and $C$ as the number of classes
  2. Estimate $\lambda$ as in Equation 5
  3. Infer the number of clusters
  for each point $x_i$ do
    for $c$ in $\{1, ..., C\}$ do
      $d_{i,c} = \begin{cases} \|h_\phi(x_i) - \mu_c\|^2 & \text{if } (x_i \text{ is labeled and } l_c = y_i) \\ & \text{or } x_i \text{ is unlabeled} \\ +\infty & \text{otherwise} \end{cases}$
    end for
    If $\min_c d_{ic} > \lambda$: set $C = C + 1$, $\mu_C = h_\phi(x_i)$, $l_C = y_i$,
    $\sigma_C = \{\sigma_l$ if $x_i$ labeled, $\sigma_u$ otherwise$\}$.
  end for
  4. Assign supports to clusters by $z_{i,c} = \frac{\mathcal{N}(h_\phi(x_i);\mu_c,\sigma_c)}{\sum_c \mathcal{N}(h_\phi(x_i);\mu_c,\sigma_c)}$
  5. For each cluster $c$, compute mean $\mu_c = \frac{\sum_i z_{i,c} h_\phi(x_i)}{\sum_i z_{i,c}}$
  6. Classify queries by Equation 6
```

**Indirect optimization of $\lambda$** While $\lambda$ is non-differentiable, we propose an indirect optimization of the effective threshold for creating a new cluster. Based on Equation 5, $\lambda$ depends on the concentration hyperparameter $\alpha$, a measure of standard deviation in the prior $\rho$, and the cluster variance $\sigma$. $\alpha$ remains a hyperparameter, but with lessened effect. We estimate $\rho$ as the variance between prototypes in each episode. As noted, $\sigma$ is differentiable, so we learn it.

We model separate variances for labeled and unlabeled clusters, $\sigma_l$ and $\sigma_u$ respectively, in order to capture differences in uncertainty between labeled and unlabeled data. In the fully-supervised setting, $\lambda$ is estimated from $\sigma_l$, and in the semi-supervised setting $\lambda$ is estimated from the mean of $\sigma_l$ and $\sigma_u$. In summary, learning the cluster variances $\sigma$ affects IMP by scaling the distances between points and clusters, and through its role in our episodic estimation of $\lambda$.

**Multi-modal loss** We optimize all models with the cross-entropy loss. For the multi-modal methods (nearest neighbors and IMP), we mask the loss to only include the closest neighbor/cluster for each class, in the same manner as inference. That is, for a class $n$, we find the most likely cluster $c^*_n \leftarrow \arg\max_{c:l_c=n} \log p(h_\phi(x)|\mu_c, \sigma_c)$ and then take the loss over the queries in the class $(Q_n)$:

$$J = \frac{1}{|Q_n|} \sum_{x \in Q_n} \Bigg[ -\log p(h_\phi(x) \mid \mu_{c^*_n}, \sigma_{c^*_n}) + \log \sum_{n' \neq n} p(h_\phi(x) \mid \mu_{c^*_{n'}}, \sigma_{c^*_{n'}}) \Bigg]. \tag{7}$$

Taking the loss for the closest clusters avoids over-penalizing multi-modality in the embedding. We found that masking improves the few-shot accuracy of these methods over other losses that incorporate all clusters.

*Table 1.* Multi-modal clustering and learning cluster variances on fully-supervised 10-way, 10-shot Omniglot alphabet recognition and 5-way, 5-shot mini-ImageNet. Scaling distances with the learned variance gives a small improvement and multi-modal clustering gives a further improvement.

| METHOD | $\sigma$ | MULTI-MODAL | ALPH. ACC. | MINI. ACC. |
|---|---|---|---|---|
| PROTOTYPES | - | - | $65.2 \pm 0.6$ | $66.1 \pm 0.6$ |
| PROTOTYPES | ✓ | - | $65.2 \pm 0.6$ | $67.2 \pm 0.5$ |
| IMP (OURS) | ✓ | ✓ | **92.0** $\pm$ 0.1 | **68.1** $\pm$ 0.8 |

*Table 2.* Learning labeled cluster variance $\sigma_l$ and unlabeled cluster variance $\sigma_u$ on semi-supervised 5-way, 1-shot Omniglot and mini-ImageNet with 5 unlabeled points per class and 5 distractors (see Section 4). Learning $\sigma_l, \sigma_u$ is better than learning a tied $\sigma$ for labeled and unlabeled clusters.

