Learning to Adapt With Memory for Probabilistic Few-Shot Learning

Lei Zhang, *Member, IEEE*, Liyun Zuo, Yingjun Du, and Xiantong Zhen

***Abstract*— Few-shot learning has recently generated increasing popularity in machine learning, which addresses the fundamental yet challenging problem of learning to adapt to new tasks with the limited data. In this paper, we propose a new probabilistic framework that learns to fast adapt with external memory. We model the classifier parameters as distributions that are inferred from the support set and directly applied to the query set for prediction. The model is optimized by formulating as a variational inference problem. The probabilistic modeling enables better handling prediction uncertainty due to the limited data. We impose a discriminative constraint on the feature representations by exploring the class structure, which can improve the classification performance. We further introduce a memory unit to store task-specific information extracted from the support set and used for the query set to achieve explicit adaption to individual tasks. By episodic training, the model learns to acquire the capability of adapting to specific tasks, which guarantees its performance on new related tasks. We conduct extensive experiments on widely-used benchmarks for few-shot recognition. Our method achieves new state-of-the-art performance and largely surpassing previous methods by large margins. The ablation study further demonstrates the effectiveness of the proposed discriminative learning and memory unit.**

***Index Terms*— Few shot learning, external memory, variational inference.**

I. Introduction

HUMANS have the instinct to effortlessly learn new concepts from a few examples and show great generalization ability to new tasks. However, existing machine learning models, e.g., deep neural networks (DNNs) [54], [59], rely heavily on large-scale annotated training data in order to achieve satisfactory performance. The huge gap between human intelligence and DNNs motivates us to try and progress the task of learning from a few samples, a.k.a. few-shot learning [19], [27], [40].

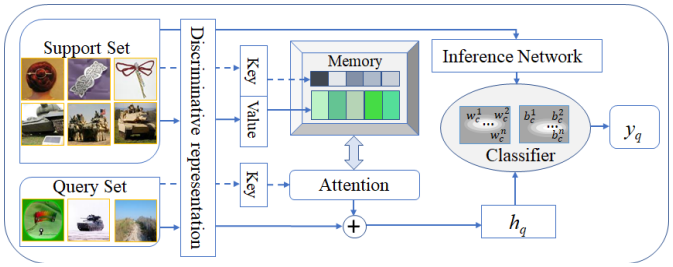


Fig. 1. Illustration of the proposed model augmented with memory for few-shot classification. For each task, information extracted from the support set is stored in the memory. The query set accesses the memory using an attention mechanism to achieve fast adaption.

Learning to recognize objects with only a few labeled samples pose great challenges to conventional machine learning models [32]. In order to deal with limited samples learning, meta-learning [51] has recently regained popularity in the machine learning community. By transferring knowledge learned from past related tasks to the new one, meta-learning provides a promising tool for few-shot learning. Generally speaking, meta-learning algorithms try to extract general knowledge from previous tasks [29], [40] that can quickly adapt to new tasks to enhance the performance of learners. In addition, the prediction can be largely uncertain since the model is learned from only a few labeled samples. To tackle the uncertainty, probabilistic models have been explored recently, showing great promise for few shot learning [11], [15]. We in this work develop a new probabilistic framework for few-shot learning, and further augment it with the memory mechanism.

Neural networks with external memory [16], [17], [55] have generated increasing attention in the machine learning community, which is inspired by the function of memory in human intelligence. External memory with the write and read modules has been explored in the meta-learning framework [35], [44], showing great promise in few-shot learning. Memory offers an effective way of leveraging context information for reasoning during the learning process. Nevertheless, there are still some open questions when using the external memory. On the one hand, what information should be stored in memory, so that it can be reliably accessed when needed. On the other hand, how does the content in memory improve the classifier performance with few samples.

In this paper, we propose a probabilistic learning model augmented with external memory for few-shot learning. The

Manuscript received September 11, 2020; revised November 12, 2020 and December 20, 2020; accepted January 6, 2021. Date of publication January 19, 2021; date of current version October 28, 2021. This work was supported in part by the Natural Science Foundation of China (NSFC) under Grant 61871016 and Grant 61976060, in part by the Department of Education of Guangdong Province under Grant 2018KCXTD019, and in part by the Open Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China, under Grant ICT 20061. This article was recommended by Associate Editor Z. Li. *(Corresponding author: Xiantong Zhen.)*

Lei Zhang and Liyun Zuo are with the College of Computer Science, Guangdong University of Petrochemical Technology, Maoming 525000, China.

Yingjun Du is with the Informatics Institute, University of Amsterdam, 1012 WX Amsterdam, The Netherlands.

Xiantong Zhen is with the College of Computer Science, Guangdong University of Petrochemical Technology, Maoming 525000, China, and also with the Inception Institute of Artificial Intelligence, Abu Dhabi, United Arab Emirates (e-mail: zhenxt@gmail.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2021.3052785.

Digital Object Identifier 10.1109/TCSVT.2021.3052785

overview of the proposed model is shown in Fig. 1. For each task, we use the support set to generate the classifier parameters by an inference network and at the same time extract information from the support set to store in an external memory unit. The query set achieves fast adaption to the task by addressing content in the memory through an attention mechanism. The adapted representations are fed to the classifier to make predictions. We make three major contributions:

- We propose a new probabilistic model for few-shot learning. The model is learned by formulating the optimization as a variational inference problem. In contrast to previous work, we introduce the extra condition on the support set in the likelihood term and the prior distribution, which enables our probabilistic modelling to offer a principled way to fully leverage information from the support set.
- We introduce an external memory unit into the probabilistic model, which enables efficient adaptation of feature representations from support set to new tasks. The introduced memory is well coupled with the probabilistic modeling and incorporates information from the support set to achieve fast adaptation. To the best of our knowledge, this is the first work that deploys a memory module to achieve adaption of feature representations for query samples in few-shot learning.
- We introduce a new discriminative loss to the optimization objective, which is seamlessly integrated into the probabilistic modeling. To the best of our knowledge, this has not been explored for probabilistic few-shot learning. In our ablation study, we experimentally demonstrate that the introduced discriminative loss can consistently improve recognition performance.

We conduct extensive experiments on three widely-used benchmark datasets for few-shot learning. We achieve new state-of-the-art performance on all benchmark datasets, which is consistently better than counterpart meta-learning algorithms including those also using external memory. We also perform an ablation study to gain insight into the proposed model, which shows the great effectiveness of the introduced discriminative constraint and memory units for fast adaption.

II. Related Work

1) Few-Shot Learning: There are diverse researches on few-shot learning. We focus on methods using the supervised meta-learning paradigm. Here, meta-learning aims to accumulate experience from learning multiple tasks that can quickly adapt to a new task with a handful of examples.

Metric learning [19], [46] has been explored for few-shot learning, which assumes that different related tasks would share the same metric to measure similarity. Vinyals *et al.* [53] proposed the matching network, which learns to map a small labeled support set and an unlabelled example to its label, obviating the need for fine-tuning to adapt to new class types. This work was originally developed for one-shot learning, and extended to a few-shot setting by Snell *et al.* [46]. They proposed the prototypical network to learn a metric space in which classification can be performed by computing distances to prototype representations of each class and the prototype of each class is the cluster of samples in that class. To enhance the expressivity of the prototypes, Allen *et al.* [1] proposed infinite mixture prototypes to adaptively represent both simple and complex data distributions for few-shot learning. Satorras *et al.* [45] solved few-shot learning with the prism of inference on a partially observed graphical model.

