A Bayesian Meta-learning Based Method for Few-Shot Hyperspectral Image Classification

Jing Zhang, Liqin Liu, Rui Zhao, and Zhenwei Shi* *Member, IEEE*

***Abstract*—Few-shot learning provides a new way to solve the problem of insufficient training samples in hyperspectral classification. It can implement reliable classification under several training samples by learning meta-knowledge from similar tasks. However, most existing works perform frequency statistics, which may suffer from the prevalent uncertainty in point estimates with limited training samples. To overcome this problem, we reconsider the hyperspectral image few-shot classification (HSI-FSC) task as a hierarchical probabilistic inference from a Bayesian view and provide a careful process of meta-learning probabilistic inference. We introduce a prototype vector for each class as latent variables and adopt distribution estimates for them to obtain their posterior distribution. The posterior of the prototype vectors is maximized by updating the parameters in the model via the prior distribution of HSI and labeled samples. The features of the query samples are matched with prototype vectors drawn from the posterior, thus a posterior predictive distribution over the labels of query samples can be inferred via an amortized Bayesian variational inference approach. Experimental results on four datasets demonstrate the effectiveness of our method. Especially given only $3-5$ labeled samples, the method achieves noticeable upgrades of overall accuracy against competitive methods.**

***Index Terms*—Bayesian Meta-learning, few-shot learning, hyperspectral image (HSI) classification, prototype vector.**

I. Introduction

HYPERSPECTRAL images (HSIs) contain abundant spectral and spatial information, which is beneficial to identifying ground objects effectively [1], [2]. Generally, hyperspectral image is widely used in many fields such as agriculture [3], forestry [4], and environmental monitoring [5]. Hyperspectral image classification (HSIC) aims to assign each HSI pixel a corresponding class label and has been studied extensively over the past decades. However, annotating HSI is time-consuming, expensive, and needs field knowledge which leads to the limited sample size of HSIC tasks.

There are mainly three solutions for the limited sample problem: feature extraction, data augment, and model learning. Many methods target to extract more efficient feature expression [6]–[8], including dimension reduction [9]–[12], manual image feature [13]–[16] and CNN feature extraction [17]–[19]. ABNet [20] modified the backbone with enhanced effective channel attention (EECA) to capture more discriminative features and proposes AFPN and CEM for multiscale features. LSLRR [21] aims to enable the learned representations to characterize the global and local structures of HSI, and to present a classwise block-diagonal matrix. However, the expression ability of the features is limited in the case of only few annotations.

Data augment expands the training set by leveraging prior knowledge either from the unlabeled samples or similar datasets. For example, Generative Neural Network (GAN) is widely used to generate pseudo samples and extend the training dataset [22], [23]. Meanwhile, many semi-supervised methods [24]–[26] choose some unlabeled samples and their predicted label as the addition to the training set. Besides, recent graph-based semi-supervised learning methods [27]–[31] have also achieved impressive performance on the task of few-shot classification of hyperspectral images. The limitation of these methods is that errors may accumulate and propagate due to unreliable sample-label pairs.

Model learning methods acquire general knowledge on massive learning tasks and utilize the knowledge to solve new tasks, such as transfer learning [32]–[34] and meta-learning [35]–[37]. Transfer learning transfers knowledge learned from the source domain to the target domain and is difficult to overcome domain gap [32]–[34]. By constructing a large number of similar meta-tasks as compensation for few-shot, meta-learning provides a new solution for few-shot tasks, especially for those that only have $3-20$ labeled samples per class [38]–[41]. Traditional machine learning methods learn a mapping from data to label during the training stage and adopt the learned mapping to predict the labels of unlabeled samples. In contrast, meta-learning leverages massive meta-tasks to learn a common solution of this kind of task during the training stage, then adopts it to solve the testing task during the testing stage.

Meta-learning-based few-shot learning methods have been attracting growing interest in hyperspectral image classification. On the one hand, practical HSIC application scenarios mainly suffer from few-shot constrain due to the high annotation difficulty and cost. On the other hand, although methods like transfer learning and semi-supervised methods have made progress in reducing the need for sample size, it is still difficult for them to handle few-shot tasks. A deep learning Few-shot learning method (DFSL) is designed for HSIC [38], it learns a metric space where training samples of the same class are close to each other and the metric

The work was supported in part by the National Natural Science Foundation of China under Grant 62125102, in part by the National Key Research and Development Program of China (Titled "Brain-inspired General Vision Models and Applications"), and in part by the Fundamental Research Funds for the Central Universities. *(Corresponding author: Zhenwei Shi (e-mail: shizhenwei@buaa.edu.cn))*

Jing Zhang, Liqin Liu, and Zhenwei Shi are with Image Processing Center, School of Astronautics, Beihang University, Beijing 100191, China, and with Beijing Key Laboratory of Digital Media, Beihang University, Beijing 100191, China, and also with State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China and also with Shanghai Artificial Intelligence Laboratory, Shanghai 200232, China.

Rui Zhao is with the Fuxi AI Laboratory, NetEase, Hangzhou, Zhejiang 310052, China

space can work well on unseen samples. The probability that the test sample belongs to each class is obtained by calculating the Euclidean distance to prototype vector [42] of each class. Relation network [43] evaluates the relationship between spatial-spectral features of different samples extracted by embedding model [39], [44]. Deep cross-domain few-shot learning (DCFSL) can handle the domain shift via the conditional adversarial domain distribution strategy [41]. Zhou et al. [45] apply the meta-learning to hyperspectral few-shot tasks, propose to train a linear classier on several labeled samples, and optimize an embedding module by minimizing the test error on query samples for each generated meta-task. SSFT [46] utilizes a 3-D residual network with ECA module to extract features, followed by a feature transformation module to enhance feature diversity. SVD is applied to the transformed features of each class, and then the subspace of each class is obtained.

However, the spectra of the same object are variable and different objects may have similar spectra, which brings great challenges to few-shot classification methods. As shown in fig.1, three samples randomly selected from 'Gravel' in Pavia University data set present different spectra, as do the three samples from 'Self-Blocking Bricks'. Meanwhile, samples from the two classes show some similarities. The existing few-shot HSIC methods are based on frequency statistics and directly determine the model by maximizing local/global likelihood in the training stage, without paying enough attention to the spectral variation phenomenon. Point estimates under the constraints of a few labeled samples can cause prevalent uncertainty, which may cause a large deviation in label prediction.

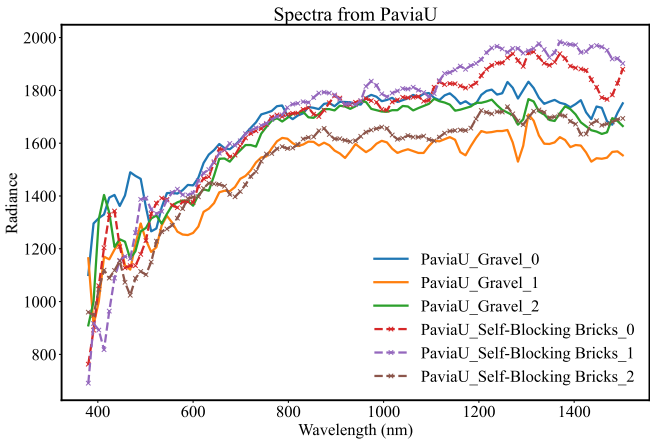


Fig. 1: The prevalent phenomenon of samples from the same class with different spectra and samples from different classes with similar spectra in hyperspectral images.

To alleviate the contradiction between few-shot and spectra variation, we re-examine the few-shot HSI classification from a Bayesian view and develop a Bayesian Meta-learning Few-Shot Classification solution (BMFSC), which conducts distribution estimates for prototype vectors rather than point estimates, as shown in fig. 2. Some researchers have suggested that the human ability for few-shot inductive reasoning could derive from a Bayesian inference mechanism [47], [48]. The

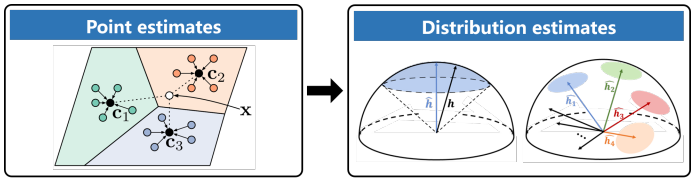


Fig. 2: (a) Prototype networks make point estimates on prototype vectors, that is, a deterministic vector is obtained. The result can be explicitly determined by the local/global likelihood maxima of the training phase. (b) We make distribution estimates on prototype vectors and assume that spectra of the same class generally follow the same distribution. The label of a query sample can be determined by comparing the probability that its spectrum belongs to the corresponding distribution of each class.

bayesian method learns the posterior distribution of model parameters from a trade-off between its prior distribution and the conditional likelihood of the observed data and is believed to make better inference [49], [50]. Bayesian meta-learning combines the advantages of meta-learning and Bayesian learning, learning task priors on massive similar meta-tasks and obtaining the posterior distribution from the task priors and data likelihood. Meanwhile, the bayesian meta-learning method can also reduce the occurrence of over-fitting of inner and outer optimization [51].