| METHOD | $\sigma$ | OMNI. ACC. | MINI. ACC. |
|---|---|---|---|
| TIED | $\sigma$ | 93.5±0.3 | 48.6±0.4 |
| IMP (OURS) | $\sigma_l, \sigma_u$ | **98.9**±0.1 | **49.6**±0.8 |

### 3.3. Ablations and Alternatives

We ablate our episodic and end-to-end extensions of DP-means to validate their importance for few-shot learning. Learning and performing inference with IMP is more robust to different choices of $\lambda$ than simply using DP-means during inference (Figure 3). Multi-modality and learned variance make their own contributions to accuracy (Table 1). Learning separate $\sigma_l, \sigma_u$, for labeled and unlabeled clusters respectively, is more accurate than learning a shared $\sigma$ for all clusters (Table 2). For full details of the datasets and settings in these ablations, refer to Section 4.

In principle, IMP's clustering can be iterated multiple times during training and inference. However, we found that one clustering iteration suffices. Two iterations during training had no effect on accuracy, and even 100 iterations during inference still had no effect on accuracy, showing that the clustering is stable.

**Alternative Algorithms** DP-means was derived through the limit of a Gibbs sampler as the variance approaches 0, and so it does hard assignment of points to clusters. With hard assignment, it is still possible to learn the embedding parameters $\phi$ end-to-end by differentiating through the softmax over distances between query points and support clusters as in Equation 4. However, hard assignment of labeled and unlabeled data is harmful in our experiments, especially early on in training (see supplement).

When reintroducing variance into multi-modal clustering as we do, a natural approach would be to reconsider the Gibbs sampler for the CRP (West et al., 1994; Neal, 2000) from which DP-means was derived, or other Dirichlet process inference methods such as expectation maximization (Kimura et al., 2013). These alternatives are less accurate in our experiments, mainly as a result of the CRP prior's "rich get richer" dynamics, which prefers clusters with more assignments (leading to accuracy drops of 5–10%). This is especially problematic early in training, when unlabeled points are often incorrectly assigned. The supplement includes derivations and experiments regarding these multi-modal clustering alternatives.

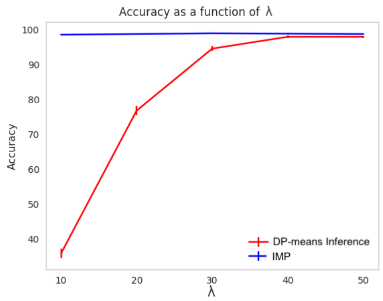

*Figure 3.* Learning and inference with IMP is more accurate and robust than DP-means inference on a prototypical network embedding alone. This plot shows the accuracy for the standard benchmark of semi-supervised 5-way, 1-shot Omniglot for different choices of the distance threshold $\lambda$ for creating a new cluster.

## 4. Experiments

We experimentally show that infinite mixture prototypes are more accurate and more general than uni-modal prototypes. We compare to a nearest neighbors baseline, which uses the same loss as IMP, but makes each data point its own cluster.

We control for architecture and optimization by comparing methods with the same base architecture of Vinyals et al. (2016) and same episodic optimization settings of Snell et al. (2017). For further details see Appendix A.1 of the supplement. All code for our method and baselines will be released at `https://github.com/k-r-allen/imp`.

We consider three datasets for few-shot learning:

**Omniglot** (Lake et al., 2015) is a dataset of 1,623 handwritten characters from 50 alphabets. There are 20 examples of each character, where the images are resized to 28x28 pixels and each image is rotated by multiples of 90°. This gives 6,492 classes in total, which are then split into 4,112 training classes, 688 validation classes, and 1,692 test classes.

