LEARNING TO GENERALIZE TO UNSEEN CLASSES WITH DISCRIMINATIVE VARIATIONAL SET EMBEDDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent embedding-based meta-learning approaches, which learns a metric space that generalizes well over large number of different classification tasks, have been shown to be effective on few-shot classification tasks. These embedding approaches are often trained by representing classes and instances as embeddings and reducing the distance of each instance to its correct class embedding. However, they generally embed the support set into a single point, which may not accurately represent the true distribution of the class prototype. To tackle this issue, we propose a meta-learning framework with a novel discriminative probablistic set-embedding model, that can learn to *discriminatively* embed sample sets for each class into non-overlapping distributions using both set-wise and instance-wise loss. Specifically, we assume that the embedding follows a Gaussian distribution and learn its mean and variance via variational inference over large number of episodes. This enables the model to consider uncertainty of the estimate which is often high when made with few samples. Further, exploiting the set-wise discriminativity in our model, we explicitly *learn to generalize to unseen classes* by enforcing the classification between the seen classes to well-separate the embeddings for the unseen classes as well. We validate our model, Discriminative Variational Set Embedding (DiVaSE) on multiple datasets for few-shot classification tasks, on which it significantly outperforms existing meta-learning methods.

1 INTRODUCTION

While deep learning models such as CNNs have been proven effective on multi-class classification (Krizhevsky et al., 2012; He et al., 2016; Huang et al., 2017), even surpassing human performances (Deng et al., 2009), such impressive performances are obtained with the availability of large number of training instances per class. However, in more realistic settings where we could have very few training instances for some classes, deep learning models may fail to obtain good accuracies due to overfitting. On the other hands, humans can generalize surprisingly well even with a single instance from each class. This problem, known as the few (one)-shot learning problem, thus has recently attracted large attention, leading to the proposal of many prior work that aim to prevent the model from overfitting when trained with few instances.

Recently, meta-learning approaches that learn to generalize over multiple tasks (learning to learn) (Vinyals et al., 2016; Santoro et al., 2016; Rezende et al., 2016; Snell et al., 2017) have obtained impressive performances on the few-shot learning tasks. They tackle the low-data challenge in few-shot learning by learning a classifier to solve any few-shot learning problem well, by training over large number of episodes, where in each episode the model randomly solves a different few-shot classification problem. A popular approach in composing such meta knowledge is to assume some common metric space and learn the optimal embedding function on it. Matching Networks (Vinyals et al., 2016) and Prototypical Networks (Snell et al., 2017) are examples of such embedding approaches, which are known to perform well and computationally efficient as well.

However, these approaches are limited in that they assume for each episode, the prototype set of each class is sampled and *mapped into a single point* in the metric space (e.g. embedding mean). Then the distance (e.g. Euclidean) between class prototypes and query instances are measured to

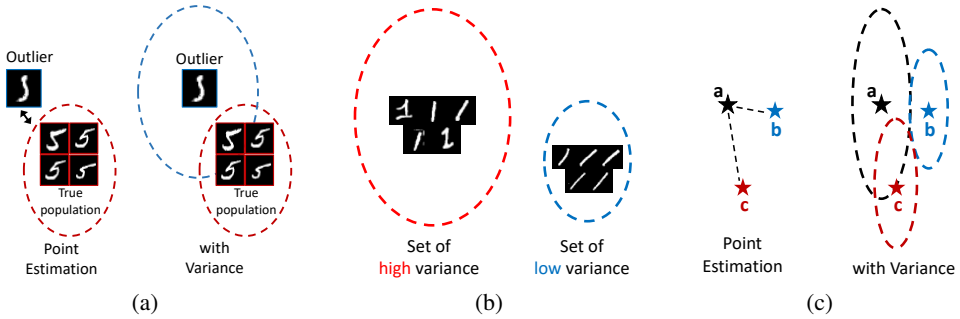


Figure 1: **Concepts.** (a) When the outlier sample locates far from the true population, it is very risky to use this sample as the class prototype. However, the problem is much alleviated with large variance to cover the true population. (b) Variance with respect to the different level of diversity of set elements. (c) In point estimation, equal distance metric is applied along all directions, and it is hard to know which direction is more sensitive (or important) than the other in classfication. On the other hand, if the class instances are more likely to locate along some specific dimensions (y-axis in the example) and in a similar vein learned variances also have narrow shapes along those dimensions, then different distance metrics can be applied along each dimension, such that some dimension (x-axis in the example) has more sensitive distance metric than the other.

determine their class identities, without consideration of other statistics of the sets such as variance. This point estimation of the true population suffers from lack of robustness in its estimation, which limits its generalization ability to unseen classes.