In BMFSC, we design a hierarchical probability model and introduce a latent variable to factorize the posterior predictive distribution of query set as integral of posterior for each latent vector. The latent variable represents the prototype vector of each class and its posterior is maximized by updating the parameters of the inference network based on the prior distribution of HSI and labeled samples. During the query stage, features of the query samples are matched with each prototype vector drawn from the posterior, thus a posterior predictive distribution over the labels of query samples can be inferred via an amortized Bayesian variational inference approach. Experimental results prove the effectiveness of our proposed approach against competitive methods.

The contribution of our work can be summarized as follows:

1) We design a Bayesian meta-learning method for hyperspectral few-shot classification and formalize the problem in a probabilistic way. The predictive probability model arises from a trade-off between its prior distribution and the conditional likelihood of the observed data.
2) We construct a hierarchical probability model conditioned on the common feature of the samples from each class, called a prototype vector of this class. We use distribution estimates for prototype vectors to cope with the contradiction between a few labeled samples and the spectral variation.

The rest of this paper is organized as follows. In section II, we introduce related work briefly. In section III, we give a process of meta-learning probabilistic inference in detail and introduce our proposed few-shot classification method for

hyperspectral images. Details about the experimental setup and results are reported in section IV. The conclusions are given in Section V.

II. Related Work

A. Bayesian Models

Given a set of observations $O = (X, Y) = \{(\boldsymbol{x}_i, y_i)\}_{i=0}^{n-1}$ where $\boldsymbol{x}_i \in X$ is the data and $y_i \in Y$ is the label, let H be a hypothesis space of all mapping functions $h : X \to Y$. Each $h \in H$ can be regarded as a concrete possible description of the model. Drawing h from a prior distribution $p(h)$ and then relating it to the observations O through the likelihood $p(O|h)$, a Bayesian model infers conditioning on data and computing the posterior:

$$p(h|O) = \frac{p(O|h)p(h)}{p(O)} \tag{1}$$

The posterior prediction distribution on labels y_* of testing data $\boldsymbol{x}_*$ can be achieved by

$$p(y_*|O, \boldsymbol{x}_*) = \int p(y_*|h, \boldsymbol{x}_*)p(h|O, \boldsymbol{x}_*)\, dh \tag{2}$$

In complex Bayesian models, the posterior calculation usually requires approximate inference methods, such as MCMC [52] and variational inference [53].

B. Bayesian Meta-learning

Deep learning has made amazing progress in the past decades with the support of strong computing power, rich data, and advanced solutions. However, there is still a long way to go before leveraging neural networks in practical scenarios where the data is scarce and requirements for accuracy are critical. Meta-learning, which is known as learning how to learn, can learn information from massive similar meta-tasks and generalize that information to unseen tasks.

The comparison of the traditional machine learning paradigm and meta-learning paradigm is shown in fig. 3. Take a testing task to classify unlabeled samples into Alfalfa, Corn and etc. for example. To solve this task, traditional machine learning is to learn a mapping function $h : X \to Y$ on the training set consisting of labeled samples from Alfalfa, Corn and etc during the training stage. Once the mapping function is determined, it will be applied to produce the label corresponding to the test input during the testing stage. However, in some practical scenarios, only a few labeled samples are available so learning from scratch is impractical. Meta-learning, which is one of the most popular learning frameworks for few-shot tasks, can learn a priori from massive similar meta-tasks during the training stage and generalize that information to unseen testing tasks during the testing stage.

There are two views of the meta-learning problem: a deterministic view and a probabilistic view [54]. The deterministic view of meta-learning takes a training data set, a test sample, and meta-parameters as input to produce the label corresponding to that test sample straightforward. The probabilistic view takes the training data set and a set of meta-parameters as input, then performs a maximum likelihood inference over the task-specific parameters.

Previous Bayesian non-deep learning meta-learners [35]–[37] aim to extract "general knowledge" or "more intelligent representations tuned to specific sub-domains of a task" from previously learned information and then represent it as a prior probability density in the space of model parameters. Once the support set of the testing task is given, the general knowledge will be updated and a posterior will be computed.

Bayesian principles have the potential to represent the uncertainty of the model, conduct sequential learning, and reduce over-fitting. However, computational concerns of the posterior distribution overshadow the theoretical advantages and make Bayesian models rarely applied in practical application scenarios. Fortunately, variational inference is emerged as a method to approximate the probability density through optimization. The most progress in Bayesian meta-learning has relied on latent variable models optimized with variational inference.

C. Variational Auto-Encoder

Variational auto-encoder (VAE) [55] provides an efficient way to approximate the posterior inference by leveraging the Evidence Lower Bound (ELBO) to estimate the (variational) lower bound on the marginal likelihood of data points.

Consider a dataset $X = \left\{\boldsymbol{x}^{(i)}\right\}_{i=1}^{N}$ containing N independent and identically distributed samples, a recognition model $q(\boldsymbol{z} \mid \boldsymbol{x})$ is an approximation to the intractable true posterior $p(\boldsymbol{z} \mid \boldsymbol{x})$, where the unobserved variables $\boldsymbol{z}$ are interpreted as a latent representation or code. The marginal likelihood is the sum over that of each data point as shown in Eq. (3).

$$\log p(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, ..., \boldsymbol{x}^{(N)}) = \sum_{i=1}^{N} \log p(\boldsymbol{x}^{(i)}) \tag{3}$$

It can be rewritten as

$$\begin{aligned} \log p(\boldsymbol{x}^{(i)}) &= \log q(\boldsymbol{z} \mid \boldsymbol{x}^{(i)}) \cdot \frac{p(\boldsymbol{x}^{(i)}, \boldsymbol{z})}{p(\boldsymbol{z} \mid \boldsymbol{x}^{(i)})q(\boldsymbol{z} \mid \boldsymbol{x}^{(i)})} \\ &= D_{KL}\left(q(\boldsymbol{z} \mid \boldsymbol{x}^{(i)})||p(\boldsymbol{z} \mid \boldsymbol{x}^{(i)})\right) + \mathcal{L}\left(\boldsymbol{x}^{(i)}\right). \end{aligned} \tag{4}$$

Due to the non-negativity of KL-divergence, the second term $\mathcal{L}\left(\boldsymbol{x}^{(i)}\right)$ is called the (variational) lower bound on the marginal likelihood of datapoint i.

$$\begin{aligned} \log p(\boldsymbol{x}^{(i)}) &\geq \mathcal{L}\left(\boldsymbol{x}^{(i)}\right) \\ &= \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x}^{(i)})}\left[\log p(\boldsymbol{x}^{(i)}, \boldsymbol{z}) - \log q(\boldsymbol{z} \mid \boldsymbol{x}^{(i)})\right] \end{aligned} \tag{5}$$

The (variational) lower bound $\mathcal{L}\left(\boldsymbol{x}^{(i)}\right)$ can be rewritten as

$$\begin{aligned} &\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x}^{(i)})}\left[-\log q(\boldsymbol{z} \mid \boldsymbol{x}^{(i)}) + \log p(\boldsymbol{z}) - \log p(\boldsymbol{z}) + \log p(\boldsymbol{x}^{(i)}, \boldsymbol{z})\right] \\ &= -D_{KL}(q(\boldsymbol{z} \mid \boldsymbol{x}^{(i)})||p(\boldsymbol{z})) + \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x}^{(i)})}\left[\log p(\boldsymbol{x}^{(i)} \mid \boldsymbol{z})\right]. \end{aligned} \tag{6}$$

$\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x}^{(i)})}\left[\log p(\boldsymbol{x}^{(i)} \mid \boldsymbol{z})\right]$ is the likelihood after sampling from the inference network q, which can be interpreted as the reconstruction loss of the decoder. $D_{KL}(q(\boldsymbol{z} \mid \boldsymbol{x}^{(i)})||p(\boldsymbol{z}))$ is the KL divergence between the inference network q and the prior network p.