**mini-ImageNet** (Vinyals et al., 2016) is a reduced version of the ILSVRC'12 dataset (Russakovsky et al., 2015), which contains 600 84x84 images for 100 classes randomly selected from the full dataset. We use the split from Ravi & Larochelle (2017) with 64/16/20 classes for train/val/test.

**tiered-ImageNet** (Ren et al., 2018) is a reduced version of the ILSVRC'12 dataset (Russakovsky et al., 2015), with 84x84 images of 608 classes from 34 super-classes. These

*Table 3.* Super and sub-class recognition accuracy. Super-classes have complex, multi-modal data distributions while sub-classes have simpler, uni-modal data distributions. IMP improves accuracy for super-classes, preserves accuracy for sub-classes, and generalizes better from super-classes to sub-classes. For Omniglot, super-class (alphabet) episodes are 10-way 10-shot; sub-class (character) episodes are 20-way 1-shot. For tiered-ImageNet, super-class episodes are 5-way 10-shot; sub-class episodes are 5-way 1-shot.

(A) OMNIGLOT

| TRAIN | TEST | PROTOS | IMP | NEIGHBORS |
|---|---|---|---|---|
| SUPER | SUPER | 65.6±0.4 | 92.0±0.1 | **92.4**±0.2 |
| SUPER | SUB | 82.1±0.4 | **95.4**±0.2 | **95.4**±0.2 |
| SUB | SUB | **94.9**±0.2 | **95.1**±0.1 | **95.1**±0.1 |

(B) TIERED-IMAGENET

| TRAIN | TEST | PROTOS | IMP | NEIGHBORS |
|---|---|---|---|---|
| SUPER | SUPER | **37.7**±0.4 | **37.9**±0.4 | **38.1**±0.4 |
| SUPER | SUB | 40.1±0.4 | **53.3**±1.0 | **52.4**±1.1 |
| SUB | SUB | **52.0**±1.1 | **52.5**±1.0 | **52.8**±1.0 |

are split into 20/6/8 super-classes for train/val/test.

### 4.1. Accuracy and Generality of Multi-modal Clustering by Infinite Mixture Prototypes

Our experiments on Omniglot alphabets and tiered-ImageNet super-classes show that multi-modal prototypes are significantly more accurate than uni-modal prototypes for recognizing complex classes (alphabets and super-classes) and recover uni-modal prototypes as a special case for recognizing simple classes (characters and sub-classes). Multi-modal prototypes generalize better for super-class to sub-class transfer learning, improving accuracy when training on super-classes but testing on sub-classes. By unifying the clustering of labeled and unlabeled data alike, our multi-modal prototypes also address fully unsupervised clustering, unlike prior prototypical network models that are undefined without labels.

We first show the importance of multi-modality for learning representations of multi-modal classes: Omniglot alphabets and tiered-ImageNet super-classes. For these experiments, we train for super-class classification, using only the super-class labels. Episodes are constructed by sampling $s$ super-classes, $n$ sub-classes within each super-class, and $k$ images of each sub-class. For Omniglot, we construct episodes with $s = 10$, $n = 10$ and $k = 1$. For tiered-ImageNet, episodes have $s = 5$, $n = 5$ and $k = 2$.

For sub-class testing, episodes are constructed with $n$ randomly sampled sub-classes, and $k$ examples of each class. Note that both super-class and sub-class testing are on held-out super-classes and sub-classes respectively.

As seen in Table 3, IMP substantially outperforms prototypical networks for sub-class recognition from super-class training. On 20-way 1-shot character recognition, IMP achieves 95.4% from alphabet supervision alone, slightly out-performing prototypical networks trained directly on character recognition (94.9%). Likewise, for tiered-ImageNet, IMP achieves 53.3% on 5-way 1-shot sub-class recognition from only super-class training, substantially outperforming prototypical networks. By clustering each super-class into multiple modes, IMP is better able to generalize to sub-classes.

For a parametric alternative, we trained MAML (Finn et al., 2017) on alphabet recognition, with the same episode composition as IMP. In our experiments, MAML achieved only 61.9% accuracy on 10-way 10-shot alphabet recognition. This demonstrates that a parametric classifier of this capacity, with decisions that are linear in the embedding, is not enough to solve alphabet recognition—instead, multi-modality is necessary.