Specifically, when we have a biased sample and want it to represent the true class population, the uncertainty should increase to make the prototype to cover other instances from the same class population (See Figure 1(a)). However, point estimation cannot represent uncertainty as it only consider the sample mean and no other statistics. Therefore, whenever we sample such biased examples as a class prototype, point estimation is likely to propagate wrong information without letting query instances be aware of the risks, resulting in degraded performance in the classification. It can be prevented by learning proper amount of uncertainties for each sampled prototype set.

Another limitation of point estimate is that it cannot represent the diversity among the elements in the prototype set. Even with the same number of samples for each episode (or shots), the sets may have varying degree of diversity (See Figure 1(b)). However, point estimation of the class embedding cannot yield any more informative statistics than their mean, and thus cannot fully exploit the benefit of having multiple samples to represent a class.

Finally, the sets representing classes should be discriminatively embedded, but with a point estimate of the class embedding it is not clear if the two sets are well separated unless there exists additional samples from the two classes that can be used to generate discriminative constraints (an instance should be closer to its own class prototype than other class prototypes). Constraining the embeddings to have minimum distances among them is one solution but it does not consider which dimension is more important than others when discriminating between the two (See Figure 1(c)). In fine-grained classification problems, two classes may be very similar in all features but one.

In conclusion, we need to develop a method that can well represent a given set, such that sets with biased samples or high variance should be mapped to a distribution with larger variance, to more accurately reflect uncertainty in the true population. In this sense, *representing each class prototype set as a distribution* is a very natural way to improve upon the current point-estimate based few-shot learning approaches. Such a probabilistic embedding model will be more robust to overfitting and generalize better on unseen classes.

Lastly, another important advantage of a set embedding method is that all the sampled instances can be set-wisely embedded at once, leading to less computational complexity compared to embedding all individual elements. For example, when we perform classification given a handful of prototype sets and query instances for each episode, we can make this classification loss to induce that the other classes that do not participate in the classification also discriminate each other, in a *set-wise manner* (See Figure 2). Exploiting such computational efficiency, we explicitly train the meta-learner to improve the generalization performance on unseen classes in the meta-test dataset. Note that simply increasing the number of classes (or way) at training time is expensive because the number of pairs

of a prototype and a query instance increases quadratically. We can minimize the compuational cost by encapsulating the whole set into a single distribution, and efficiently performing set-wise discrimination.

We validate our model on multiple public benchmark datasets for few-shot classification task against state-of-the-art few-shot meta-learning models, of which our model significantly outperforms.

In summary, our contribution is twofold:

- We propose a novel *discriminative variational set embedding* (DiVaSE) model that discriminatively embeds an entire set into a Gaussian distribution, whose mean and variance is learned with variationl inference to consider both the uncertainty and diversity among the embedded samples, as well as to maximize discriminativity among classes.
- Using our discriminative variational set embedding model, we propose an efficient meta-learning algorithm that explicitly targets to improve generalization to unseen classes, by introducing a contextual loss for performing set-wise discrimiation between meta-test classes.

2 Related Work

Meta Learning for few-shot learning Meta-learning (Thrun, 1998) generally refers to an approach to learn to generalize to a new task by learning over large number of tasks, rather than to a new data instance from a single task. Meta-learning is recently gaining popularity as an effective tool for few-shot leanring as its aim in learning easily-generalizable model helps overcome the overfitting problem. Matching networks (Vinyals et al., 2016) proposed to train a model over multiple episodes (tasks), where at each episode the training set for each class is divided into the support set that contain few instances that represent the class, and the query set that is used to generate the instance-wise classification loss. Prototypical networks (Snell et al., 2017) introduced a simple inductive bias to this meta-learning strategy, such that the instances from the same class should be clustered around a single class prototype, which achieved even better performances on few-shot learning. Yang et al. (2018) proposed to learn the metric with additional nonlinear transformations. Finn et al. (2017) proposed model agnostic meta-learning (MAML), a non-embedding approach that works with gradients of any model instead of working on an embedding space, and Finn et al. (2018) and Kim et al. (2018) propose probabilistic version of this MAML. To our knowledge, none of the existing meta-learning approaches for few-shot learning explicilty targeted to improve the generalization performance on unseen classes as done in our work.