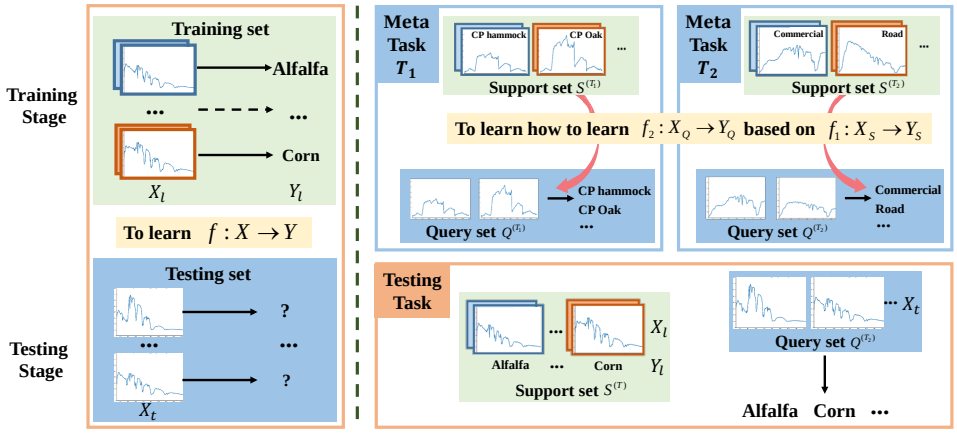


Fig. 3: Comparison of traditional machine learning and meta-learning during the training stage and testing stage for solving a testing task (in orange box). Traditional machine learning aims at learning a mapping from data to label in the training stage and adopt it to obtain the labels of testing samples. In the contrast, meta-learning aims at learning meta-knowledge, which can propose a common solution of massive similar meta-tasks, during the training stage while generalize the meta-knowledge to testing tasks.

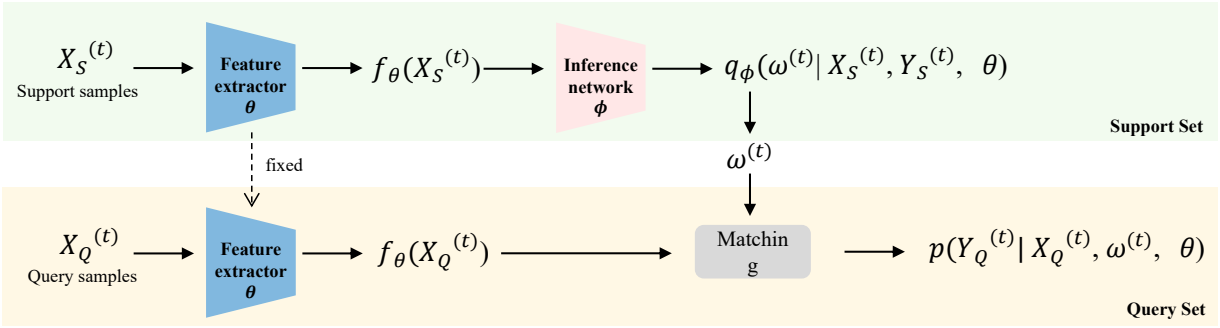


Fig. 4: An overview of our proposed method, BMFSC, for each task (whether meta-task or test task). Our approach aims at providing a probabilistic solution for the hyperspectral image (HSI) few-shot classification (HSI-FSC) task. The prototype vector $\boldsymbol{\omega}^{(t)}$ is treated as a latent variable and its posterior distribution is approximated based on the prior distribution of HSI and support samples given. The prototype vectors drawn from the posterior are used to compute the posterior predictive distribution over the labels of query samples. In meta-tasks, the parameters of the feature-extractor network θ and inference network ϕ are updated by minimizing the predictive performance on query samples. In testing tasks, since θ and ϕ are learned, a feed-forward inference is performed to obtain posterior distribution over the prototype vector and then the posterior predictive distribution.

Note $q(\boldsymbol{z} \mid \boldsymbol{x}^{(i)})$, $p(\boldsymbol{x}^{(i)} \mid \boldsymbol{z})$ and $p(\boldsymbol{z})$ are represented by neural networks. When using variational inference in the context of meta-learning, meta-parameters represent the parameters of $p(\boldsymbol{x}^{(i)} \mid \boldsymbol{z})$ and $p(\boldsymbol{z})$, while task-specific parameters represent the parameters of $q(\boldsymbol{z} \mid \boldsymbol{x}^{(i)})$.

III. Model

Few-shot learning provides a new solution for hyperspectral image classification in the case the data is scarce and requirements for model accuracy are critical. Meta-learning, which is one of the most popular learning frameworks for the few-shot task, can learn prior from massive similar meta-tasks and generalize that information to unseen tasks. Bayesian models relate the prior information to the observations through the likelihood, infer conditioning on data, and compute the posterior.

We assume the prior knowledge acquired from various few-shot HSIC meta-tasks is about how to capture the discriminating features and how to infer the common features of the samples from the same class, and the acquired knowledge is learnable and generalizable. For convenience, we represent the common features of the samples from a specific class as the prototype vector of this class. Therefore, our model has two main components, namely feature extraction network and inference network. The parameters of feature extraction network θ are the knowledge about how to extract discriminating features, while the parameters of inference network ϕ are the knowledge about how to infer an approximate posterior distribution over the prototype vector. The knowledge, θ, and ϕ are obtained via various meta-tasks during the training stage and adopted to new tasks during the testing stage.

To be specific, in a task t, support samples from class i are

fed into the feature extraction network to extract distinguishable features of these samples. Once features of samples from class i are obtained, they are fed into the inference network. The inference network performs inferences and outputs an approximate posterior distribution of the prototype vector of class i. Features of query samples are obtained via the same feature extraction network. The posterior predictive distribution over the labels of query samples is calculated by matching features of query samples and prototype vectors of all classes. Fig. 4 demonstrates the workflow of our proposed new method.

Given a training HSI data set D ($N_D > N$), massive various meta-tasks are generated from D. Each meta-task has the same N and K as the testing task T. Note N and K represent the number of classes and the number of labeled samples in each class respectively. For each meta-task t, we sample N classes from all classes of D to form a subset Γ_t ($N_{\Gamma_t} = N$) , so as to guarantee each classification meta-task is different. We assume knowledge about how to observe and how to infer is learnable and generalizable. Therefore, the parameters of feature extraction network θ and the parameters of inference network ϕ that acquired via various meta-tasks can be shared among testing tasks. However, due to each task with different support samples from different classes, the prototype vectors $\boldsymbol{\omega}^{(t)}$ are task-specific, even for the prototype vector of the same class.

Since the data for all tasks will help ascertain the values of the shared parameters, we will use point estimates for feature extraction network parameters θ and inference network parameters ϕ. In the training stage, our model is trained via gradient descent. The parameters of feature extraction network θ and the parameters of inference network ϕ are updated via various meta-tasks generated from the training HSI data set. In the testing stage, the trained θ and ϕ are shared in different tasks. Meanwhile, we use distributional estimates for prototype vectors $\boldsymbol{\omega}^{(t)}$ since only a few labeled samples can help constrain them. Feature extraction of support samples and query samples is followed by a feed-forward inference procedure for the approximate posterior distribution of prototype vectors. Then a posterior predictive distribution over the labels of query samples is inferred.

A. Probability Expression

We draw on the inference processes that humans perform in few-shot learning tasks. When we are to solve an N-way K-shot hyperspectral classification task t, which aims at classifying query samples in the query set $Q^{(t)} = \{\boldsymbol{x}_q\}$ into N classes with $N \times K$ support samples in the support set $S^{(t)} = \{(\boldsymbol{x}_s, y_s)\}$, we compute the following posterior predictive distribution on the labels of query samples $Y_{Q^{(t)}} = \{y_q\}$ based on support samples $X_{S^{(t)}} = \{\boldsymbol{x}_s\}$, labels of support samples $Y_{S^{(t)}} = \{y_s\}$ and query samples $X_{Q^{(t)}} = \{\boldsymbol{x}_q\}$:

$$p(Y_{Q^{(t)}} \mid X_{S^{(t)}}, Y_{S^{(t)}}, X_{Q^{(t)}}) \tag{7}$$

Since the posterior predictive distribution is hard to be calculated directly, we assume a hierarchical probability structure. We first infer the common features of the samples from the same class, namely prototype vectors of all the N classes. Then the labels of query samples $Y_{Q^{(t)}} = \{y_q\}$ are predicted by the classifier by matching query features and the inference result, prototype vectors. Therefore, our model first extracts discriminating features of support samples $f_\theta(X_{S^{(t)}})$ and features of query samples $f_\theta(X_{Q^{(t)}})$ via a feature extraction network f_θ. Then the model aims at making inferences for the posterior distribution of prototype vectors $\boldsymbol{\omega}^{(t)}$ based on support features $f_\theta(X_{S^{(t)}})$ and labels of support samples $Y_{S^{(t)}} = \{y_s\}$. We use distributional estimates rather than point estimates for prototype vectors $\boldsymbol{\omega}^{(t)}$ because only a few labeled samples can constrain them. By introducing the prototype vector as a hidden variable, the posterior predictive distribution can be factorized as:

$$\begin{aligned} &p(Y_{Q^{(t)}} \mid X_{S^{(t)}}, Y_{S^{(t)}}, X_{Q^{(t)}}, \theta) \\ &= \int p(Y_{Q^{(t)}} \mid X_{Q^{(t)}}, \boldsymbol{\omega}^{(t)}, \theta) \cdot p(\boldsymbol{\omega}^{(t)} \mid X_{S^{(t)}}, Y_{S^{(t)}}, \theta) d\boldsymbol{\omega}^{(t)} \end{aligned} \tag{8}$$

where $\boldsymbol{\omega}^{(t)} = [\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_N]$ and $\boldsymbol{\omega}_i$ is the prototype vector corresponding to the i^{th} class.