*Table 4.* Generalization to held-out characters on 10-way, 5-shot Omniglot alphabet recognition. 40% of the characters are kept for training and 60% held out for testing. IMP maintains accuracy on held-out characters, suggesting that multi-modal clustering is more robust to new and different sub-classes from the same super-class.

| METHOD | TRAINING MODES | TESTING MODES | BOTH MODES |
|---|---|---|---|
| IMP (OURS) | **99.0**±0.1 | **94.9**±0.2 | **96.6**± 0.2 |
| PROTOTYPES | 92.4±0.3 | 77.7±0.4 | 82.9±0.4 |

To further examine generalization, we consider holding out character sub-classes during alphabet super-class training for Omniglot. In this experiment the training and testing alphabets are the same, but the characters within each alphabet are divided into training (40%) and testing (60%) splits. We compare alphabet recognition accuracy using training characters, testing characters, and all characters to measure generalization to held-out modes in Table 4. While prototypical networks achieve good accuracy on training modes, their accuracy drops 15% relative on testing modes, and still drops 10% relative on the combination of both modes. The reduced accuracy of prototypical networks on held-out modes indicates that uni-modality is not maintained on unseen characters even when they are from the same alphabets. IMP accuracy drops less than 5% relative from training to testing modes and both modes, showing that multi-modal clustering generalizes better to unseen data.

**Fully Unsupervised Clustering** IMP is able to do fully unsupervised clustering via multi-modality. Prototypical networks (Snell et al., 2017) and semi-supervised prototypical networks (Ren et al., 2018) are undefined without labeled data during testing because the number of clusters is defined by the number of classes.

*Table 5.* Unsupervised clustering of unseen Omniglot characters by IMP. Learning with IMP makes substantially purer clusters than DP-means inference on a prototypical network embedding, showing that the full method is necessary for best results.

| METHOD | METRIC | 10-WAY | 100-WAY | 200-WAY |
|---|---|---|---|---|
| IMP | PURITY | **0.97** | **0.90** | **0.91** |
| DP-MEANS | | 0.91 | 0.73 | 0.71 |
| IMP | NMI | **0.97** | **0.95** | **0.94** |
| DP-MEANS | | 0.89 | 0.88 | 0.87 |
| IMP | AMI | **0.92** | **0.81** | **0.70** |
| DP-MEANS | | 0.76 | 0.58 | 0.51 |

For this unsupervised clustering setting, we use the models that were optimized for alphabet recognition. For testing, we randomly sample 5 examples of $n$ character classes from the test set without labels.

IMP handles labeled and unlabeled data by the same clustering rule, infers the number of clusters as needed, and achieves good results under the standard clustering metrics of purity, and normalized/adjusted mutual information (NMI/AMI). We examine IMP's clustering quality on purely unlabeled data in Table 5. IMP maintains strong performance across a large number of unlabeled clusters, without knowing the number of classes in advance, and without having seen any examples from the classes during training.

As a baseline, we evaluate multi-modal inference by DP-means (Kulis & Jordan, 2012) on the embedding from a prototypical network with the same architecture and training data as IMP. We cross-validate the cluster threshold $\lambda$ on validation episodes for each setting, choosing by AMI.

### 4.2. Few-Shot Classification Benchmarks

We evaluate IMP on the standard few-shot classification benchmarks of Omniglot and mini-ImageNet in the fully-supervised and semi-supervised regimes.

We consider five strong fully-supervised baselines trained on $100\%$ of the data. We compare to three parametric methods, MAML (Finn et al., 2017), Reptile (Nichol & Schulman, 2018), and few-shot graph networks (Garcia & Bruna, 2018), as well as three nonparametric methods, nearest neighbors, prototypical networks (Snell et al., 2017), and the memory-based model of Kaiser et al. (2017).