Optimization algorithms have been developed for few-shot learning. The core idea is to learn an optimization procedure that is shared across tasks, which can be applied to new tasks for quick adaptation, once learned in the meta-train stage. Ravi *et al.* [40] proposed an LSTM based meta-learner to learn the exact optimization algorithm used to train a neural network classifier in the few-shot regime; in their methods, apart from the learned optimization algorithm, a good initialization of model parameters is also obtained after training. Finn *et al.* [10] proposed a model agnostic meta-learning (MAML) algorithm, which was built upon the assumption that related tasks sharing initial parameters of neural networks could be adapted to specific tasks with a few steps of gradient descent updates. It was believed that the initial weights combined with a few more steps of gradient descent can approximate any learning algorithm, and thus gradient-based meta-learning had a number of practical benefits. Finn *et al.* [11] extended MAML in a probabilistic framework. Rajeswaran *et al.* [38] developed the implicit MAML [9] algorithm, which depended only on the solution to the inner level optimization. Zintgraf *et al.* [60] updated context parameters with one or several gradient steps on a task-specific loss that serves as an additional input to the model and were adapted on individual tasks.

Explicitly designing a meta-learner to learn a base-learner has also been studied. Bertinetto *et al.* [2] explored the feasibility of incorporating fast solvers with closed-form solutions as the base learning component of a meta-learning system. Gordon *et al.* [15] developed meta-learning approximate probabilistic inference for prediction. The support set was used to produce the parameter distribution of the classifier, which was applied to the query set for prediction. Mishra *et al.* [33] proposed a generic meta-learner architecture that used a novel combination of temporal convolutions and soft attention. Memory has recently generated increasing attention in the machine learning community, which was used to augment deep neural networks [17], [55]. It has also been introduced to the meta-learning framework for few-shot learning [34], [44].

2) Deep Latent Variable Models: A large family of deep latent variable model has arisen. By using Gaussian latent variables, [47] allowed for fast prediction using stochastic feed-forward inference. Reference [42] married ideas from deep neural networks and approximate Bayesian inference to derive a generalised class of deep, directed generative models. Reference [24] adopted generative model for effective generalisation from small labelled data sets to large unlabelled ones. Among those deep latent variable models, the variational autoencoder (VAE) gained increasing attention. Furthermore, VAE was extended to the conditional variational auto-encoder (CVAE) [47] for supervised learning. The neural statistician [7], which can be regarded as a complex version of CVAE, incorporated a global latent variable that captures the global uncertainty over a set. [30] adopted CVAE to generate

blur image. Recently, a new class of latent variable models has been introduced [13], [14], [21], called neural processes, which married the worlds of neural networks and Gaussian processes.

3) Memory: Memory plays an essential role in intelligent agents to gain extensive reasoning abilities. Currently, the researches focus on combining memory into the neural network. Sukhbaatar *et al.* [48] developed memory networks that could reason with a long-term memory module via reading and write. Munkhdalai *et al.* [34] proposed MetaNet as a new model while this model was based on memory and used of loss gradients as meta information. Wu *et al.* [56] introduced Bayesian relational memory to improve the generalization ability for semantic visual navigation agents in unseen environments. Huang *et al.* [18] proposed an aligned cross-modal memory model to memorize the rarely appeared content. Yu et. al. [57] designed a multimodal memory attentive network as an in-depth reasoning engine to enhance question answering quality.

The idea of using an external memory module has been explored and shown to be effective in few-shot learning. Santoro *et al.* [44] demonstrated the ability of a memory-augmented neural network to rapidly assimilate new data. Kaiser *et al.* [20] exploited fast nearest-neighbor algorithms for efficiency and thus scales to large memory sizes. Ramalho *et al.* [39] remembered the most surprising observations it has encountered to approximate probability distributions. Reference [34] proposed meta networks to learn a meta-level knowledge across tasks and shifts its inductive biases via fast parameterization for rapid generalization. Reference [35] proposed conditionally shifted neurons to modify the activation values of task-specific shifts retrieved from a memory module, which is populated rapidly based on limited task experience.

In contrast to previous work, we develop a new memory augmented probabilistic learning model for few-shot learning. A new discriminative constraint is introduced to further enhance classification performance. The external memory that stores information of the support set enables the feature representation of query set to be adapted to the specific task.

III. Probabilistic Few-Shot Learning

We present our probabilistic model with memory in the meta-learning framework for few-shot classification. In our model, we treat the classifier parameter W as the stochastic latent variable in our probabilistic modeling. To infer W, we derive a new evidence lower bound (ELBO), based on which we formulate the optimization of the model as a variational inference problem. To infer the posterior $p(W|\mathbf{x}_q, S)$ over W on support set S and query $\mathbf{x}_q$, which is generally intractable, we resort to using a variational distribution $q_\phi(W|S)$ to approximate it under the meta-learning framework.

This section is structured as follows: we describe the problem definition in § III-A, and our proposed probabilistic model in § III-B, which includes variational inference to approximate posterior distribution and amortization technique across classes. To further enhance the performance, we introduce the discriminative constraint in § III-C and augment it with an external memory in § III-D. Finally, learning and prediction algorithms by the proposed probabilistic model are given in § III-E.

A. Problem Definition

Few-shot classification is commonly learned by constructing few-shot tasks from a large dataset and optimizing the model parameters on these tasks. A task, also called an episode, is defined as a N-way K-shot classification problem, which is comprised of the support S and query $\mathcal{Q}$ sets. The 'way' of the episode refers to the number of classes in the support, and the 'shot' of the episode refers to the number of examples of each class. 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. Episodic optimization [53] iteratively trains the model by taking one episode update at a time.

B. Probabilistic Modeling

We develop the model under the probabilistic framework [8], [11], [15], which models parameters of classifier W as distributions instead of fixed vectors. We infer W from the support set by an amortization inference network and apply it to the query set for prediction. In order to adapt to the current task, we introduce a memory unit, based on which we generate the feature representation of query images.

For a few-shot learning task, given the support set $S = \{\mathbf{x}_s, \mathbf{y}_s\}$, and a query sample $(\mathbf{x}_q, \mathbf{y}_q)$, we consider the predictive conditional distribution as

$$p(\mathbf{y}_q|\mathbf{x}_q, S; \Theta) = \iint p(\mathbf{y}_q|W, \mathbf{h}_q; \Theta)\, p(\mathbf{h}_q|M, \mathbf{x}_q; \Theta)\, p(W|S; \Theta)\, p(M|S; \Theta) dW d\mathbf{h}_q \quad (1)$$

where Θ encloses model parameters, which are shared across all tasks; M denotes the content stored in the memory; the parameter of classifier is denoted as W which are task specific; and $\mathbf{h}_q$ is the intermediate representation of $\mathbf{x}_q$, which is used to predict $\mathbf{y}_q$. In our model, we treat $\mathbf{h}_q$ as a deterministic variable, which depends on both its input $\mathbf{x}_q$ and the content in the memory M. The graphical illustration of the proposed model is shown in Fig. 2.

We treat the classifier parameter W as the stochastic latent variable in our probabilistic modeling. To infer W, we derive an evidence lower bound (ELBO), based on which we formulate the optimization of the model as a variational inference problem. To optimize the model, we introduce the parameterization of the model based on neural networks by using the amortization techniques, and we further introduce a discriminative constraint, which can enhance the classification performance.

1) Variational Inference: From a probabilistic perspective, the parameters of classifier can be obtained by maximizing the conditional predictive log-likelihood of samples from the query set $\mathcal{Q}$, given the support set S.

$$\max_p \sum_{(\mathbf{x}_q, \mathbf{y}_q) \in \mathcal{Q}} \log p(\mathbf{y}_q|\mathbf{x}_q, S) \quad (2)$$

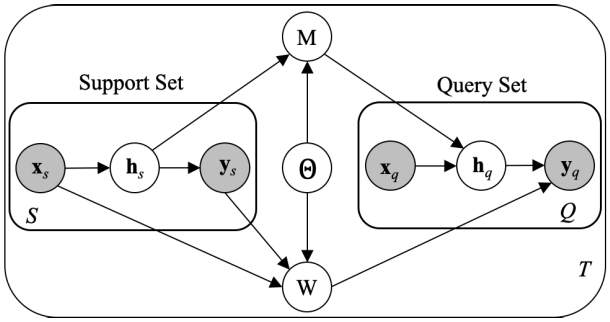


Fig. 2. Graphical illustration of the proposed probabilistic model with memory for few-shot classification. Θ encloses the global parameters of the model. **h** is the feature representation of the input **x**, and **y** is the associated class label. M denotes the memory unit and W is the parameter of classifiers.

$$= \max_{p} \sum_{(\mathbf{x}_q, \mathbf{y}_q) \in \mathcal{Q}} \log \int p(\mathbf{y}_q|\mathbf{x}_q, S, W) p(W|\mathbf{x}_q, S) dW. \tag{3}$$

We adopt a conditional prior distribution $p(W|\mathbf{x}_q, S)$ over the classifier parameters W as in the conditional variational auto-encoder (CVAE) [47] rather than an uninformative prior [23], [42]. By depending on the input $\mathbf{x}_q$, we infer W that can specifically represent the parameters of the classifier of the current data $\mathbf{x}_q$, while leveraging $\mathbf{x}_q$ to represent the samples of the current task by conditioning on the support set S.