Metric learning and discriminative embedding methods for classification Our embedding model in some sense can be viewed as a metric learning approach. Metric learning is an extensively studied topic, and we only name a few. Xing et al. (2002) proposed a metric learning method parameterized by a Mahalanobis matrix that transforms the input features to minimize distances between the instances in the same class with the constraints that instances in different classes should stay further than a large margin. Large-margin metric learning (Weinberger & Saul, 2009) reformulated the objective by introducing a large-margin triplet loss, for mapping an instance to be closer to instances from the same class than instances from different classes by a large margin. To achieve scalability to large-scale settings, some approaches simply map each instance close to the mean of the sampled instances from the training set for each class, such as Mensink et al. (2013) and Prototypical Networks (Snell et al., 2017), considering it as the class prototype. On the contrary, our model embeds a set of prototype samples from each class as a distribution, to represent the uncertainty in the estimation of the true distribution as well as its variance.

Set Encoding Working with a set of instances as a whole, instead of with individual instances, is an important property of our meta-learning model which enables it to generalize over datasets with varying size and labels. One important factor with set representation is its invariance to instance order, and recent work have proposed ways to obtain such reprsentation using deep learning. Vinyals et al. (2015) proposed a way to obtain order-invariant representations by randomly shuffling the instances that is fed into the bidirectional LSTM, and Zaheer et al. (2017) proposed an additive approach to achieve the same goal. Perhaps the most relevant to our work is Neural Statistician (Edwards & Storkey, 2016), which learned to represent the distribution of the given set and

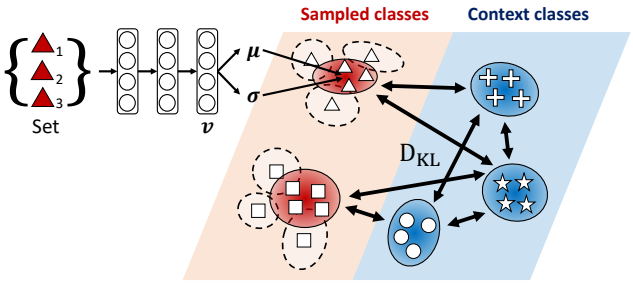


Figure 2: **Our meta-learning framework.** A neural network takes as an input each class prototype set, and map it to a Gaussian distribution with mean and variance. KL divergences between sampled and context classes, and within context classes are maximized to learn to generalize on unseen classes.

applied it to few-shot learning. However, it is trained neither with discriminative objective nor with meta-learning and thus does not obtain practical performance for classification.

3 APPROACH

We start by introducing the general problem statement. Define a set $[N] := \{1, \ldots, N\}$. We consider the problem of generalizing to unseen classes, where we have a training set consisting of S *seen* classes: $\mathcal{D}_{\text{tr}} = \{\mathcal{X}_1, \ldots, \mathcal{X}_S\}$, and a test dataset $\mathcal{D}_{\text{te}} = \{\mathcal{X}_{S+1}, \ldots, \mathcal{X}_{S+U}\}$ consisting of U *unseen* classes. Our goal then is to minimize the classification error between samples from this unseen classes. For each class $i \in [S+U]$, set $\mathcal{X}_i$ has N_i number of datapoints $\mathbf{x}_{i,1}, \ldots, \mathbf{x}_{i,N_i}$. We do not assume that N_i is fixed across classes, and in real-world datasets, N_i may actually vary largely across classes.

Episodes At train time, we are interested in episodic distribution p_{epi} where an episode is sampled from $\mathcal{D}_{\text{tr}}$. Specifically when generating an episode, a subset class indices $\mathcal{I} \subset [S]$ is first sampled, then for each class $i \in \mathcal{I}$, a subset $\hat{\mathcal{X}}_i \subset \mathcal{X}_i$ is sampled to form the prototype of the class. The collection of them is denoted as $\mathcal{P} = \{\hat{\mathcal{X}}_i \subset \mathcal{X}_i : i \in \mathcal{I}\}$. The remaining instances in $\mathcal{X}_i \backslash \hat{\mathcal{X}}_i$ are query instances for class i, whose distances from each of the class prototypes $\hat{\mathcal{X}}_j$ for all $j \in \mathcal{I}$ are measured in some metric space to determine their class identity. Note that $\mathcal{Q} = \{\mathcal{X}_i \backslash \hat{\mathcal{X}}_i : i \in \mathcal{I}\}$ is fully specified once we sample $\mathcal{P}$ from $p_{\text{epi}}(\mathcal{P}|\mathcal{D}_{\text{tr}})$. An episode is then defined as a tuple $(\mathcal{P}, \mathcal{Q})$.