Fully-supervised results are reported in Table 6. In this setting, we evaluate IMP in the standard episodic protocol of few-shot learning: shot and way are fixed and classes are balanced within an episode. IMP learns to recover uni-modal clustering as a special case, matching or out-performing prototypical networks when the classes are uni-modal.

In the semi-supervised setting of labeled and unlabeled examples we follow Ren et al. (2018). We take only $40\%$ of the data as labeled *for both supports and queries* while the rest of the data is included as unlabeled examples. The unlabeled data is then incorporated into episodes as (1) within-support examples that allow for semi-supervised refinement of the support classes or (2) *distractors* which lie in the complement of the support classes. Semi-supervised episodes augment the fully supervised $n$-way, $k$-shot support with 5 unlabeled examples for each of the $n$ classes and include 5 more distractor classes with 5 unlabeled instances each. The query set still contains only support classes.

Semi-supervised results are reported in Table 7. We train and test IMP, existing prototypical methods, and nearest neighbors in this setting. Semi-supervised prototypical networks (Ren et al., 2018) incorporate unlabeled data by soft $k$-means clustering (of their three comparable variants, we report "Soft $k$-Means+Cluster" results). Prototypical networks (Snell et al., 2017) and neighbors are simply trained on the $40\%$ of the data with labels.

Through multi-modality, IMP clusters labeled and unlabeled data by a single rule. In particular this helps with the distractor distribution, which is in fact more diffuse and multi-modal by comprising several different classes.

The results reported on these benchmarks are for models trained and tested with $n$-way episodes. This is to equalize comparison across methods[1].

## 5. Related Work

**Prototypes** Prototypical networks (Snell et al., 2017) and semi-supervised prototypical networks (Ren et al., 2018) are the most closely related to our work. Prototypical networks simply and efficiently represent each class by its mean in a learned embedding. They assume that the data is fully labeled and uni-modal in the embedding. Ren et al. (2018) extend prototypes to the semi-supervised setting by refining prototypes through soft $k$-means clustering of the unlabeled data. They assume that the data is at least partially labeled and retain the uni-modality assumption. Both Snell et al. (2017) and Ren et al. (2018) are limited to one cluster per class. Mensink et al. (2013) represent classes by the mean of their examples in a linear embedding to incorporate new classes into large-scale classifiers without re-training. They extend their approach to represent classes by multiple prototypes, but the number of prototypes per class is fixed and hand-tuned, and their approach does not incorporate unlabeled data. We define a more general and adaptive approach

[1] Snell et al. (2017) train at higher way than testing and report a boost in accuracy. We find that this boost is somewhat illusory, and at least partially explained away by controlling for the number of gradients per update. We show this by experiment through the use of gradient accumulation in Appendix A.2 of the supplement. (For completeness, we confirmed that our implementation of prototypical networks reproduces reported results at higher way.)

*Table 6.* Fully-supervised few-shot accuracy using 100% of the labeled data. IMP performs equal to or better than prototypical networks (Snell et al., 2017). Although IMP is more general, it can still recover uni-modal clustering as a special case.

| | **Omniglot** | | | | **mini-ImageNet** | |
|---|---|---|---|---|---|---|
| | 5-way | | 20-way | | 5-way | |
| **Method** | 1-shot | 5-shot | 1-shot | 5-shot | 1-shot | 5-shot |
| IMP (ours) | 98.4±0.3 | 99.5±0.1 | 95.0±0.1 | 98.6±0.1 | 49.6±0.8 | **68.1±0.8** |
| Neighbors | 98.4±0.3 | 99.4±0.1 | 95.0±0.1 | 98.3±0.1 | 49.6±0.8 | 59.4±1.0 |
| Snell et al. (2017) | 98.2±0.3 | 99.6±0.1 | 94.9±0.2 | 98.6±0.1 | 47.0±0.8 | 66.1±0.7 |
| Finn et al. (2017) | 98.7±0.4 | 99.9±0.3 | 95.8±0.3 | **98.9**±0.2 | 48.7±1.84 | 63.1±0.92 |
| Garcia & Bruna (2018) | **99.2** | 99.7 | **97.4** | **99** | **50.3** | 66.41 |
| Kaiser et al. (2017) | 98.4 | 99.6 | 95 | 98.6 | - | - |

*Table 7.* Semi-supervised few-shot accuracy on 40% of the labeled data with 5 unlabeled examples per class and 5 distractor classes. The distractor classes are drawn from the complement of the support classes and are only included unlabeled. IMP achieves equal or better accuracy than semi-supervised prototypical networks (Ren et al., 2018).