Once the posterior distribution of prototype vectors is characterized by the support feature, namely $p(\boldsymbol{\omega}^{(t)} \mid X_{S^{(t)}}, Y_{S^{(t)}}, \theta)$, we consider the predictive distribution for labels of query samples that factor $p(Y_{Q^{(t)}} \mid X_{Q^{(t)}}, \boldsymbol{\omega}^{(t)}, \theta) = \prod_{\boldsymbol{x}_q \in Q^{(t)}} p(y_q \mid \boldsymbol{x}_q, \boldsymbol{\omega}^{(t)}, \theta)$. Without reliable assumptions on query labels $Y_{Q^{(t)}}$, this is the easiest way to guarantee a valid stochastic process.

For an N-way K-shot hyperspectral classification task t, N classes form a subset Γ_t. The probability that a query sample $\boldsymbol{x}_q \in Q^{(t)}$ is classified into class i is given by:

$$p(y_q = i \mid \boldsymbol{x}_q, \boldsymbol{\omega}^{(t)}, \theta) = \frac{e^{f_\theta(\boldsymbol{x}_q) \cdot \boldsymbol{\omega}_i}}{\sum_{i' \in \Gamma} e^{f_\theta(\boldsymbol{x}_q) \cdot \boldsymbol{\omega}_{i'}}} \tag{9}$$

where $f_\theta(\boldsymbol{x}_q)$ is the feature of the query sample $\boldsymbol{x}_q$.

This approach shifts the key of an N-way K-shot hyperspectral classification problem to parameterize the posterior distribution of prototype vectors $p(\boldsymbol{\omega}^{(t)} \mid X_{S^{(t)}}, Y_{S^{(t)}}, \theta)$.

B. Approximation of the Posterior Predictive Distribution

Since the true posterior distribution of prototype vectors is hard to solve directly, our approach takes a variational inference approach to approximate the target distribution p with a simpler distribution q^* by minimizing their KL divergence. The posterior predictive distribution over the labels of query samples $Y_{Q^{(t)}} = \{y_q\}$ based on support samples $X_{S^{(t)}} = \{\boldsymbol{x}_s\}$, labels of support samples $Y_{S^{(t)}} = \{y_s\}$, and query samples $X_{Q^{(t)}} = \{\boldsymbol{x}_q\}$ can be approximated by an amortized distribution as follow:

$$\begin{aligned} &q_\phi(Y_{Q^{(t)}} \mid X_{S^{(t)}}, Y_{S^{(t)}}, X_{Q^{(t)}}, \theta) \\ &= \int p(Y_{Q^{(t)}} \mid X_{Q^{(t)}}, \boldsymbol{\omega}^{(t)}, \theta) \cdot q_\phi(\boldsymbol{\omega}^{(t)} \mid X_{S^{(t)}}, Y_{S^{(t)}}, \theta) d\boldsymbol{\omega}^{(t)} \end{aligned} \tag{10}$$

where θ is the parameters of the feature extraction network, ϕ is the parameters of the inference network.

A problem with this setup is that Evidence Lower Bound contains an expectation w.r.t q, which indicates that we have to be able to back-propagate into q. Since sampling is not

differentiable, we utilize the reparametrization trick, representing $q(\boldsymbol{\omega}^{(t)} \mid X_{S^{(t)}}, Y_{S^{(t)}})$ by the mean ($\boldsymbol{\mu}$) plus the variance ($\boldsymbol{\sigma}$) times the noise (ϵ): $q(\boldsymbol{\omega}^{(t)} \mid X_{S^{(t)}}, Y_{S^{(t)}}) = \boldsymbol{\mu_q} + \boldsymbol{\sigma_q}\epsilon$. Means and variances are determined by the inference network. That is, the output of the inference network delivers mean-field Gaussian parameters to parameterize the approximate posterior distribution of prototype vectors. When drawn from the approximate posterior, prototype vectors are sampled from the implied distribution over the logits rather than directly from the variational distribution. Given that a prototype vector represents the common features of samples from its corresponding class, a factorized Gaussian distribution is adapted for $q_\phi(\boldsymbol{\omega}^{(t)} \mid X_{S^{(t)}}, Y_{S^{(t)}}, \theta)$ in this work. To perform quick predictions on testing tasks, we amortize the approximate posterior distribution directly on the extracted features $f_\theta(\boldsymbol{x})$:

$$q_\phi(\boldsymbol{\omega}^{(t)} \mid X_{S^{(t)}}, Y_{S^{(t)}}, \theta) = \prod_{i=1}^{N} q_\phi(\boldsymbol{\omega}_i \mid f_\theta(X_{S^{(t)}}), Y_{S^{(t)}}) \tag{11}$$

C. Parameters Optimization on Meta-tasks

In order to acquire a model that can solve an N-way K-shot hyperspectral classification task T on the test data set, we train the model on various meta-tasks generated from the training HSI data set D. For each meta-task t, different N classes are sampled from the set of all classes of D to generate a different classification meta-task. The N classes of t are represented as Γ. Once Γ is determined, we sample K samples from each class in Γ as support samples to form t's support set $S^{(t)}$, and sample other L samples from the remaining samples of each class in Γ as query samples to form t's query set $Q^{(t)}$. The optimal parameters of the inference network ϕ^* that can best approximate the posterior predictive distribution, and thus are found by minimizing average expected loss over tasks. The optimal parameters of the feature extraction network θ^* are found by maximizing the predictive performance.

Intuitively, the quality of the approximate posterior predictive distribution can be evaluated by the KL-divergence between the true and approximate posterior predictive distribution:

$$\begin{aligned} &KL[p(\cdot) \parallel q(\cdot)] \\ &p(\cdot) = p(Y_{Q^{(t)}} \mid X_{S^{(t)}}, Y_{S^{(t)}}, X_{Q^{(t)}}, \theta) \\ &q(\cdot) = q_\phi(Y_{Q^{(t)}} \mid X_{S^{(t)}}, Y_{S^{(t)}}, X_{Q^{(t)}}, \theta) \end{aligned} \tag{12}$$

The goal of optimization is to minimize the value of the KL-divergence, so the objective can be given as follows:

$$\begin{aligned} L(\phi) &= - \mathop{\mathbb{E}}_{p(S^{(t)}, Q^{(t)})} [\log q_\phi(Y_{Q^{(t)}} \mid X_{Q^{(t)}}, \theta)] \\ &= - \mathop{\mathbb{E}}_{p(S^{(t)}, Q^{(t)})} [\log \textstyle\int p(Y_{Q^{(t)}} \mid X_{Q^{(t)}}, \boldsymbol{\omega}^{(t)}, \theta) \\ &\qquad \cdot q_\phi(\boldsymbol{\omega}^{(t)} \mid X_{S^{(t)}}, Y_{S^{(t)}}, \theta) d\boldsymbol{\omega}^{(t)}] \end{aligned} \tag{13}$$

It is intractable to compute the log-probability in Equation 10. Therefore, we estimate the probability with M Monte Carlo sampling, and our approach can be effectively optimized in an end-to-end way. The training objective for the shared parameters θ, ϕ becomes:

$$L(\theta, \phi) = \frac{1}{NLT} \sum_{NL,T} \log \frac{1}{M} \sum_{m=1}^{M} p(Y_{Q^{(t)}} \mid X_{Q^{(t)}}, \boldsymbol{\omega}_m^{(t)}, \theta) \tag{14}$$

where $\boldsymbol{\omega}_m^{(t)}$ is sampled from the approximate posterior distribution over prototype vectors, namely $\boldsymbol{\omega}_m^{(t)} \sim q_\phi(\boldsymbol{\omega}^{(t)} \mid X_{S^{(t)}}, Y_{S^{(t)}}, \theta)$, L is the number of query samples per class, $N \times L$ is the total number of query samples and T is the number of the meta-tasks.

The optimization algorithm is shown in Algorithm 1. An unbiased estimate of the optimization objective can be obtained by computing the average of the results and can then be optimized. From this perspective, the information about how to observe (parameters θ) and how to infer (parameters ϕ) can be learned during the procedure of scoring the approximate inference process.