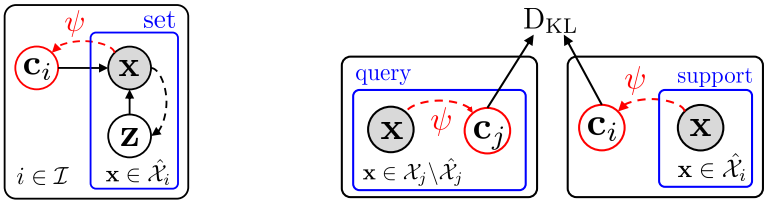
Training and Testing The goal of meta learning based on a metric space is to obtain the optimal embedding function q_ψ for the given series of random splits of $\mathcal{P}$ and $\mathcal{Q}$. The embedding function q_ψ maps a set to a vector (Snell et al., 2017) or to a distribution (Edwards & Storkey, 2016). While using diverse forms of learning objective is possible depending on modeling assumption, for now we simply denote it as a $\mathcal{L}$.

$$\psi^* = \arg\min_{\psi} \mathbb{E}_{p_{\text{epi}}(\mathcal{P}|\mathcal{D}_{\text{tr}})} \left[\mathcal{L}(\psi; \mathcal{P}, \mathcal{Q}) \right] \tag{1}$$

At each stochastic gradient descent step, we approximate the gradient of equation 1 with a single Monte-Carlo sample $\mathcal{P}_t$ and $\mathcal{Q}_t$, that is $\nabla \mathcal{L}(\psi; \mathcal{P}_t, \mathcal{Q}_t)$. Testing is done in a similar way. We randomly generate a pre-defined number of episodes (e.g. 1000) from $p_{\text{epi}}(\mathcal{P}|\mathcal{D}_{\text{te}})$. For each episode, we calculate accuracy with the learned embedding function q_{ψ^*}, and then take the average of theses accuracies.

3.1 LEARNING TO DISCRIMINATIVELY ENCODE LABELED SETS TO DISTRIBUTIONS

In this section, we introduce our novel approach to solve the problem described above. Our goal is to learn embedding space from a set into a distribution in which embedded distributions corresponding to different classes are well separated. Toward this, we devise a novel loss function $\mathcal{L}$ that consists of two terms: (i) a generative loss to encode a set into a distribution and (ii) a discriminative loss to locate different distributions in separated regions.



(a) generative model

(b) KL divergence for discriminating between classes

Figure 3: (a) The generative model for DiVaSE, for each sampled classes $i \in \mathcal{I}$ and dataset $\hat{\mathcal{X}}_i$. The posterior of $\mathbf{c}_i$ is obtained given a dataset $\hat{\mathcal{X}}_i$, which is the embedding distribution we work with. (b) For all the class pairs $i, j \in \mathcal{I}$, the embedding of each query instance and the that of each support set are compared through KL divergence, to determine class identities of query instances.

Our generative loss basically is built on similar intuition in Edwards & Storkey (2016), with slight modification in the graphical structure (see below for details). The original Neural Statistician developed in Edwards & Storkey (2016) assumes hierarchical latent variables to generate $\mathcal{D}_{\text{tr}}$: the first variable $\mathbf{c}$ is to locate class-level variance, and the other variable $\mathbf{z}$ captures instance-level variance (Kingma et al., 2014). Each datapoint $\mathbf{x}$ is assumed to be generated conditionally on those two latent variables. Unlike Edwards & Storkey (2016), we adopt the dependency between variables as shown in Figure 3(a) to simplify the inference procedure and allow efficient episodic meta-training[1]. For each episode, the generative process of the sampled dataset $\mathcal{P} = \{\hat{\mathcal{X}}_1, \dots, \hat{\mathcal{X}}_S\}$ is as follows:

$$p(\mathbf{c}_i) = \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad p(\mathcal{P}) = \prod_{i \in \mathcal{I}} \int p(\mathbf{c}_i) \left[\prod_{\mathbf{x} \in \hat{\mathcal{X}}_i} \int p(\mathbf{z}) p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{c}_i) \mathrm{d}\mathbf{z} \right] \mathrm{d}\mathbf{c}_i \quad (2)$$