| | **Omniglot** | | | | **mini-ImageNet** | |
|---|---|---|---|---|---|---|
| | 5-way | | 20-way | | 5-way | |
| **Method** | 1-shot | 5-shot | 1-shot | 5-shot | 1-shot | 5-shot |
| IMP (ours) | **98.9 ± 0.1** | **99.4 ± 0.1** | **96.9 ± 0.2** | **98.3 ± 0.1** | **49.2 ± 0.7** | **64.7 ± 0.7** |
| Ren et al. (2018) | 98.0 ± 0.1 | **99.3 ± 0.1** | 96.2 ± 0.1 | 98.2 ± 0.1 | **48.6 ± 0.6** | 63.0 ± 0.8 |
| Neighbors | 97.9 ± 0.2 | 99.1 ± 0.1 | 93.8 ± 0.2 | 97.5 ± 0.1 | 47.9 ± 0.7 | 57.3 ± 0.8 |
| Snell et al. (2017) | 97.8 ± 0.1 | 99.2 ± 0.1 | 93.4 ± 0.1 | 98.1 ± 0.1 | 45.1 ± 1.0 | 62.5 ± 0.5 |

through infinite mixture modeling that extends prototypical networks to multi-modal clustering, with one or many clusters per class, of labeled and unlabeled data alike.

**Metric Learning** Learning a metric to measure a given notion of distance/similarity addresses recognition by retrieval: given an unlabeled example, find the closest labeled example. Kulis (2013) gives a general survey. The contrastive loss and siamese network architecture (Chopra et al., 2005; Hadsell et al., 2006) optimize an embedding for metric learning by pushing similar pairs together and pulling dissimilar pairs apart. Of particular note is research in face recognition, where a same/different retrieval metric is used for many-way classification (Schroff et al., 2015). Our approach is more aligned with metric learning by meta-learning (Koch, 2015; Vinyals et al., 2016; Snell et al., 2017; Garcia & Bruna, 2018). These approaches learn a distance function by directly optimizing the task loss, such as cross-entropy for classification, through episodic optimization (Vinyals et al., 2016) for each setting of way and shot. Unlike metric learning on either neighbors (Goldberger et al., 2004; Schroff et al., 2015) or prototypes (Snell et al., 2017; Ren et al., 2018), our method adaptively interpolates between neighbor and uni-modal prototype representation by deciding the number of modes during clustering.

**Cognitive Theories of Categorization** Our approach is inspired by the study of categorization in cognitive science. Exemplar theory (Nosofsky, 1986) represents a category by storing its examples. Prototype theory (Reed, 1972) represents a category by summarizing its examples, by for instance taking their mean. Vanpaemel et al. (2005) recognize that exemplars and prototypes are two extremes, and define intermediate models that represent a category by several clusters in their varying abstraction model. However, they do not define how to choose the clusters or their number, nor do they consider representation learning. Griffiths et al. (2007) unify exemplar and prototype categorization through the hierarchical Dirichlet process to model the transition from prototypes to exemplars as more data is collected. They obtain good fits for human data, but do not consider representation learning.

## 6. Conclusion

We made a case for the importance of considering the complexity of the data distribution in the regime of few-shot learning. By incorporating infinite mixture modeling with deep metric learning, we developed infinite mixture prototypes, a method capable of learning end-to-end and adapting its model capacity to the given data. Our multi-modal extension of prototypical networks additionally allows for fully unsupervised inference, and the natural incorporation of semi-supervised data during learning. As few-shot learning is applied to increasingly challenging tasks, models with adaptive complexity will become more important. Adaptive, multi-modal representation is likely to prove important for life-long learning settings, as well as for integrating multiple input modalities such as joint visual/auditory signals.