To infer the posterior $p(W|\mathbf{x}_q, S)$ over W, which is generally intractable, we resort to using a variational distribution $q_\phi(W|S)$ to approximate it. In order to achieve $q_\phi(W|S)$ approximation, we consider the log posterior predictive distribution in terms of latent variables W as follows.

$$\log p(\mathbf{y}_q|\mathbf{x}_q, S) = \log p(W, \mathbf{y}_q|\mathbf{x}_q, S) - \log p(W|\mathbf{y}_q, \mathbf{x}_q, S) \tag{4}$$

By introducing the variational distribution $q_\phi(W|S)$, we can obtain

$$\log p(\mathbf{y}_q|\mathbf{x}_q, S) = \log \frac{p(W, \mathbf{y}_q|\mathbf{x}_q, S)}{q_\phi(W|S)} - \log \frac{p(W|\mathbf{y}_q, \mathbf{x}_q, S)}{q_\phi(W|S)} \tag{5}$$

Taking the expectation on both sides with respective to the variational distribution $q_\phi(W|S)$, we arrive at

$$\begin{aligned} &\log p(\mathbf{y}_q|\mathbf{x}_q, S) \\ &\quad = \mathbb{E}_{q_\phi(W|S)}[-\log q_\phi(W|S) + \log p(W, \mathbf{y}_q|\mathbf{x}_q, S)] \\ &\qquad + D_{\mathrm{KL}}[q_\phi(W|S)||p(W|\mathbf{y}_q, \mathbf{x}_q, S)] \end{aligned} \tag{6}$$

By dropping the KL term in (6) that is always non-negative, we can obtain the evidence lower bound (ELBO) as follows.

$$\begin{aligned} &\log p(\mathbf{y}_q|\mathbf{x}_q, S) \\ &\quad \geq \mathbb{E}_{q_\phi(W|S)}[-\log q_\phi(W|S) + \log p(W, \mathbf{y}_q|\mathbf{x}_q, S)] \\ &\quad = \mathbb{E}_{q_\phi(W|S)}[-\log q_\phi(W|S) + \log p(W|\mathbf{x}_q, S)] \\ &\qquad + \mathbb{E}_{q_\phi(W|S)}[\log p(\mathbf{y}_q|\mathbf{x}_q, S, W)] \\ &\quad = \mathbb{E}_{q_\phi(W|S)}[\log p(\mathbf{y}_q|\mathbf{x}_q, S, W)] \\ &\qquad - D_{\mathrm{KL}}[q_\phi(W|S)||p(W|\mathbf{x}_q, S)] = \mathrm{ELBO} \end{aligned} \tag{7}$$

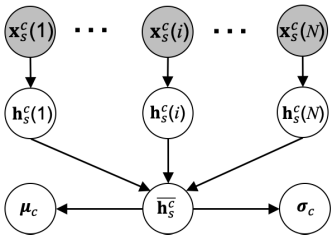


Fig. 3. The illustration of the amortized inference. The aggregated feature representation $\bar{\mathbf{h}}_s^c$ of samples in the same class by instance pooling is taken as input of the inference network, which produces the mean ($\boldsymbol{\mu}_c$) and variance ($\boldsymbol{\sigma}_c$) of the distribution of the classifier parameter.

The first term of the ELBO is the predictive log-likelihood conditioned on the observation $\mathbf{x}_q$, S and the inferred classifier W. Maximizing it enables us to make an accurate prediction for the query set $\mathbf{x}_q$. The second term in the ELBO minimizes the discrepancy between the meta variational distribution $q_\phi(W|S)$ and the meta prior $p(W|\mathbf{x}_q, S)$. Note that in contrast to regular conditional inference, our model introduces extra condition on the support set into both the predictive likelihood and the prior. The major benefit of the extra condition on information from the support set is that it enables the model to achieve fast adaption to the current task. In practice, we store the information of the support set in the memory, which is used in the incorporation of the extra condition on the support set into the inference of classifiers.

We now obtain the objective by maximizing the ELBO with respect to a batch of T tasks:

$$\begin{aligned} \mathcal{L}_{\mathrm{ELBO}} = \frac{1}{T} \sum_{t=1}^{T} \Big[&\sum_{(\mathbf{x}_q, \mathbf{y}_q) \in \mathcal{Q}^t} \frac{1}{L} \sum_{\ell} \log\, p(\mathbf{y}_q|\mathbf{x}_q, S^t, W_\ell^t) \\ &- D_{\mathrm{KL}}[q_\phi(W^t|S^t)||p(W^t|\mathbf{x}_q, S^t)] \Big] \end{aligned} \tag{8}$$

where S^t is the support set of the t-th task associated with its specific classifier W^t, $(\mathbf{x}_q, \mathbf{y}_q) \in \mathcal{Q}^t$ is the sample from the query set of the t-th task. Here we use Monte Carlo method to draw L samples of W from the variational posterior distribution $W_\ell^t \sim q_\phi(W|S^t)$.

2) Amortization: We implement the model with deep neural networks by using the amortization technique [23]. A deep convolutional neural networks are used to extract feature representations, where the networks are shared across tasks. Based on the feature representation, we deploy an inference network based on a multi-layer perceptron (MLP) to generate classifier with weight $W = \{\mathbf{w}_1, \ldots, \mathbf{w}_c, \ldots, \mathbf{w}_C\}$. If the classifier includes bias, then $\mathbf{b} = \{\mathbf{b}_1, \ldots, \mathbf{b}_c, \ldots, \mathbf{b}_C\}$ is also under consideration.

In order to reduce computational cost, we amortize the inference across classes, that is, the parameter distribution of the classifier is produced by the samples from the corresponding classes. To be more specific, as shown in Fig. 3, the amortization inference network takes the the average feature representations by instance pooling of samples in each class and produces the classifier distribution.

To summarize, the inference of classifier parameters from the support set using the amortized inference network can be conducted as follows.

- A deep convolutional network takes each individual image $\mathbf{x}_s$ and generate a feature representation vector $\mathbf{h}_s$.
- An permutation-invariant instance pooling layer collapses the matrix $(\mathbf{h}_s^c(1), \ldots, \mathbf{h}_s^c(N))$ to a single vector $\overline{\mathbf{h}_s^c}$. In our approach, we adopt average pooling in each class c.
- An inference network takes $\overline{\mathbf{h}_s^c}$ as the input and produces $\boldsymbol{\mu}_c$ and $\boldsymbol{\sigma}_c^2$, where we assume the distribution of the classifier parameter to be a diagonal Gaussian.

To enable the back-propagation for stochastic optimization with the sampling operation during training, we leverage the reparametrization trick [23] as follows:

$$\begin{aligned} \mathbf{w}_c &= \boldsymbol{\mu}_c^w + \boldsymbol{\sigma}_c^w \odot \boldsymbol{\epsilon}_1, \quad \text{where} \quad \boldsymbol{\epsilon}_1 \sim N(0, \mathbf{I}) \\ \mathbf{b}_c &= \boldsymbol{\mu}_c^b + \boldsymbol{\sigma}_c^b \odot \boldsymbol{\epsilon}_2, \quad \text{where} \quad \boldsymbol{\epsilon}_2 \sim N(0, \mathbf{I}) \end{aligned} \tag{9}$$

where $\odot$ is an element-wise product.