where θ is the collection of parameters for the generative process. Note that the exact posterior of $\mathbf{c}$ and $\mathbf{z}$ is intractable. Instead, we resort to variational mean-field approximation with tractable form of lower bound. If we further approximate it with a single Monte-Carlo sample, then we obtain the following generative loss:

$$\mathcal{L}_{\text{gen}}(\psi, \theta, \phi; \mathcal{P}) = \sum_{i \in \mathcal{I}} \sum_{\mathbf{x} \in \hat{\mathcal{X}}_i} \Big(\log p_\theta(\mathbf{x}|\tilde{\mathbf{c}}_i, \tilde{\mathbf{z}}) - D_{\text{KL}}[q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})] \Big) - D_{\text{KL}}[q_\psi(\mathbf{c}_i|\hat{\mathcal{X}}_i) \| p(\mathbf{c}_i)] \quad (3)$$

where $\tilde{\mathbf{c}}_i$ and $\tilde{\mathbf{z}}$ are sampled from $q_\psi(\mathbf{c}_i|\hat{\mathcal{X}}_i)$ and $q_\phi(\mathbf{z}|\mathbf{x})$ respectively, and backpropagated with reparameterization trick w.r.t. the variational parameters ψ and ϕ (Kingma & Welling, 2013).

Note that q_ψ is the embedding function discussed above, which maps set $\hat{\mathcal{X}}_i$ to the corresponding posterior distribution of $\mathbf{c}_i$. In doing so, we use Deep set (Zaheer et al., 2017) to convert the set $\hat{\mathcal{X}}_i$ into a single vector, namely $\mathbf{v}_i$. Specifically, each elements $\mathbf{x} \in \hat{\mathcal{X}}_i$ is nonlinearly transformed with multiple layers denoted as $f_\psi(\mathbf{x})$ and they are collected and averaged into $\mathbf{v}_i$. Finally, mean and log variance of $\mathbf{c}_i$ are generated from $\mathbf{v}_i$:

$$\mathbf{v}_i = \frac{1}{\hat{N}_i} \sum_{\mathbf{x} \in \hat{\mathcal{X}}_i} f_\psi(\mathbf{x}), \quad q_\psi(\mathbf{c}_i|\hat{\mathcal{X}}_i) = \mathcal{N}(\boldsymbol{\mu}_\psi(\mathbf{v}_i), \mathrm{diag}(\boldsymbol{\sigma}^2_\psi(\mathbf{v}_i))) \quad (4)$$

In all our experiments, we use a single layer implementation for each $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, which seems to perform better than applying additional nonlinearities on $\mathbf{v}_i$.

The most critical limitation of Edwards & Storkey (2016) as a classifier is that the generative loss $\mathcal{L}_{\text{gen}}$ embeds each class set independently. There is no explicit consideration for discriminating between embeddings for different classes, hence they start to collide and the performance degrades even with the moderate number of classes considered at a time. To solve this issue, we propose to minimize the distance $d(\cdot, \cdot)$ between the distribution of prototype set $q_\psi(\mathbf{c}_i|\hat{\mathcal{X}}_i)$ and that of query

[1] We also used this version of Neural Statistician as our baseline.

instances $q_\psi(\mathbf{c}_i|\mathbf{x}), \mathbf{x} \in (\mathcal{X}_i \backslash \hat{\mathcal{X}}_i)$ if they belong to the same class i. Otherwise, we maximize the distance between them:

$$\mathcal{L}_{\text{disc}}(\psi; \mathcal{P}, \mathcal{Q}) = \sum_{i \in \mathcal{I}} \left[\sum_{\mathbf{x} \in (\mathcal{X}_i \backslash \hat{\mathcal{X}}_i)} d\Big(q_\psi(\mathbf{c}_i|\hat{\mathcal{X}}_i), q_\psi(\mathbf{c}_i|\mathbf{x})\Big) + \log \sum_{j \in \mathcal{I}} \exp\Big(-d\Big(q_\psi(\mathbf{c}_i|\hat{\mathcal{X}}_j), q_\psi(\mathbf{c}_i|\mathbf{x})\Big)\Big) \right] \tag{5}$$

why is there a x?

The form of the objective is log of softmax probabilities, which is similar to that of Prototypical Network (Snell et al., 2017). However, since we need to measure *the similarity between two probabilistic distributions*, Euclidean distance or cosine similarity cannot be used as a proper metric. Here, although it is not a distance, we use Kullback-Leibler (KL) divergence,

$$d(P, Q) := D_{\text{KL}}[P \| Q]. \tag{6}$$

While D_{KL} is not commutative, it turns out using opposite direction of KL divergence still works well mainly because both distributions in equation 5 are in the same Gaussian family.