In the testing phase, support samples are the few labeled samples in the testing data set, and query samples are exactly all unlabeled samples to be classified. The trained feature extraction network f_θ and the inference network q_ϕ are fixed. Similar to the training proceeding presented in Algorithm 1, the support samples $X_{S^{(t)}} = \{\boldsymbol{x}_s\}_{s=1}^{s=N\times K}$ and all query samples $X_{Q^{(t)}} = \{\boldsymbol{x}_q\}$ are input into the feature extraction network f_θ to obtain supported features and query features. And then support features are input into the inference network q_ϕ to obtain four parameters of the approximate posterior distribution over prototype vectors $\omega^{(t)}$, namely means and log variances for prototype vectors and biases. For each prototype vector sampled from the obtained approximate distribution, the probability that a query sample $\boldsymbol{x}_q$ belongs to each class can be calculated by $p(y_q = i \mid \boldsymbol{x}_q, \boldsymbol{\omega}^{(t)}, \theta) = \frac{e^{f_\theta(\boldsymbol{x}_q)\cdot\boldsymbol{\omega}_i}}{\sum_{i'\in\Gamma} e^{f_\theta(\boldsymbol{x}_q)\cdot\boldsymbol{\omega}_{i'}}}$. Finally, all the predictions are averaged.

Algorithm 1 Training Proceeding

Input: HSI dataset $D = [(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)]$, $\boldsymbol{x}_i \in \mathbb{R}^{w\times h\times c}$ obtained by merging four training data sets.

Repeat

1. Sample N classes from all classes of D to form a subset Γ_t for a meta-task t.
2. Sample K samples from each class in Γ to form support set $S^{(t)}$ = $\{(\boldsymbol{x}_s, y_s)\}$, and sample other L samples from remaining samples of each class in Γ to form query set $Q^{(t)} = \{\boldsymbol{x}_q\}$.
3. Feed support samples $X_{S^{(t)}} = \{\mathbf{x}_s\}$ and query samples $X_{Q^{(t)}}$ = $\{\mathbf{x}_q\}$ to a feature extraction network f_θ to obtain support features $f_\theta\left(X_{S^{(t)}}\right)$ and query features $f_\theta\left(X_{Q^{(t)}}\right)$.
4. Feed support features $f_\theta\left(X_{S^{(t)}}\right)$ to a inference network q_ϕ, to obtain the parameters of the approximate posterior distribution over prototype vectors $\omega^{(t)}$, namely means and log variances for prototype vectors and biases.
5. Sample prototype vectors $\omega^{(t)}$ from the obtained approximate distribution $q_\phi\left(\omega^{(t)} \middle| X_{S^{(t)}}, Y_{S^{(t)}}, \theta\right)$.
6. Compute the probability that a query sample $\mathbf{x}_q$ is classified into class i conditioned on prototype vectors $\omega^{(t)}$ with Equation 9.
7. Approximate the posterior predictive distribution in Equation 10 with M Monte Carlo sampling. Parameters θ and ϕ are optimized with Equation 14.

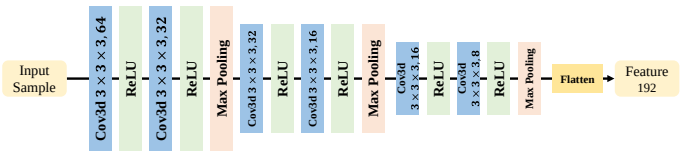


Fig. 5: Architecture of Feature Extractor Network.

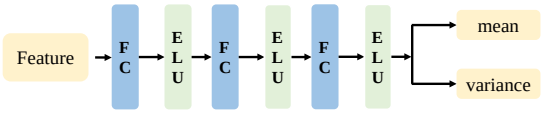


Fig. 6: Architecture of Inference Network.

D. Implementation Details

The architecture of the feature extractor network and the inference network for BMFSC can be chosen arbitrarily. Li et al. [56] indicates that 3-D CNN with small $3\times3\times3$ kernels are effective for hyperspectral image classification. In this paper, we fix the size of the 3-D convolution kernels to $3 \times 3 \times 3$. More detailed information for the two networks is summarized in fig. 6 and 5, where FC represents Fully Connected layer and BN represents Batch Normalization.

Fig. 5 shows the architecture of our feature extractor network f_θ. When the feature extractor network takes a sample as input, the output size of its extracted feature is 192. Fig. 6 presents the architecture of our inference network q_ϕ ("Number of Filters/Kernel Size"). The inference network takes features of K support samples as input, and output mean-field Gaussian parameters to parameterize the approximate posterior distribution of prototype vectors. The function of the Fully Connected layer is to integrate the highly abstracted features obtained after multiple convolutions. We apply the exponential linear unit (ELU) as our active function of inference network primarily for two reasons: 1) The output value of ELU can be negative, thus ELU has a similar effect of Batch Normalization but with lower computational complexity. 2) ELU has the characteristic of soft saturation, which makes it relatively robust to noise in the inactive state. The mathematical expressions of ELU with $0 < \alpha$ is $f(x) = \begin{cases} x, & x > 0 \\ \alpha(e^x - 1), & x \leq 0 \end{cases}$.

IV. Experimental Setup

We evaluate BMFSC on few-shot hyperspectral image classification tasks, where the number of labeled support samples per class (shot) K and the number of classes (way) N change during the test stage. All the experiments are conducted using python on a workstation with an Intel(R) Core(TM) i7-10700K processor (3.80 GHz), 64 GB of memory, and Nvidia GeForce RTX 2080 Ti.

A. Datasets

1) Training Set: To learn the parameters of feature extraction network θ and the parameters of inference network ϕ via various meta-tasks generated from the training HSI data set, we aggregate four publicly available HSI datasets to a large training HSI data set D. This training HSI data set D ranges from field parcel to urban areas and shows different spatial resolutions, which guarantees sample diversity. Detailed information of the four data sets, namely the Kennedy Space Center (KSC) data set, the Chikusei dataset [57], the Houston dataset, and the Botswana dataset is presented in Table I.

The Kennedy Space Center (KSC) data set has 13 classes with more than 100 samples, while the Chikusei data set has 19, the Houston data set has 15, and the Botswana data set has 13. The training HSI data set D has 60 classes in total with 100 samples per class. Note that BS-Nets [58] is adapted to select 100 bands in order to ensure the dimensions fed to the feature extractor f_θ are consistent.

2) Testing Set: To evaluate the classification performance of BMFSC, several experiments are performed on three popular datasets: the Indian Pines (IP) data set, the Salinas Valley (SV) data set and the Pavia University (PU) data set. For the three data sets, 100 pixels are randomly selected from each class for testing. Table II introduces the details of the three datasets. Similarly, 100 bands were selected via BS-Nets [58] to feed into the feature extractor f_θ.

B. Parameter Settings

The number of tasks per batch, the number of episodes, the way (number of classes N for each task t), the shot (number of support samples for each class K), and the learning rate need to be set in advance. In this paper, the number of tasks per batch is set to 4 because of the limited computing power. The number of episodes is set to $15,000$. The way N is set to be the same as the number of classes of the testing tasks. The learning rate is set to 0.001. For the shot numbers, we respectively set $1, 2, 3, 4, 5, 10, 15, 20$. In order to make a fair comparison, we followed the existing work [38] and set the query number L to 19.

For 16-way 5-shot tasks, the window size is set to $5, 7, 9$ and 11 respectively, the corresponding overall accuracy on the testing data sets changes, which is reported in fig. 7. In general, the size of the windows will influence the classification performance. Fig. 7 demonstrates that both a small size of the window (e.g., $5, 7$) and a large size of the window (e.g., 11) greatly decreases the overall classification accuracy. Therefore, the optimum parameter for window size seems to be 9. In this paper, the window size is set to 9. Therefore, when input into the network, each sample is a data cube of $9 \times 9 \times 100$.

V. Experiment

Several parameter setting experiments are conducted to figure out the optimal number of query samples per class L and the optimal window size. The performance of BMFSC-G that adopts the Gaussian distribution to approximate the posterior distribution over prototype vector and BMFSC-D that adopts the Dirichlet distribution are compared. We also conduct a 5-way 5-shot HSIC toy experiment to evaluate the approximate inference performed by the inference network. Experiments on four data sets prove the effectiveness of our proposed approach against competitive baselines in a few-shot setting.

TABLE I: Detailed information of the four data sets that constitute the training set, namely the Kennedy Space Center (KSC) data set, the Chikusei data set [57], the Houston data set, and the Botswana data set.

Dataset	Sensor	No. of band	Wavelength	Spatial resolution	Size	No. of sample	No. of class
KSC	AVIRIS	176	$400-2,500nm$	$18m$	512×614	$5,211$	13
Chikusei	Headwall Hyperspec-VNIR-C	128	$343-1,018nm$	$2.5m$	$2,517\times2,335$	$77,592$	19
Houston	ITRES-CASI 1500	144	$364-1046nm$	$2.5m$	$349\times1,905$	$15,029$	15
Botswana	EO-1	145	$400-2,500nm$	$30m$	$1,476\times256$	$3,248$	14

TABLE II: Detailed information of the four testing data sets, namely the Indian Pines (IP) data set, the Salinas Valley (SV) data set and the Pavia University (PU) data set.