C. Discriminative Learning

The classification performance can be affected by the intra-class and inter-class structure of the learned feature representation space. Directly optimizing the above objective (8) would not guarantee the extracted features with sufficiently discriminative ability since the class structure is not taken into consideration in the learning process.

To guarantee high prediction performance, the distances between samples from different classes should be as far apart as possible, while the samples of the same classes should be as close as possible. To this end, we impose a constraint on the learned representation by exploring the class structure. As shown in Fig. 3, we adopt average pooling of samples in the classes and treat $\overline{\mathbf{h}}_c$ as the class centroid. We design the discriminative constraint as follows:

$$\begin{aligned} \mathcal{L}_{\mathrm{DC}} = \frac{1}{N_c C} \sum_{c=1}^{C} \sum_{i=1}^{N_c} [d(f(\mathbf{x}_{q,c}(i)), \overline{\mathbf{h}}_c) \\ + \exp(-\beta \sum_{c' \neq c} d(f(\mathbf{x}_{q,c}(i)), \overline{\mathbf{h}}_{c'}))] \end{aligned}$$

where $\beta > 0$ is the hyperparameter, $f(\cdot)$ is the feature extraction network, c' indicates a category different from $\mathbf{x}_{q,c}(i)$, N_c is the number of samples in the query set from class c, and $d(\cdot, \cdot)$ is the Euclidean distance.

To achieve the goal of augmenting discriminative ability, we introduce the $\mathcal{L}_{\mathrm{DC}}$ term as a regularizer in our objective function of (8). We obtain the objective function of discriminative learning as follows

$$\mathcal{L} = \mathcal{L}_{\mathrm{ELBO}} - \beta \mathcal{L}_{\mathrm{DC}} \tag{10}$$

where $\beta > 0$ is the hyperparameter. It is easy to check that the objective in (10) is still a valid evidence lower bound of the conditional predictive log-likelihood.

D. Fast Adaptation With Memory

Since the feature extraction network is shared across different tasks, no explicit adaption to specific tasks is used to take the speciality of the tasks into consideration. In this end, we introduce a memory unit to augment the model to enable adaption to individual tasks. To be more specific, we produce the adapted feature representations $\mathbf{h}_q$ of the query image by accessing the content M in the memory through an attention mechanism. By incorporating the memory M into (8), the corresponding objective function becomes:

$$\begin{aligned} \mathcal{L}_{\mathcal{M}} = \frac{1}{T} \sum_{t=1}^{T} \Big[\sum_{(\mathbf{x}_q, \mathbf{y})_q \in \mathcal{Q}^t} \frac{1}{L} \sum_{\ell} \log\, p(\mathbf{y}_q | \mathbf{h}_q(\mathbf{x}_q, M), W_\ell^t) \\ - D_{\mathrm{KL}}\big[q_\phi(W^t | S^t) || p(W^t | \mathbf{h}_q(\mathbf{x}_q, M))\big]\Big] \end{aligned}$$

where we use $\mathbf{h}_q(\mathbf{x}_q, M)$ to represent the adapted representation of $\mathbf{x}_q$ through memory. The condition on the support set S is reflected in the use of the memory M that is essentially generated by S.

By combing the discriminative constraint, we obtain the final objective function of our model as

$$\tilde{\mathcal{L}} = \mathcal{L}_{\mathcal{M}} - \beta \mathcal{L}_{\mathrm{DC}} \tag{11}$$

1) Memory Creation: As shown in Fig. 4, the memory unit is composed of pairs of keys and values, which are usually produced on support set.

$$M = (\mathbf{k}_1, \mathbf{v}_1), (\mathbf{k}_2, \mathbf{v}_2), \ldots, (\mathbf{k}_m, \mathbf{v}_m) \tag{12}$$

where m represents the memory size.

The key-value pair $(\mathbf{k}_i, \mathbf{v}_i)$ produced by the dedicated neural networks $f_K(\cdot)$ and $f_V(\cdot)$, respectively. Specifically, $\mathbf{k}_i$ and $\mathbf{v}_i$ in memory are generated as follows:

$$\mathbf{k}_i = f_K(\mathbf{x}_i); \qquad \mathbf{v}_i = f_V(f(\mathbf{x}_i)) \tag{13}$$

where $f(\cdot)$ is the shared convolutional network.

Due to the deployed memory unit, the task specific knowledge from the support set is extracted and stored in the memory. The memory structure in our work and [44] follows the same key-value pair paradigm in the neural Turing machine (NTM). However, we use memory to store the feature activations of all layers, which enables the model to further adapt feature representations of the query samples to the current task.

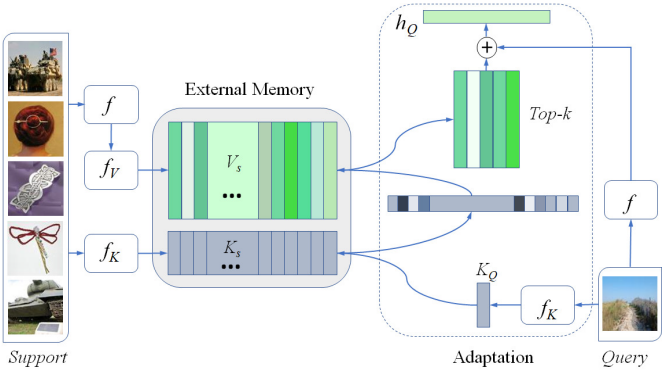


Fig. 4. Illustration of fast adaptation with external memory. $f(\cdot)$ is the shared convolutional neural network for feature extraction. $f_V(\cdot)$ is a multiple-layer perceptron (MLP) that generates the values for the memory unit. $f_K(\cdot)$ is a convolutional neural network that generates keys for the memory unit. $f_V(\cdot)$ extracts the knowledge from support set, which is stored in the memory and accessed by the query image to achieve adapted feature representation $\mathbf{h}_q$.

This is fundamentally different from [44], which stores final feature representations of the support samples and does not conduct any feature adaptation. In contrast to previous work using a shared feature extraction network, by deploying the extra step of adaptation, we are able to obtain more specific representations to the current task, therefore improving the performance.

2) Fast Adaptation: We use an attention module to access the content in the memory. As shown in Fig. 4, we compute the semantic relation between the query and the retrieved content from memory, the correlation score $\omega_{q,i}$ based on the attention mechanism using the keys can be computed for the similarity as

$$\omega_{q,i} = \frac{\exp(Sim(\mathbf{k}_q, \mathbf{k}_i))}{\sum_{j=1}^{m} \exp(Sim(\mathbf{k}_q, \mathbf{k}_j))} \tag{14}$$

where the cosine similarity is adopted, which is defined as

$$Sim(\mathbf{k}_q, \mathbf{k}_i) = \frac{\mathbf{k}_q \cdot \mathbf{k}_i}{||\mathbf{k}_q|| ||\mathbf{k}_i||}, \tag{15}$$

The top k similarities of $\{\omega_{q,i} | i \in \{1, \ldots, m\}\}$ are selected and then the weighted average of the retrieved memory content can be computed as

$$\mathbf{v}_q = \sum_{j=1}^{k} \omega_{q,j} \cdot \mathbf{v}_j \tag{16}$$

From (16), we can see that the obtained $\mathbf{v}_q$ contains the context information contained in the support set. This indeed makes $\mathbf{x}_q$ absorb the task specific information and therefore able to adapt to the current task, since all information about the current task is contained in the support set of the task.

$$\mathbf{h}_q = \tau \mathbf{v}_q + (1 - \tau) f(\mathbf{x}_q) \tag{17}$$

where τ is the hyper-parameter determined by cross validation.