## Acknowledgements

We gratefully acknowledge support from DARPA L2M grant 6938423 and KA is supported by NSERC. We thank Trevor Darrell and Ghassen Jerfel for advice and helpful discussions.

## References

Aldous, D. J. Exchangeability and related topics. In *École d'Été de Probabilités de Saint-Flour XIII1983*, pp. 1–198. Springer, 1985.

Chopra, S., Hadsell, R., and LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, volume 1, pp. 539–546. IEEE, 2005.

Cover, T. and Hart, P. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

Fei-Fei, L., Fergus, R., and Perona, P. One-shot learning of object categories. *PAMI*, 2006.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

Garcia, V. and Bruna, J. Few-shot learning with graph neural networks. In *ICLR*, 2018.

Goldberger, J., Roweis, S., Hinton, G., and Salakhutdinov, R. Neighbourhood components analysis. In *NIPS*, pp. 513–520, 2004.

Griffiths, T. L., Canini, K. R., Sanborn, A. N., and Navarro, D. J. Unifying rational models of categorization via the hierarchical dirichlet process. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 2007.

Hadsell, R., Chopra, S., and LeCun, Y. Dimensionality reduction by learning an invariant mapping. In *CVPR*, volume 2, pp. 1735–1742. IEEE, 2006.

Hjort, N. L., Holmes, C., Müller, P., and Walker, S. G. *Bayesian nonparametrics*, volume 28. Cambridge University Press, 2010.

Jones, M. C., Marron, J. S., and Sheather, S. J. A brief survey of bandwidth selection for density estimation. *Journal of the American statistical association*, 91(433):401–407, 1996.

Kaiser, L., Nachum, O., Roy, A., and Bengio, S. Learning to remember rare events. In *ICLR*, 2017.

Kimura, T., Tokuda, T., Nakada, Y., Nokajima, T., Matsumoto, T., and Doucet, A. Expectation-maximization algorithms for inference in dirichlet processes mixture. *Pattern Analysis and Applications*, 16(1):55–67, 2013.

Koch, G. Siamese neural networks for one-shot image recognition. In *NIPS Deep Learning Workshop*, 2015.

Kulis, B. Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5(4):287–364, 2013.

Kulis, B. and Jordan, M. I. Revisiting k-means: New algorithms via bayesian nonparametrics. In *ICML*, 2012.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Mensink, T., Verbeek, J., Perronnin, F., and Csurka, G. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2624–2637, 2013.

Miller, E. G., Matsakis, N. E., and Viola, P. A. Learning from one example through shared densities on transforms. In *CVPR*, volume 1, pp. 464–471. IEEE, 2000.

Neal, R. M. Markov chain sampling methods for dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265, 2000.

Nichol, A. and Schulman, J. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2018.

Nosofsky, R. M. Attention, similarity, and the identification–categorization relationship. *Journal of experimental psychology: General*, 115(1):39, 1986.

Parzen, E. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3): 1065–1076, 1962.

Rasmussen, C. E. The infinite gaussian mixture model. In *Advances in neural information processing systems*, pp. 554–560, 2000.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. In *ICLR*, 2017.

Reed, S. K. Pattern recognition and categorization. *Cognitive psychology*, 3(3):382–407, 1972.

Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J. B., Larochelle, H., and Zemel, R. S. Meta-learning for semi-supervised few-shot classification. In *ICLR*, 2018.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

Schroff, F., Kalenichenko, D., and Philbin, J. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pp. 815–823, 2015.

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning. In *NIPS*, pp. 4080–4090, 2017.

Vanpaemel, W., Storms, G., and Ons, B. A varying abstraction model for categorization. In *Proceedings of the Annual Conference of the Cognitive Science Society*, volume 27, pp. 2277–2282, 2005.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. In *NIPS*, pp. 3630–3638, 2016.

West, M., Mller, P., and Escobar, M. Hierarchical priors and mixture models, with application in regression and density estimation. *Aspects of Uncertainty: A Tribute to DV Lindley*, pp. 363–386, 1994.