Dataset	Sensor	No. of band	Wavelength	Spatial resolution	Size	No. of sample	No. of class
Indian Pines	AVIRIS	200	$400-2,500nm$	$20m$	145×145	$10,249$	16
Salinas Valley	AVIRIS	204	$400-2,500nm$	$3.7m$	512×217	$54,129$	16
Pavia University	ROSIS	103	$430-860nm$	$1.3m$	610×340	$42,776$	9

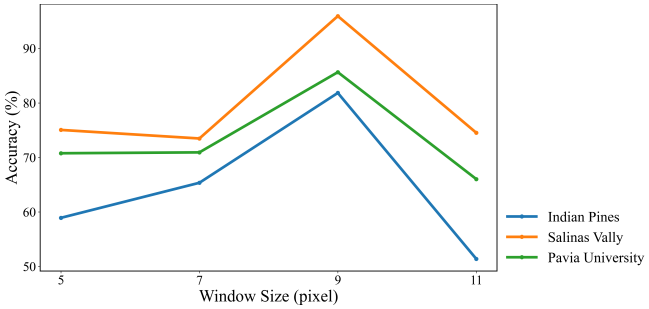


Fig. 7: Assuming the posterior distribution over the prototype vector is a Gaussian distribution, classification accuracy varies as the window size increases.

A. *Classification performance*

In this section, we compare the classification performance of BMFSC with DFSL [38]), DCFSL [41], CMFSL [59] and CFSL [31]. All the comparison methods are specifically designed to solve the few-shot hyperspectral image classification problem. The numerical results for the classification models above are compared with BMFSC in detail with the shot (numbers of support samples per class K) ranges in $1,2,3,4,5,10,15,20$. Classification maps for few-shot hyperspectral image classification models are compared with BMFSC when given 5 support samples.

Overall accuracy is reported in III. Classification maps for different classifiers on the four testing data sets with 5 support samples are presented in figs. 8 to 10. The time taken by BMFSC in model training and testing is presented in Table IV. Table V to VII show the individual class-wise accuracies of the different methods on the three testing data sets, namely the Indian Pines data set, the Salinas Valley data set and the Pavia University data set. From Table III, the results of the experiments prove that the overall accuracy of every classification method generally increases with the increase of support samples per class K. For the performance on the three datasets, we can find BMFSC performs better than others generally, which indicates the superiority of our design. It can be observed that the proposed BMFSC gets a significant performance boost compared with deterministic FSL methods (DFSL+NN [38], DFSL+SVM [38]), DCFSL [41], CMFSL [59], CFSL [31]) given a few labeled samples. A possible reason is BMFSC can propose multiple potential solutions when there might simply not be enough information in a small dataset for a new task. Especially for a small number of support samples per class,$1-5$, BMFSC is proven to be effective.

The overall accuracy and standard deviation (percentage) of the different methods on the Indian Pines Data Set are presented in Table III. Compared with DFSL+NN and DFSL+SVM methods (using point estimates for prototype vectors), BMFSC (using distributional estimates) increases OA by 17.21% given only 3 labeled samples, 14% given 5 labeled samples. It indicates that using distributional estimates for prototype vectors is effective. It can be observed from fig. 8 that BMFSC can perform better classification on samples from corn-mintill, grass tree, and soybean-mintill. Similar conclusions can be drawn with the results on the Salinas Valley data set and the Pavia University data set reported in III and figs. 9 and 10.

Experimental results show that there are performance differences between BMFSC and other methods. One possible reason for the difference in performance is that BMFSC adopts distribution estimates, while other comparison methods adopt point estimates. For the performance difference on different classes, a possible reason is that for these classes, samples from the same class do not strictly follow the same Gaussian distribution. Therefore, the Gaussian distribution assumption used in our method cannot bring a noticeable improvement on these classes.

B. *Ablation Experiments*

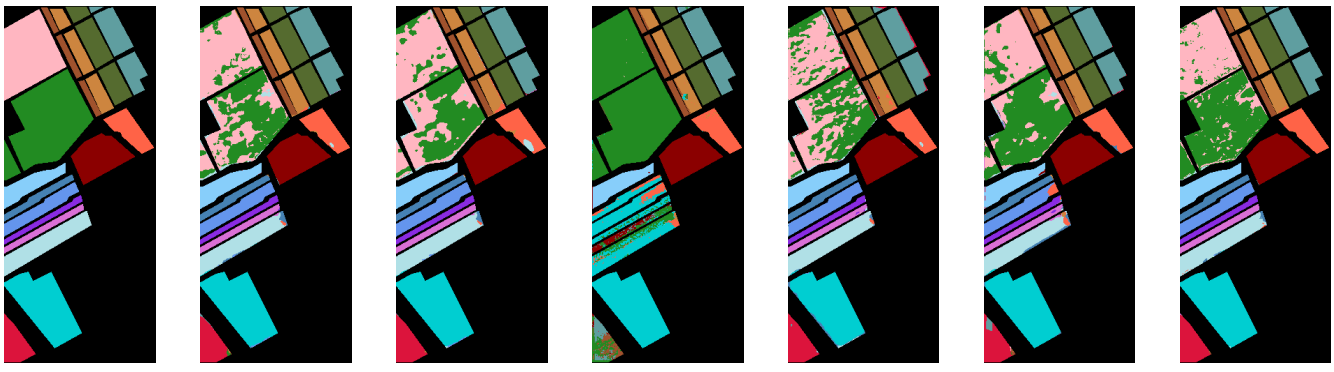
To prove the effectiveness of the novelty, we have conducted three ablation experiments on the architecture of the feature extraction network (FE1, FE2), the estimates fashion (point estimates, PE and distribution estimates, DE), the approximate distribution type (Gaussian, Dirichlet). The results are shown in Table VIII.

TABLE III: Overall accuracy and standard deviation (percentage) of the Different Methods on the Four Test Data Sets, namely Indian Pines Data Set,Salinas Valley Data Set, and Pavia University Data Set. K is the number of labeled samples as support from each class. Best results are highlighted in bold.