3.2 Learning to Generalize to Unseen Classes

In the episodic training strategy, only subset of classes are sampled (classes in $\mathcal{I}$) and used to construct the discriminative loss in equation 5. However, we still want to make our embedding function q_ψ to consider even remaining classes (that is, classes not in $\mathcal{I}$) as well, which we call *context classes*: $\mathcal{C} \subset [S] \backslash \mathcal{I}$. In order to achieve this purpose, we propose additional scheme where we periodically (e.g. for every 100th episode) store the set vector $\mathbf{v}_i$ and use them to maximize the KL divergences even for context classes:

idea: keep these valu

$$\mathcal{L}_{\text{context}}(\psi; \mathcal{P}, \bar{\mathcal{V}}) = \underbrace{\sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{C}} -d(q_\psi(\mathbf{c}_i|\hat{\mathcal{X}}_i), q_\psi(\mathbf{c}_j|\bar{\mathbf{v}}_j))}_{\text{between sampled and context classes}} + \underbrace{\sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{C}} -d(q_\psi(\mathbf{c}_i|\bar{\mathbf{v}}_i), q_\psi(\mathbf{c}_j|\bar{\mathbf{v}}_j))}_{\text{within context classes}} \tag{7}$$

where we abuse notation and $q_\psi(\mathbf{c}_i|\bar{\mathbf{v}}_i) = \mathcal{N}(\boldsymbol{\mu}_\psi(\bar{\mathbf{v}}_i), \text{diag}(\boldsymbol{\sigma}^2_\psi(\bar{\mathbf{v}}_i)))$. This scheme is based on past embedding, which is not perfect, but can prevent severe collisions among unselected classes while greatly reducing the computational cost of inferring $q_\psi(\mathbf{c}_i|\mathcal{X}_i)$ for all classes every episode.

The final version of our learning algorithm is described in Algorithm 1. The model has two hyperparameters λ and γ for scaling each loss term. We can tune them via cross-validation in practice. More detailed experimental setup is introduced in the next section.

Algorithm 1 Discriminative Variational Set Embedding (DiVaSE) with Contexts

for each *test* episode t **do**
  $\mathcal{I} \leftarrow M$ indices uniformly sampled from $[S]$
  $\mathcal{C} \leftarrow M'$ indices uniformly sampled from $[S] \backslash \mathcal{I}$ ▷ Context classes
  $\mathcal{P}, \mathcal{Q} \leftarrow \{\hat{\mathcal{X}}_i : i \in \mathcal{I}\}, \{\mathcal{X}_i \backslash \hat{\mathcal{X}}_i : i \in \mathcal{I}\}$ ▷ Random split of support and query set
  if t is at the predefined interval **then**
    Compute $\bar{\mathbf{v}}_i = \frac{1}{\hat{N}_i} \sum_{\mathbf{x} \in \hat{\mathcal{X}}_i} f_\psi(\mathbf{x}), \quad \forall i \in [S]$
    $\bar{\mathcal{V}} \leftarrow (\mathbf{v}_1, \ldots, \mathbf{v}_S)$ ▷ Update set vectors and store them
  end if
  $\mathcal{L}_t \leftarrow \mathcal{L}_{\text{gen}}(\psi, \theta, \phi; \mathcal{P}) + \lambda \mathcal{L}_{\text{disc}}(\psi; \mathcal{P}, \mathcal{Q}) + \gamma \mathcal{L}_{\text{context}}(\psi; \mathcal{P}, \bar{\mathcal{V}})$
  Update parameters with $\nabla \mathcal{L}_t$
end for

4 Experiment

We evaluate our approach on three datasets against relevant baselines. We follow the conventional experimental setup, where in each episode we consider multi-class classification between fixed number of classes to compare (way), with fixed number of instances sampled for each class prototype (shot). We train with both 1-shot and 5-shot to generate a single model, and test the model with each 1 and 5 shot separately. We train with 5-way, and test with both 5-way and 20-way. The number of context classes is set to 10 times to that of sampled classes: $M' = |\mathcal{C}| = 10|\mathcal{I}|$.

Table 1: **Generalization performance on unseen classes in test error (%).** The reported numbers are mean and standard errors with 95% confidence interval over 5 runs. Each run consists of the mean accuracy over 1000 episodes.