E. Learning and Prediction

In the meta-training stage, the model parameter Θ that encloses weights of the neural networks ($f(\cdot)$, $f_V(\cdot)$ and $f_K(\cdot)$) and the parameter ϕ of the amortized inference network ($g(\cdot)$) are jointed optimized by stochastic gradient decent. The procedure of the meta-training is summarized in Algorithm 1. Once learned, the model is directly applied to the meta-test set for prediction, which does not involve any optimization. The inference procedure is described in Algorithm 2.

Algorithm 1 Learning in the Meta-Training Stage

Require: A set of tasks drawn from $p(\mathcal{T})$
Require: the learning rate λ; the number of iterations N_{iter}

Randomly initialize the parameters Θ
for *iter* in N_{iter} **do**
  Sample batch of tasks $\mathcal{T}_t \sim p(\mathcal{T})$;
  for all $\mathcal{T}_t$ **do**
    Sample S, $\mathcal{Q}$ from $\mathcal{T}_t$
    for $\mathbf{x}_s(i)$ in S where i in $1 : N$ **do**
      $\mathbf{h}_s(i) = f(\mathbf{x}_s(i))$, $\mathbf{v}_s(i) = f_V(\mathbf{h}_s(i))$, $\mathbf{k}_s(i) = f_K(\mathbf{x}_s(i))$
    end for
    for c in $1 : C$ **do**
      $\overline{\mathbf{h}_s^c} = \frac{1}{N_c} \sum_{i=1}^{N_c} \mathbf{h}_s^c(i)$;
      ${\boldsymbol{\mu}_c}^w, {\boldsymbol{\sigma}_c}^w = g_\phi(\overline{\mathbf{h}_s^c})$;
      ${\boldsymbol{\mu}_c}^b, {\boldsymbol{\sigma}_c}^b = g_\phi(\overline{\mathbf{h}_s^c})$;
      Obtain $\mathbf{w}_c$ and $\mathbf{b}_c$ according to (9);
    end for
    $W = [\mathbf{w}_1, \ldots, \mathbf{w}_c, \ldots, \mathbf{w}_C]$;
    $b = [\mathbf{b}_1, \ldots, \mathbf{b}_c, \ldots, \mathbf{b}_C]$
    For $\mathcal{Q}$, compute $\omega_{q,i}$, $\mathbf{v}_q$ in (14) and (16)
    $\mathbf{h}_q = \tau \mathbf{v}_q + (1 - \tau) f(\mathbf{x}_q)$;
    Compute $\tilde{\mathcal{L}}$ according to (11).
  end for
  Update parameters: $\mathbf{\Theta} \leftarrow \mathbf{\Theta} - \lambda \sum_{\mathcal{T}_t \sim p(\mathcal{T})} \nabla_\Theta \tilde{\mathcal{L}}$.
end for

IV. Experiments

In this section, we evaluate our model on standard few-shot classification tasks compared with previous work. We conduct experiments on three commonly-used benchmark datasets, i.e., *Omniglot* [27], *miniImageNet* [26], and *CIFAR-FS* [3]. We randomly sample C classes from the training classes, and for each class $(k+15)$ examples are randomly sampled. Thus, we can divide $C \times k$ examples into the support set and $C \times 15$ examples into the query set. The similar sampling strategy is also used in validation and test.

A. Datasets

To make comprehensive comparison with counterpart methods, we conduct experiments on three benchmark datasets widely-used in previous works.

Omniglot [27] contains 1623 handwritten characters (each with 20 examples). All characters are grouped in 50 alphabets. The training, validation, and testing are composed of a random split of [1100, 200, 423]. The dataset is augmented with rotations of 90 degrees, resulting in 4000 classes for training, 400 for validation, and 1292 for testing. All images are resized to 28×28.

miniImageNet [53] is a challenging dataset constructed from ImageNet [43], which comprises a total of 100 different classes (each with 600 instances). All these images have been downsampled to 84×84. We use the same splits of [40], where there are [64, 16, 20] classes for training, validation and testing.

CIFAR-FS [3] is adapted from the CIFAR-100 dataset [26] for few-shot learning. In the image classification benchmark CIFAR-100, there are 100 classes grouped into 20 superclasses (each with 600 instances). *CIFAR-FS* adopts the same split criteria as [64, 16, 20] classes for training, validation and testing. The resolution of all images is 32×32.

B. Experimental Settings

In our model, image features are extracted by a shallow convolutional neural network ($f(\cdot)$). We follow the experimental

Algorithm 2 Inference in the Meta-Test Stage

Require: Learned Θ

Require: A new task with $S=\{\mathbf{x}_s(i), \mathbf{y}_s(i)\}$; $\mathcal{Q}=\{\mathbf{x}_q\}$

for $\mathbf{x}_s(i)$ in S where i in $1:N$ **do**
  $\mathbf{h}_s(i) = f(\mathbf{x}_s(i))$, $\mathbf{v}_s(i) = f_V(\mathbf{h}_s(i))$, $\mathbf{k}_s(i) = f_K(\mathbf{x}_s(i))$
end for
for c in $1:C$ **do**
  $\overline{\mathbf{h}_s^c} = \frac{1}{N_c}\sum_{i=1}^{N_c} \mathbf{h}_s^c(i)$;
  ${\boldsymbol{\mu}_c}^w, {\boldsymbol{\sigma}_c}^w = g_\phi(\overline{\mathbf{h}_s^c})$;
  ${\boldsymbol{\mu}_c}^b, {\boldsymbol{\sigma}_c}^b = g_\phi(\overline{\mathbf{h}_s^c})$;
  Obtain $\mathbf{w}_c$ and $\mathbf{b}_c$ according to (9);
end for
$W = [\mathbf{w}_1, \ldots, \mathbf{w}_c, \ldots, \mathbf{w}_C]$;
$b = [\mathbf{b}_1, \ldots, \mathbf{b}_c, \ldots, \mathbf{b}_C]$
For $\mathcal{Q}$, compute $\omega_{q,i}$ and $\mathbf{v}_q$ by (14) and (16)
$\mathbf{h}_q = \tau\mathbf{v}_q + (1-\tau)f(\mathbf{x}_q)$;
Predict: $\mathbf{y}_q = -(W \cdot \mathbf{h}_q + b)$

protocol established by [15] for image classification. We do not use any fully connected layer for these CNNs. The key is generated by a 5 convolutional neural network $f_K(\cdot)$, each layer with the max-pooling operation. The memory network $f_V(\cdot)$ is of 3-layer fully connected network with 64 units per layer and rectifier non-linear activation function. For the value V, we first reshape the generated feature maps by the channel dimensions, i.e., feature maps [batch, width, height, channel] $\rightarrow$ [batch, width $\times$ height, channel]. Then we input the reshaped feature maps to the memory network $f_V(\cdot)$ to get the values V. As for memory size m, we set it as the size of support set.

We use the Adam optimizer [22] and a Gaussian form for $q_\phi(W_c|S)$. For the 5-way 5-shot model, we train 4 tasks per batch for 100, 000 iterations and use a constant learning rate of 0.0001. For the 5-way 1-shot model, the batch size is 8 tasks, and it keeps 150, 000 iterations and learning rate as 0.00025.

C. Results

We make a comprehensive comparison with state-of-the-art meta-learning algorithms for few-shot learning. Table I-III detail few-shot classification performances for our model as well as competitive approaches such as [2], [9], [15], [60]. It is worth to note that, in Table I-II, it focus on shallow feature extraction while for Table III, it verifies the influences on deep feature extraction.

For shallow feature extraction structure, in Table I, besides original MAML with 32 channels, we also implement the MAML (64C) with 64 channels in each convolutional layer. As for SNAIL approach [33], the original one is with a very deep ResNet-12 network. In order to make it comparable to our shallow network for feature embedding, we cite the result of SNAIL reported in [2] using similar shallow networks in Table I. Under the similar consideration, we also cite the original results of R2-D2 [2] using the same 64 channels for a fair comparison.