Dataset	Methods	K = 1	K = 2	K = 3	K = 4	K = 5	K = 10	K = 15	K = 20
IP	DFSL+NN	48.19 ± 5.60	58.60 ± 5.53	64.05 ± 4.10	61.73 ± 3.44	67.84 ± 1.29	76.49 ± 1.44	78.62 ± 1.59	81.74 ± 0.95
	DFSL+SVM	49.36 ± 7.82	58.46 ± 4.61	60.59 ± 5.09	64.20 ± 5.69	64.58 ± 2.78	75.53 ± 1.89	79.98 ± 2.23	83.01 ± 1.67
	DCFSL	41.87 ± 5.90	50.97 ± 5.52	55.67 ± 3.84	57.55 ± 4.50	60.35 ± 3.99	71.00 ± 1.84	77.72 ± 2.29	79.53 ± 1.29
	CMFSL	43.42 ± 3.57	52.61 ± 4.23	58.32 ± 3.67	62.33 ± 2.48	64.47 ± 2.14	73.40 ± 2.08	78.71 ± 1.55	82.34 ± 1.66
	CFSL	41.85 ± 5.57	52.48 ± 4.84	59.37 ± 3.11	63.70 ± 3.16	66.64 ± 2.40	76.21 ± 2.55	81.51 ± 0.95	82.60 ± 5.24
	BMFSC	$\mathbf{56.67 \pm 3.27}$	$\mathbf{67.35 \pm 2.08}$	$\mathbf{81.26 \pm 0.98}$	$\mathbf{81.55 \pm 0.75}$	$\mathbf{81.84 \pm 0.88}$	$\mathbf{83.42 \pm 0.73}$	$\mathbf{84.01 \pm 0.67}$	$\mathbf{84.89 \pm 0.68}$
SV	DFSL+NN	$\mathbf{81.19 \pm 3.76}$	$\mathbf{82.73 \pm 3.55}$	86.66 ± 2.26	87.21 ± 1.25	88.40 ± 1.54	89.86 ± 1.69	92.15 ± 1.24	92.69 ± 0.98
	DFSL+SVM	77.28 ± 4.83	80.82 ± 3.86	84.86 ± 2.14	85.43 ± 4.04	85.58 ± 1.87	89.73 ± 1.24	91.21 ± 1.64	93.42 ± 1.25
	DCFSL	74.29 ± 2.34	81.83 ± 2.04	83.14 ± 3.15	87.72 ± 1.76	88.34 ± 0.94	90.82 ± 3.01	92.85 ± 0.92	93.95 ± 0.56
	CMFSL	76.40 ± 4.64	80.67 ± 1.94	84.64 ± 2.65	86.60 ± 1.75	87.51 ± 1.67	90.89 ± 1.62	92.22 ± 1.04	92.81 ± 0.78
	CFSL	75.69 ± 3.18	81.34 ± 2.68	85.30 ± 3.30	88.13 ± 1.63	89.24 ± 2.01	92.36 ± 1.04	93.52 ± 0.94	94.14 ± 0.71
	BMFSC	75.46 ± 1.95	80.95 ± 1.01	$\mathbf{94.83 \pm 0.55}$	$\mathbf{95.24 \pm 1.97}$	$\mathbf{95.94 \pm 0.38}$	$\mathbf{96.07 \pm 0.34}$	$\mathbf{96.13 \pm 0.33}$	$\mathbf{96.45 \pm 0.29}$
PU	DFSL+NN	49.99 ± 8.25	62.12 ± 4.52	68.80 ± 5.61	73.25 ± 4.91	80.81 ± 3.12	84.79 ± 2.27	86.68 ± 2.61	89.59 ± 1.05
	DFSL+SVM	53.88 ± 6.32	58.88 ± 8.12	64.06 ± 6.12	66.74 ± 7.37	72.57 ± 3.93	84.56 ± 1.83	87.23 ± 1.38	90.69 ± 1.29
	DCFSL	59.62 ± 8.23	67.16 ± 7.68	72.21 ± 4.02	78.59 ± 1.37	78.73 ± 1.20	84.81 ± 3.51	87.02 ± 3.26	90.53 ± 1.25
	CMFSL	58.37 ± 9.89	68.94 ± 7.05	73.61 ± 6.25	80.29 ± 3.00	80.58 ± 3.10	86.46 ± 4.13	89.16 ± 3.80	91.54 ± 2.23
	CFSL	60.51 ± 6.57	73.29 ± 4.71	77.45 ± 5.34	83.66 ± 2.17	84.26 ± 4.11	$\mathbf{89.09 \pm 4.03}$	$\mathbf{91.44 \pm 3.56}$	$\mathbf{94.42 \pm 0.91}$
	BMFSC	$\mathbf{61.48 \pm 2.54}$	$\mathbf{75.77 \pm 1.39}$	$\mathbf{82.77 \pm 0.90}$	$\mathbf{83.73 \pm 1.67}$	$\mathbf{85.65 \pm 0.71}$	87.08 ± 0.56	87.88 ± 0.51	88.00 ± 0.47

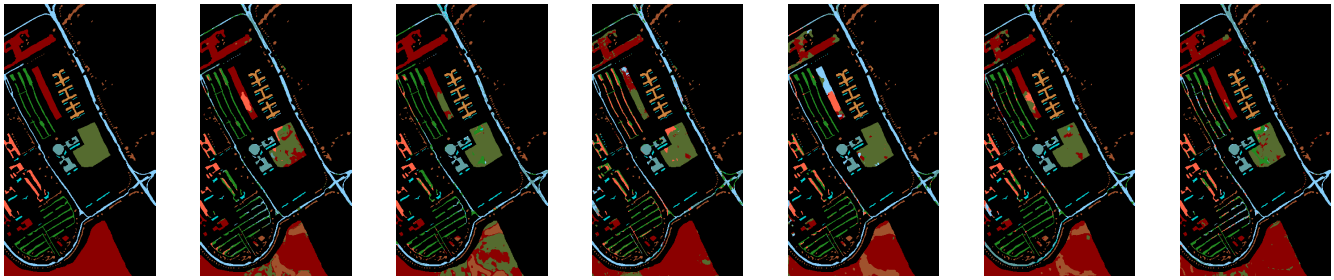
(a) GT (b) DFSL+NN (c) DFSL+SVM (d) DCFSL (e) CMFSL (f) CFSL (g) BMFSC

Fig. 8: Classification maps for the Indian Pines data set with 5 support samples. (a) Ground-truth. (b)–(e) Classification maps for different classifiers. (b) DFSL+NN. (c) DFSL+SVM. (d) DCFSL. (e) CMFSL. (f) CFSL. (g) BMFSC.



(a) GT (b) DFSL+NN (c) DFSL+SVM (d) DCFSL (e) CMFSL (f) CFSL (g) BMFSC

Fig. 9: Classification maps for the Salinas Valley data set with 5 support samples. (a) Ground-truth. (b)–(e) Classification maps for different classifiers. (b) DFSL+NN. (c) DFSL+SVM. (d) DCFSL. (e) CMFSL. (f) CFSL. (g) BMFSC.



(a) GT (b) DFSL+NN (c) DFSL+SVM (d) DCFSL (e) CMFSL (f) CFSL (g) BMFSC

Fig. 10: Classification maps for the Pavia University data set with 5 support samples. (a) Ground-truth. (b)–(e) Classification maps for different classifiers. (b) DFSL+NN. (c) DFSL+SVM. (d) DCFSL. (e) CMFSL. (f) CFSL. (g) BMFSC.

TABLE IV: Time Taken by BMFSC in Model Training and Testing

Data Set	Train + Validation Time	Test Time
PU	23883s (6.63h)	5.25s
SV	63850s (17.73h)	7.79s
IP	66266s (18.41h)	1.99s

TABLE V: Class-wise accuracies on Indian Pines Data Set.

Class	K	L	Accuracy					
			DFSL+NN	DFSL+SVM	DCFSL	CMFSL	CFSL	BMFSC
1	5	41	91.3	97.83	95.37	99.27	95.85	**100**
2	5	1423	40.55	37.18	43.26	43.77	48.65	**62.04**
3	5	825	51.45	79.76	57.95	56.05	58.33	**90.6**
4	5	232	60.34	77.22	80.6	79.35	79.35	**96.2**
5	5	478	91.93	85.3	72.91	74.9	74.85	**98.14**
6	5	725	90	**90.96**	87.96	77.17	87.13	90.82
7	5	23	**100**	**100**	99.57	99.13	98.7	**100**
8	5	473	90.17	94.14	86.26	86.79	89.01	**98.54**
9	5	15	**100**	**100**	99.33	99.33	98.67	**100**
10	5	967	58.44	64.4	62.44	62.35	61.3	**68.21**
11	5	2450	63.46	40.37	62.75	56.11	59.07	**73.4**
12	5	588	55.99	78.92	48.72	46.99	46.17	**88.2**
13	5	200	94.63	92.68	**99.35**	99	98.45	99.02
14	5	1260	86.25	80.55	85.4	83.47	83.01	**90.43**
15	5	381	75.91	94.82	66.69	75.83	76.69	**96.37**
16	5	88	100	100	97.61	99.77	97.27	**100**
OA			67.32	65.84	66.81	64.47	66.64	**81.65**
Kappa			63.19	62.11	62.64	59.99	62.46	**79.28**

K is the number of support samples. L is the number of query samples.

TABLE VI: Class-wise accuracies on Salinas Valley Data Set.

Class	K	L	Accuracy					
			DFSL+NN	DFSL+SVM	DCFSL	CMFSL	CFSL	BMFSC
1	**5**	2004	99.7	99.1	99.4	97.52	99.58	**100**
2	5	3721	99.41	94.79	99.76	98.44	99.35	**99.87**
3	5	1971	99.04	91.8	91.96	92.64	92.37	**99.39**
4	5	1389	99.28	97.7	**99.55**	98.97	98.95	99.07
5	5	2673	93.69	95.26	92.7	93.04	93.37	**96.86**
6	5	3954	99.72	97.75	99.52	99.54	99.15	**100**
7	5	3574	99.25	**99.69**	98.88	96.86	98.06	99.53
8	5	11266	68.13	45.94	74.57	69.17	74.26	**87.56**
9	5	6198	98.86	99.05	**99.59**	98.97	99.24	99.39
10	5	3273	94.87	95.39	86.42	84.85	84.58	**98.05**
11	5	1063	**99.44**	98.88	96.61	98.21	97.79	97.94
12	5	1922	99.9	99.9	99.93	99.41	98.82	**100**
13	5	911	100	100	99.3	99.68	98.74	**100**
14	5	1065	99.25	**99.35**	98.85	98.88	97.63	98.97
15	5	7263	68.51	**88.95**	75.38	74.36	77.63	88.47
16	5	1802	98.17	**100**	92.22	87.25	90.09	99
OA			88.12	85.68	89.34	87.51	89.24	**95.34**
Kappa			86.81	84.19	88.17	86.14	88.06	**94.81**

K is the number of support samples. L is the number of query samples.

TABLE VII: Class-wise accuracies on Pavia University Data Set.