	5-way		20-way	
Models	1-shot	5-shot	1-shot	5-shot
Neural Statistician	54.42 ± 1.62	84.99 ± 0.35	35.01 ± 3.81	57.83 ± 4.47
Relation Network	86.89 ± 1.70	95.63 ± 0.31	68.21 ± 0.75	85.45 ± 0.56
Prototypical Network	87.38 ± 1.70	96.87 ± 0.09	70.94 ± 0.46	90.55 ± 0.11
DiVaSE (w/o context)	90.59 ± 0.33	96.98 ± 0.23	77.25 ± 0.31	91.08 ± 0.51
DiVaSE (w/ context)	$\mathbf{91.12 \pm 0.23}$	$\mathbf{97.20 \pm 0.26}$	$\mathbf{77.79 \pm 0.29}$	$\mathbf{91.53 \pm 0.22}$

Baselines and our models We first introduce relevant baselines and our models.

1) Neural Statistician (Edwards & Storkey, 2016). This is a generative model based on VAE (Kingma & Welling, 2013) that does not explicitly consider discrimination between classes. Following the evaluation process in this work, we calculate test accuracies by comparing KL divergences of the posteriors.

2) Relation Network (Yang et al., 2018). This is a model that also learns a distance metric by stacking multiple layers with nonlinearity, on the concatenated vectors for each class prototype and query instance to generate relation scores between them.

3) Prototypical Network (Snell et al., 2017). This is an embedding-based model that uses the embedding of the sample mean as the class prototype. The model is discriminatively trained to minimize the relative Euclidean distance between each instance and its correct class embedding to other class embeddings.

4) Discriminative Variational Set Embedding (DiVaSE). Our novel set-embedding model that encodes a set into a distribution, while discriminating between classes as well. We experiment with two versions of our model - one with context loss in equation 7 which enables to explicilty learn to generalize to unseen classes, and the other without it.

Datasets We next introduce the datasets we used.

1) Omniglot. This dataset consists of 1623 hand-written characters, and each of the classes has 20 instances. We followed Vinyals et al. (2016) to augment and split the data, resulting in 4800 classes for training and 1692 classes for testing. We use a network with 2 fully connected layers each of 500 dimension to generate embedding vectors or set vectors.

2) OMNIST (Omniglot + MNIST). This dataset consists of 120 classes, where 110 classes are from Omniglot and 10 classes from MNIST. Each class from Omniglot has 10 instances, whereas each class from MNIST has 50 instances. Thus the resultant dataset is quite imbalanced. We split the total classes into 55/10/55 classes for train/validation/test set, and each set has almost the same ratio of classes between Omniglot and MNIST. We use the same network used in Omniglot dataset.

3) AwA. This dataset Lampert et al. (2009) consists of 30,475 images of 50 animal classes. We split total classes into 20/10/20 classes for train/validation/test set. We use a network with 4 convolutional blocks and batch normalization, following the work done by Vinyals et al. (2016).

Implementation Details We use Adam optimizer (Kingma & Ba, 2014) for all the experiments. The learning rate starts from 10^{-3} and multiplied by $1/2$ for every quarter of total length of episodes, which ranges from 30,000 to 40,000.

4.1 Results and Analysis

Table 1 shows the result accuracies on Omniglot dataset, and Table 2 and Table 3 shows the results on OMNIST and AWA dataset, respectively. First of all, for all the experiments, Neural Statistican suffers from poor generalization on unseen classes, especially when too many number of classes are considered at a time. The added hierarchical dependencies between latent variables Edwards & Storkey (2016) might be beneficial, but then the number of parameters will not be comparable to

Table 2: **(OMNIST) Generalization performance on unseen classes in test error (%).**

	5-way		20-way	
Models	1-shot	5-shot	1-shot	5-shot
Neural Statistician	56.78 ± 0.79	71.03 ± 0.98	28.59 ± 1.53	43.84 ± 1.81
Relation Network	57.61 ± 0.87	70.57 ± 0.88	29.47 ± 0.69	42.98 ± 0.49
Prototypical Network	57.99 ± 0.91	73.77 ± 1.04	30.63 ± 0.63	47.74 ± 0.98
DiVaSE (w/o context)	59.42 ± 0.07	74.14 ± 0.36	31.06 ± 0.19	47.93 ± 0.15
DiVaSE (w/ context)	$\mathbf{60.86 \pm 0.88}$	$\mathbf{75.73 \pm 1.12}$	$\mathbf{32.66 \pm 0.62}$	$\mathbf{48.59 \pm 0.83}$