TABLE I

PERFORMANCE (ACCURACY IN %) ON *miniImagenet* AND *CIFAR-FS*

Method	miniImageNet, 5-way 1-shot	miniImageNet, 5-way 5-shot	CIFAR-FS, 5-way 1-shot	CIFAR-FS, 5-way 5-shot
MATCHING NET [53]	44.2	57	—	—
MAML [9]	48.7±1.8	63.1±0.9	58.9±1.9	71.5±1.0
MAML (64C)	46.7±1.7	61.1±0.1	58.9±1.8	71.5±1.1
META-LSTM [40]	43.4±0.8	60.6±0.7	—	—
PROTO NET [46]	47.4±0.6	65.4±0.5	55.5±0.7	72.0±0.6
RELATION NET [50]	50.4±0.8	65.3±0.7	55.0±1.0	69.3±0.8
SNAIL (32C) by [2]	45.1	55.2	—	—
GNN [12]	50.3	66.4	61.9	75.3
PLATIPUS [11]	50.1±1.9	—	—	—
VERSA [15]	53.3±1.8	67.3±0.9	62.5±1.7	75.1±0.9
R2-D2* [2]	50.5±0.2	65.4±0.2	62.3±0.2	77.4±0.2
R2-D2 [5]	51.7±1.8	63.3±0.9	60.2±1.8	70.9±0.9
CAVIA [60]	51.8±0.7	65.6±0.6	—	—
IMAML [38]	49.3±1.9	—	—	—
Baseline	53.1±1.7	67.1±0.8	62.1±0.9	75.7±0.8
w/ Memory	54.5±1.8	68.1±0.7	63.1±0.8	76.9±0.6
w/ DC	54.1±1.7	67.8±0.8	62.8±0.9	76.3±0.7
Full Model (ours)	**55.3**±1.8	**68.9**±0.8	**63.8**±0.7	**77.8**±0.9

*training with 20 ways, test on 5 ways.

TABLE II

PERFORMANCE (ACCURACY IN %) ON *Omniglot*

Method	Omniglot, 5-way 1-shot	Omniglot, 5-way 5-shot	Omniglot, 20-way 1-shot	Omniglot, 20-way 5-shot
SIAMESE NET [25]	96.7	98.4	88	96.5
MATCHING NET [53]	98.1	98.9	93.8	98.5
MAML [9]	98.7±0.4	**99.9**±0.1	95.8±0.3	98.9±0.2
PROTO NET [46]	98.5±0.2	99.5±0.1	95.3±0.2	98.7±0.1
SNAIL [33]	99.1±0.2	99.8 ±0.1	97.6 ±0.3	**99.4** ±0.2
GNN [12]	99.2	99.7	97.4	99.0
VERSA [15]	99.7±0.2	99.8±0.1	97.7±0.3	98.8±0.2
R2-D2 [2]	98.6	99.7	94.7	98.9
IMP [1]	98.4±0.3	99.5±0.1	95.0±0.1	98.6±0.1
Baseline	99.1±0.2	99.4±0.1	97.3±0.4	98.8 ±0.2
w/ Memory	99.5±0.1	99.8±0.1	98.1±0.3	99.0 ±0.1
w/ DC	99.3±0.2	99.7±0.2	97.9±0.2	98.9±0.1
Full Model (ours)	**99.8**±0.2	**99.9**±0.1	**98.3**±0.3	**99.2**±0.2

On the *miniImageNet* dataset, as shown in Table I, our model achieves new state-of-the-art results (55.3 - up 2.0% over the previous best) on 5-way - 1-shot and (68.9 - up 1.6%) on 5-way - 5-shot. On the *CIFAR-FS* dataset, it is demonstrated that our approach outperforms state-of-the-art (63.8 - up 1.3% over the previous best) on 5-way - 1-shot and (77.8 - up 0.4% over the previous best) on 5-way - 5-shot. Additionally, on the *Omniglot* dataset, our method achieves almost the best overall performance, as shown in Table II. All these results demonstrate the advantages of our model over previous methods. The compared methods including matching networks [53] and prototypical networks [46] adopt the nearest neighbor retrieval approach. Our method with external memory outperforms their performance consistently on all benchmark datasets. It is worth mentioning that our model consistently outperforms the recently proposed meta-learning model, i.e., VERSA, which is also based on probabilistic modeling. The major advantages of our approach lie in that our model is augmented with a discriminative constraint and a memory unit. In addition, our model is optimized via a variation inference formulation,

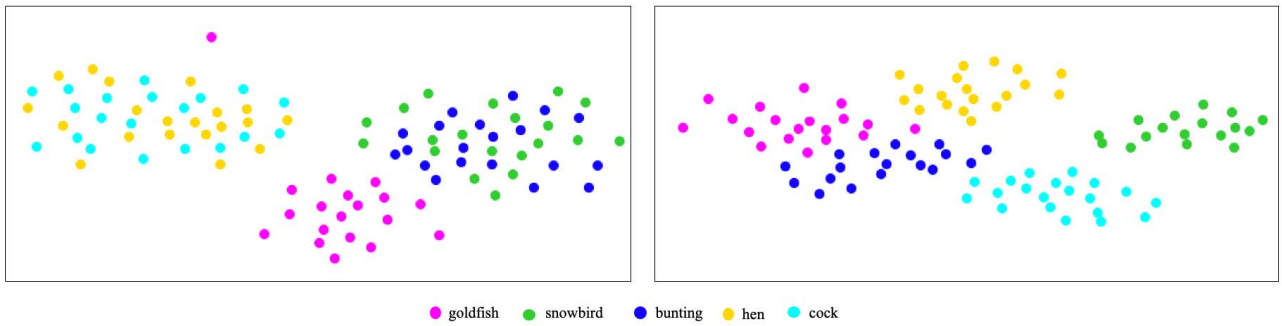


Fig. 5. Visualization of feature representations from *miniImageNet* without (Left) and with (Right) discriminative constraint. Different colors indicate different classes. Due to the discriminative constraint, the model can achieve more discriminative representations. Samples from the same classes tend to be clustered while those from different classes are scattered far apart.

TABLE III

PERFORMANCE (ACCURACY IN %) ON *miniImageNet* USING A DEEP FEATURE EXTRACTOR

	miniImageNet, 5-way	
	1-shot	5-shot
SNAIL [33]	55.71 ± 0.99	68.88 ± 0.92
AdaResNet [35]	56.88 ± 0.62	71.94 ± 0.57
TADAM [36]	58.50 ± 0.30	76.70 ± 0.30
Shot-Free [41]	59.04 ± n/a	77.64 ± n/a
TEWAM [37]	60.07 ± n/a	75.90 ± n/a
MTL [49]	61.20 ± 1.80	75.50 ± 0.80
Variational FSL [58]	61.23 ± 0.26	77.69 ± 0.17
MetaOptNet [28]	62.64 ± 0.61	78.63 ± 0.46
Diversity w/ Cooperation [6]	59.48 ± 0.65	75.62 ± 0.48
Meta-Baseline [4]	63.17 ± 0.23	79.26 ± 0.17
Tian et al. [52]	64.82 ± 0.60	82.14 ± 0.43
Baseline	63.67±0.57	80.28±0.44
w/ Memory	64.73±0.61	82.01±0.51
w/ Discriminative Constraint	64.12±0.55	81.78±0.41
Full Model (ours)	**65.91**±0.53	**82.91**±0.49

TABLE IV

BENEFIT OF THE KEY OF MEMORY IN (%) ON *miniImageNet* AND *CIFAR-FS*

	miniImageNet, 5-way		**CIFAR-FS, 5-way**	
Method	1-shot	5-shot	1-shot	5-shot
Baseline	53.1±1.7	67.1±0.8	62.1±0.9	75.7±0.8
w/o f_K	53.7±1.8	67.9±0.7	62.8±0.8	76.2±0.6
w/ f_K	**55.3**±1.8	**68.9**±0.8	**63.8**±0.7	**77.8**±0.9

in which the KL divergence term encourages the query set to leverage knowledge from the support set.

For fair comparison with previous works, we experiment with both shallow convolutional neural networks deep architecture for feature extraction. The results using ResNet-12 are shown in Table III. Our model using the deep network also achieves highest recognition accuracy, surpassing the second best method, i.e., Tian [52], by a margin of 1.09% under the 5-way 1-shot setting and 0.77% under the 5-way 5-shot setting. The consistent state-of-the-art results on all benchmarks using either shallow or deep feature extraction networks validate the effectiveness of our model for few-shot learning.

D. Ablation Study

We have also conducted extensive ablation studies to demonstrate the effectiveness of the proposed model for few-shot learning. We define the model using the objective in (8) as the baseline. We test the effect of the introduced discriminative learning and memory on the overal performance, respectively. The full model is the baseline with both the discriminative constraint and the memory unit. We compare with several alternative models on *miniImageNet* and *CIFAR-FS* in Table I, and *Omniglot* in Table II.

1) Benefit of Discriminative Constraint: To show the benefit of discriminative constraint, by comparing the baseline plus discriminative constraint and the Baseline model in Table I, there are (54.1- up 1.0%) on 5-way - 1-shot on the *miniImageNet* and (76.3- up 0.6%) on 5-way - 5-shot on the *CIFAR-FS*. For other conditions, baseline with discriminative constraint performs better than only baseline. Similar conclusion can be drawn from Table III when deep feature extraction structure are employed. The results indicate that the model with a discriminative constraint can extract the features with sufficiently discriminative ability, which can obtain high prediction performance.

To further look into the discriminative learning, we visualize the representation with and without the discriminative constraint for intuitive illustration. We use the t-SNE [31] to embed the feature representation into a two-dimensional space as shown in Fig. 5. As can be clearly seen, the model with the discriminative constraint can better scatter samples in the embedding space. Samples from the same classes tend to be clustered while those from different classes fall apart.

2) Benefit of Memory: To demonstrate the advantage of the proposed memory, we implement a baseline with only memory. The memory mechanism can enhance the performance in *miniImageNet* with 1.4% and 1.0% improvement for 5-way - 1-shot and 5-way - 5-shot, respectively. Similar performance improvements on *CIFAR-FS* and *Omniglot* are achieved as shown in Tables I, II and III, using either shallow or deep feature extraction structure. This shows that the benefit of the memory unit fast adaption to specific tasks, which transfers the knowledge from support set to query set. The full model with both of them achieves the best performance consistently on all datasets, which verifies the advantage of the proposed discriminative constraint and the memory unit.

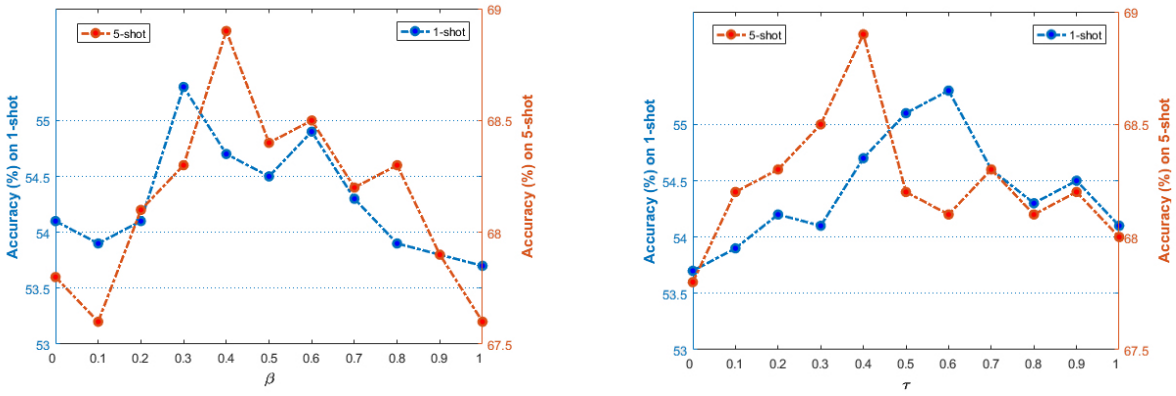


Fig. 6. Performance on *miniImageNet* under the 1-shot and 5-shot setting with different values of β (left) and τ (right).

To verify the effect of value f_K on the final result, we implemented the model with and without f_K, as shown in Table IV. On the *miniImageNet*, The model with f_K outperforms that without f_K by 1.6% and 1.0% on the 5-way-1-shot and 5-way-5-shot. The improvements on *CIFAR-FS* are also obvious with f_K. The results indicate that using f_K we can better find the closest samples.

3) Influence of the Hyper-Parameters β and τ: Our model involves two hyper-parameters β and τ, where β controls the importance of the discriminative constraint in the optimization, and τ indicates the extent of the adaptation by the memory. To study their influences on our model, we perform experiments on *miniImageNet* dataset on the variation set to evaluate the effect of these two hyper-parameters. We change one parameter at a time, while fixing the other one. The results are plotted in Figure 6. In our experiments, we set $\beta = 0.3, \tau = 0.6$ on 1-shot and $\beta = 0.4, \tau = 0.4$ on 5-shot.

V. Conclusion

In this paper, we presented a new probabilistic model for few-shot learning. We treated the classifier parameter as latent variables that were inferred from data by formulating as a variational inference problem. We further introduced a discriminative constraint by exploring the class structure in the learned feature representation space. To achieve adaption to specific tasks, we introduced a memory unit into the model to store the task-specific information extracted from its support set, which was used to achieve adapted feature representations of the query set. Our model consistently achieved high performance and advances the state of the art on three benchmarks. Comprehensive ablation studies validated the benefits of memory unit and discriminative learning for few-shot learning.

References

[1] K. R. Allen, E. Shelhamer, H. Shin, and J. B. Tenenbaum, "Infinite mixture prototypes for few-shot learning," 2019, *arXiv:1902.04552*. [Online]. Available: https://arxiv.org/abs/1902.04552

[2] L. Bertinetto, J. F. Henriques, P. H. Torr, and A. Vedaldi, "Meta-learning with differentiable closed-form solvers," 2019, *arXiv:1805.08136*. [Online]. Available: https://arxiv.org/abs/1805.08136

[3] L. Bertinetto, F. J. Henriques, J. Valmadre, H. S. P. Torr, and A. Vedaldi, "Learning feed-forward one-shot learners," in *Proc. NIPS*, 2016, pp. 523–531.

[4] Y. Chen, X. Wang, Z. Liu, H. Xu, and T. Darrell, "A new meta-baseline for few-shot learning," 2020, *arXiv:2003.04390*. [Online]. Available: http://arxiv.org/abs/2003.04390

[5] A. Devos, S. Chatel, and M. Grossglauser, "Reproducing meta-learning with differentiable closed-form solvers," in *Proc. ICLR*, 2019, pp. 1–8.

[6] N. Dvornik, J. Mairal, and C. Schmid, "Diversity with cooperation: Ensemble methods for few-shot classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3723–3731.

[7] H. Edwards and A. Storkey, "Towards a neural statistician," 2016, *arXiv:1606.02185*. [Online]. Available: http://arxiv.org/abs/1606.02185

[8] M.-Á. Fernández-Torres, I. González-Díaz, and F. Diaz-de-Maria, "Probabilistic topic model for context-driven visual attention understanding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1653–1667, Jun. 2020.

[9] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. ICML*, 2017, pp. 1126–1135.

[10] C. Finn and S. Levine, "Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm," in *Proc. ICLR*, 2018, pp. 1–20.

[11] C. Finn, K. Xu, and S. Levine, "Probabilistic model-agnostic meta-learning," in *Proc. NeurIPS*, 2018, pp. 9516–9527.

[12] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," 2018, *arXiv:1711.04043*. [Online]. Available: https://arxiv.org/abs/1711.04043

[13] M. Garnelo *et al.*, "Conditional neural processes," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1690–1699.