Table 3: **(AWA) Generalization performance on unseen classes in test error (%).**

	5-way		20-way	
Models	1-shot	5-shot	1-shot	5-shot
Neural Statistician	31.95 ± 1.57	42.25 ± 1.95	10.27 ± 0.62	16.18 ± 0.37
Relation Network	41.18 ± 0.49	61.50 ± 1.22	14.86 ± 1.02	27.00 ± 0.84
Prototypical Network	40.63 ± 0.53	66.89 ± 0.73	14.85 ± 1.60	32.88 ± 1.63
DiVaSE (w/o context)	41.62 ± 0.09	67.94 ± 0.95	14.77 ± 1.74	34.38 ± 2.08
DiVaSE (w/ context)	$\mathbf{42.15 \pm 0.45}$	$\mathbf{68.17 \pm 1.01}$	$\mathbf{14.97 \pm 1.21}$	$\mathbf{35.28 \pm 1.25}$

other models. Prototypical Network and Relation Network performs better, but also shows degraded accuracies compared to our DiVaSE, especailly in the 1-shot learning. This is because our DiVaSE model can learn variance to capture the uncertainty of each prototype set, and the variances also help to better discriminate between classes Moreover, DiVaSE with context loss in equation 7 performs even better. This is because the model has been meta-learned to generalize on unseen classes. The learned variances for DiVaSE with context loss seems more compact than the variances without it, meaning that the model has learned quite *conservative* embeddings in preparing the simulated unseen classes at meta-train time.

5 Conclusion

We proposed a novel meta-learning framework for few-shot learning that embeds support sets for each class as distributions rather than points, and explicitly targets generalization performance, that is trained using variational inference. By meta-learning over large number of episodes, our discriminative variational set embedding (DiVaSE) learns variance of a Gaussian distribution along with its mean with generative and discriminative objectives, where the latter contains both set-wise and element-wise discriminative loss based on KL-divergence. Our method's ability to embed a set into a distribution allowed the embedding to capture intrinsic variance in the class, and identify which direction of the variance should be suppressed or allowed for class discrimianation. It also captures the uncertainty of the estimate which could be high in few-shot learning cases, and thus obtained better performance than point estimate especially when the number of training examples per class is low. Finally, by representing the classes as distributions, we can discriminate between two classes efficiently simply by maximizing the divergence between two distributions, and by exploiting this point, we proposed to explicitly *learn to generalize to unseen classes* by augmenting the learning objective with set-wise discrimination loss between unseen classes not selected for meta-learning. We validate our model on multiple datasets for few-shot classification, whose results show that the base DiVaSE model significantly outperforms the baselines, and achieves even higher performance with meta-learning to generalize to unseen classes.

References

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-F ei. Imagenet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

Harrison Edwards and Amos Storkey. Towards a neural statistician. *arXiv preprint arXiv:1606.02185*, 2016.

C. Finn, P. Abbeel, and S. Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *ICML*, 2017.

C. Finn, K. Xu, and S. Levine. Probabilistic Model-Agnostic Meta-Learning. *ArXiv e-prints*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

T. Kim, J. Yoon, O. Dia, S. Kim, Y. Bengio, and S. Ahn. Bayesian Model-Agnostic Meta-Learning. *ArXiv e-prints*, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 27*, pp. 3581–3589. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5352-semi-supervised-learning-with-deep-generative-models.pdf.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, 2012.

Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 951–958. IEEE, 2009.

Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriella Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(11), November 2013.

Danilo Jimenez Rezende, Shakir Mohamed, Ivo Danihelka, Karol Gregor, and Daan Wierstra. One-shot generalization in deep generative models. *arXiv preprint arXiv:1603.05106*, 2016.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pp. 1842–1850, 2016.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 4080–4090, 2017.

Sebastian Thrun. *Lifelong Learning Algorithms*, pp. 181–209. Springer US, Boston, MA, 1998.

Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pp. 3630–3638, 2016.

Kilian Q. Weinberger and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.*, 10, June 2009.

Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart Russell. Distance metric learning, with application to clustering with side-information. In *Proceedings of the 15th International Conference on Neural Information Processing Systems*, NIPS'02, 2002.

Flood Sung Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. 2018.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in Neural Information Processing Systems*, pp. 3394–3404, 2017.