[14] M. Garnelo *et al.*, "Neural processes," 2018, *arXiv:1807.01622*. [Online]. Available: http://arxiv.org/abs/1807.01622

[15] J. Gordon, J. Bronskill, M. Bauer, S. Nowozin, and R. E. Turner, "Meta-learning probabilistic inference for prediction," 2019, *arXiv:1805.09921*. [Online]. Available: https://arxiv.org/abs/1805.09921

[16] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," 2014, *arXiv:1410.5401*. [Online]. Available: http://arxiv.org/abs/1410.5401

[17] A. Graves *et al.*, "Hybrid computing using a neural network with dynamic external memory," *Nature*, vol. 538, no. 7626, pp. 471–476, Oct. 2016.

[18] Y. Huang and L. Wang, "ACMM: Aligned cross-modal memory for few-shot image and sentence matching," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5774–5783.

[19] W. Jiang, K. Huang, J. Geng, and X. Deng, "Multi-scale metric learning for few-shot learning," *IEEE Trans. Circuits Syst. Video Technol.*, early access, May 20, 2020, doi: 10.1109/TCSVT.2020.2995754.

[20] Ł. Kaiser, O. Nachum, A. Roy, and S. Bengio, "Learning to remember rare events," 2017, *arXiv:1703.03129*. [Online]. Available: http://arxiv.org/abs/1703.03129

[21] H. Kim *et al.*, "Attentive neural processes," 2019, *arXiv:1901.05761*. [Online]. Available: http://arxiv.org/abs/1901.05761

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

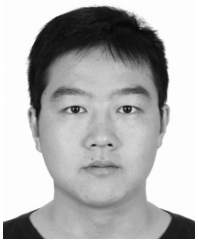
[23] D. P Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*. [Online]. Available: http://arxiv.org/abs/1312.6114

[24] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3581–3589.

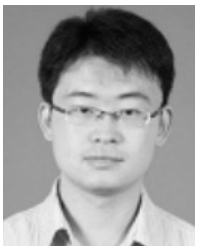
[25] G. Koch, "Siamese neural networks for one-shot image recognition," in *Proc. ICML Workshop*, 2015, pp. 1–8.
[26] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009.
[27] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, Dec. 2015.
[28] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10657–10665.
[29] Y. Liu *et al.*, "Learning to propagate labels: Transductive propagation network for few-shot learning," *Proc. ICLR*, 2019, pp. 1–14.
[30] Z.-S. Liu, W.-C. Siu, and Y.-L. Chan, "Photo-realistic image super-resolution via variational autoencoders," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Jun. 19, 2020, doi: 10.1109/TCSVT.2020.3003832.
[31] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
[32] F. Markatopoulou, V. Mezaris, and I. Patras, "Implicit and explicit concept relations in deep neural networks for multi-label video/image annotation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 6, pp. 1631–1644, Jun. 2019.
[33] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, "A simple neural attentive meta-learner," 2018, *arXiv:1707.03141*. [Online]. Available: https://arxiv.org/abs/1707.03141
[34] T. Munkhdalai and H. Yu, "Meta networks," in *Proc. ICML*, 2017, p. 2554.
[35] T. Munkhdalai, X. Yuan, S. Mehri, and A. Trischler, "Rapid adaptation with conditionally shifted neurons," 2017, *arXiv:1712.09926*. [Online]. Available: http://arxiv.org/abs/1712.09926
[36] B. Oreshkin, P. R. López, and A. Lacoste, "Tadam: Task dependent adaptive metric for improved few-shot learning," in *Proc. NeurIPS*, 2018, pp. 721–731.
[37] S. Qiao, C. Liu, W. Shen, and A. Yuille, "Few-shot image recognition by predicting parameters from activations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7229–7238.
[38] A. Rajeswaran, C. Finn, S. Kakade, and S. Levine, "Meta-learning with implicit gradients," 2019, *arXiv:1909.04630*. [Online]. Available: http://arxiv.org/abs/1909.04630
[39] T. Ramalho and M. Garnelo, "Adaptive posterior learning: Few-shot learning with a surprise-based memory module," *Proc. ICLR*, 2019, pp. 1–14.
[40] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *Proc. ICLR*, 2017, pp. 1–11.
[41] A. Ravichandran, R. Bhotika, and S. Soatto, "Few-shot learning with embedded class models and shot-free meta training," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 331–339.
[42] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. ACM Int. Conf. Mach. Learn.*, 2014, pp. 1–14.
[43] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[44] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1842–1850.
[45] V. G. Satorras and J. B. Estrach, "Few-shot learning with graph neural networks," in *Proc. ICLR*, 2018, pp. 1–13.
[46] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. NIPS*, 2017, pp. 4077–4087.
[47] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 3483–3491.
[48] S. Sukhbaatar, J. Weston, and R. Fergus, "End-to-end memory networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2440–2448.
[49] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 403–412.
[50] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1199–1208.
[51] S. Thrun and L. Pratt, *Learning to Learn*. Berlin, Germany: Springer, 2012.
[52] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola, "Rethinking few-shot image classification: A good embedding is all you need?" 2020, *arXiv:2003.11539*. [Online]. Available: http://arxiv.org/abs/2003.11539
[53] O. Vinyals, C. Blundell, T. Lillicrap, and D. Wierstra, "Matching networks for one shot learning," in *Proc. NIPS*, 2016, pp. 3630–3638.
[54] Q. Wang, J. Wan, and Y. Yuan, "Deep metric learning for crowdedness regression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2633–2643, Oct. 2018.
[55] J. Weston, S. Chopra, and A. Bordes, "Memory networks," 2015, *arXiv:1410.3916*. [Online]. Available: https://arxiv.org/abs/1410.3916
[56] Y. Wu, Y. Wu, A. Tamar, S. Russell, G. Gkioxari, and Y. Tian, "Bayesian relational memory for semantic visual navigation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2769–2779.
[57] T. Yu, J. Yu, Z. Yu, Q. Huang, and Q. Tian, "Long-term video question answering via multimodal hierarchical memory attentive networks," *IEEE Trans. Circuits Syst. Video Technol.*, early access, May 20, 2020, doi: 10.1109/TCSVT.2020.2995959.
[58] J. Zhang, C. Zhao, B. Ni, M. Xu, and X. Yang, "Variational few-shot learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1685–1694.
[59] W. Zhu, X. Wang, and H. Li, "Multi-modal deep analysis for multimedia," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 10, pp. 3740–3764, Oct. 2020.
[60] L. Zintgraf, K. Shiarli, V. Kurin, K. Hofmann, and S. Whiteson, "Fast context adaptation via meta-learning," in *Proc. ICML*, 2019, pp. 7693–7702.

Lei Zhang (Member, IEEE) received the Ph.D. degree in computer science from the Harbin Institute of Technology (HIT), Harbin, Heilongjiang, China, in 2004. She is currently a Professor with the Computer Science College, Guangdong University of Petrochemical Technology, Maoming, Guangdong, China. Her research interests include signal/image processing, computer vision, and machine learning.

Liyun Zuo received the Ph.D. degree in computer science from the South China University of Technology, Guangzhou, China, in 2016. She is currently a Professor with the Computer Science College, Guangdong University of Petrochemical Technology, Maoming, Guangdong, China. Her research interests include cloud computing, computer vision, and machine learning.



Yingjun Du received the B.E. degree from Hainan University, Hainan, China, in 2016, and the master's degree from the College of Software, Beihang University, Beijing, China, in 2020. He is currently pursuing the Ph.D. degree with the Informatics Institute, University of Amsterdam, The Netherlands. His research interests include machine learning and computer vision.



Xiantong Zhen received the B.S. and M.E. degrees from Lanzhou University, Lanzhou, China, in 2007 and 2010, respectively, and the Ph.D. degree from the Department of Electronic and Electrical Engineering, The University of Sheffield, U.K., in 2013. He worked as a Post-Doctoral Fellow with Western University, London, ON, Canada, and The University of Texas at Arlington, Arlington, TX, USA, from 2013 to 2017. He is currently with the Guangdong University of Petrochemical Technology, Maoming, Guangdong, China, and the Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, United Arab Emirates. His research interests include machine learning and computer vision.