Class	K	L	Accuracy					
			DFSL+NN	DFSL+SVM	DCFSL	CMFSL	CFSL	BMFSC
1	5	6626	74.27	47.04	**82.2**	71.83	78.3	78.51
2	5	18644	79.94	74.51	**87.74**	84.14	86.98	84.99
3	5	2094	76.56	74.7	67.46	61.97	72.25	**82.85**
4	5	3059	87.76	**94.68**	93.16	93.97	94.48	88.45
5	5	1340	**99.93**	99.26	99.49	99.6	99.52	99.7
6	5	5024	85.58	78.35	77.32	76.75	76.44	**96.4**
7	5	1325	90.3	**95.86**	81.18	78.94	85.67	93.61
8	5	3677	**87.4**	65.21	66.73	71.82	80.3	75.29
9	5	942	85.43	97.68	98.66	**99.32**	99.2	96.09
OA			80.52	73.31	83.65	80.58	84.26	**85.61**
Kappa			75.28	66.37	78.7	74.99	79.61	**81.44**

K is the number of support samples. L is the number of query samples.

TABLE VIII: The ablation experiments. We finally selected the last setting.

Settings						IP	SV	PU
FE1[a]	FE2[b]	PE[c]	DE[d]	Gaussian[e]	Dirichlet[f]			
✓		✓		✓		59.22 ± 1.27	82.26 ± 0.24	80.13 ± 0.34
	✓	✓		✓		64.52 ± 3.13	83.96 ± 1.64	81.67 ± 1.65
✓			✓	✓		73.60 ± 2.11	88.04 ± 1.51	82.76 ± 2.66
	✓		✓		✓	80.85 ± 0.24	88.87 ± 0.19	81.08 ± 0.25
	✓		✓	✓		$\mathbf{81.84 \pm 0.88}$	$\mathbf{95.94 \pm 0.38}$	$\mathbf{85.65 \pm 0.71}$

[a]FE1 is the feature extraction network in DFSL [38].
[b]FE2 is the feature extraction network in fig. 5.
[c]PE is point estimates.
[d]DE is distribution estimates.
[e]Gaussian to approximate the posterior distribution over prototype vector.
[f]Dirichlet to approximate the posterior distribution over prototype vector.

Both the proposed architecture of the feature extraction network and distribution estimates contribute to the superiority of our method. For the posterior distribution over the prototype vector, since Gaussian distribution and Dirichlet distribution are commonly used in the field of hyperspectral image processing, both Gaussian and Dirichlet are considered. We conduct an experiment to evaluate which distribution can better approximate the posterior distribution over the prototype vector. It can be drawn from Table VIII that the Gaussian distribution is more suitable to approximate the posterior distribution of the prototype vector. The probable reason why Gaussian distribution is a better choice for our method might be two-fold. On one hand, hyperspectral images are generally considered to follow a mixture of Gaussian distributions. On the other hand, Dirichlet distribution is more suitable for ap-

plication scenarios where the number of Gaussian distributions producing the data is unknown and hence the number of clustering cannot be determined.

C. The approximate inference

BMFSC formalizes the few-shot hyperspectral image classification problem in a probabilistic way and provides a careful process of meta-learning probabilistic inference. The key of the proposed method, BMFSC, is the feed-forward inference procedure for approximate posterior distribution over prototype vectors $q_\phi(\boldsymbol{\omega}^{(t)} \mid X_{S^{(t)}}, Y_{S^{(t)}}, \theta)$. Because only when the inference process is able to get appropriate posterior distributions over prototype vectors can we draw appropriate prototype vectors from it, which is the premise and basement of more reliable classification performance.

To evaluate the approximate inference performed by the inference network, we conduct a 5-way 5-shot HSIC toy experiment. We train a model via gradient descent on various 5-way 5-shot meta-tasks generated from the training HSI data set and evaluate its performance on the testing 5-way 5-shot task T. For each meta-task t, we sampled different 5 classes Γ from all classes of the training data set in order to generate a different 5-way 5-shot meta-task. For each class in Γ, we draw 5 support samples and 10 query samples to form the support set $S^{(t)}$ and the query set Q^t. Totally, $1,000$ various meta-tasks are generated. Once the optimal parameters of the feature extractor network θ^* and the optimal parameters of the inference network ϕ^* are derived via meta-tasks, we apply them to solve the testing task T. The approximate posterior distribution over the prototype vector is inferred by the learned parameters ϕ. Then the posterior predictive distribution over the labels of query samples $Y_{Q^{(t)}} = \{y_q\}$ based on support samples $X_{S^{(t)}} = \{\boldsymbol{x}_s\}$, labels of support samples $Y_{S^{(t)}} = \{y_s\}$, and query samples $X_{Q^{(t)}} = \{\boldsymbol{x}_q\}$ can be obtained.

We draw a prototype vector of class 1 $\boldsymbol{\omega}_1$ from the approximate posterior distribution over the prototype vector of class 1 delivered by the inference network. Similarly, prototype vectors of 5 class $\boldsymbol{\omega}_i, i = 1, ..., 5$ are sampled from their corresponding approximate posterior distribution. The feature of a query sample from the first class can be considered as a sample from the true posterior distribution for the prototype vector of the first class. Therefore, the similarity between the extracted feature of the query sample from the first class (Query from Class 1 in fig. 11) and the prototype vector of its corresponding class (PV of Class 1 in fig. 11) can to some extent reflect the similarity between the true posterior distribution and the approximate posterior distribution over prototype vector. In fig. 11, we show the inferred five prototype vectors of the 5 new classes of the testing task T and the extracted feature of a query sample from the first class of the testing task T. Note the output size of the extracted feature and inferred prototype vector is 192. To offer direct and easy observing, we add bias $5, 10, 15, 20, 25$ to each prototype vector respectively.

It can be observed that the extracted feature of the query sample (Query from Class 1) is almost consistent with the prototype vector of its corresponding class (PV of Class 1) and they show roughly the same change tendency. To account for this, cosine similarity is adapted to measure the similarity between the query feature and each prototype vector. The cosine similarity between the query feature and the prototype vector of the first class is far closer to 1. Therefore, the inference procedure can obtain appropriate posterior distributions over prototype vectors $\boldsymbol{\omega}^{(t)}$, though we directly focus on the posterior predictive distribution and minimize the predictive KL divergence $KL[p(Y_{Q^{(t)}} \mid X_{Q^{(t)}}, X_{S^{(t)}}, Y_{S^{(t)}}, \theta) \parallel q_\phi(Y_{Q^{(t)}} \mid X_{Q^{(t)}}, X_{S^{(t)}}, Y_{S^{(t)}}, \theta)]$.

VI. Conclusion

We propose a Bayesian Meta-learning solution for few-shot classification in hyperspectral images called BMSFC. We build a hierarchical probability model from the Bayesian view and provide a process of meta-learning probabilistic inference. The prototype vectors are introduced as latent variables and their posterior is obtained to acquire the posterior predictive distribution over the labels of test samples. During the test stage, rather than using gradient-based optimization, we perform a feed-forward inference procedure for amortizing posterior distribution over the prototype vectors. All the processes are under a meta-learning framework and meta-knowledge from the similar few-shot classification tasks are learned on other extraneous datasets of known labels. Experiments on four common datasets show the superiority of our method especially when only a few samples are given. BMFSC may provide new thinking of investigation in hyperspectral image few-shot classification, and even other applications, such as the fast object detection and instance segmentation of remote sensing images.

Acknowledgment

The codes of the band selection method, BS-Nets, were provided by the https://github.com/AngryCai/BS-Nets.

References

[1] Y. Ding, S. Pan, and Y. Chong, "Robust spatial–spectral block-diagonal structure representation with fuzzy class probability for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 3, pp. 1747–1762, 2020.

[2] H. Zhai, H. Zhang, L. Zhang, and P. Li, "Total variation regularized collaborative representation clustering with a locally adaptive dictionary for hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 1, pp. 166–180, 2019.

[3] S. Zhou, L. Sun, and Y. Ji, "Germination prediction of sugar beet seeds based on hsi and svm-rbf," in *2019 4th International Conference on Measurement, Information and Control (ICMIC)*, pp. 93–97, 2019.

[4] M. A. Haq, G. Rahaman, P. Baral, and A. Ghosh, "Deep learning based supervised image classification using uav images for forest areas classification," *Journal of the Indian Society of Remote Sensing*, vol. 49, pp. 601–606, Mar 2021.

[5] B. Pan, Z. Shi, Z. An, Z. Jiang, and Y. Ma, "A novel spectral-unmixing-based green algae area estimation method for goci data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, pp. 1–13, Jul 2016.

[6] P. Chavez, Jr and A. Kwarteng, "Extracting spectral contrast in landsat thematic mapper image data using selective principal component analysis," *Photogrammetric Engineering and Remote Sensing*, vol. 55, pp. 339–348, Jan 1989.

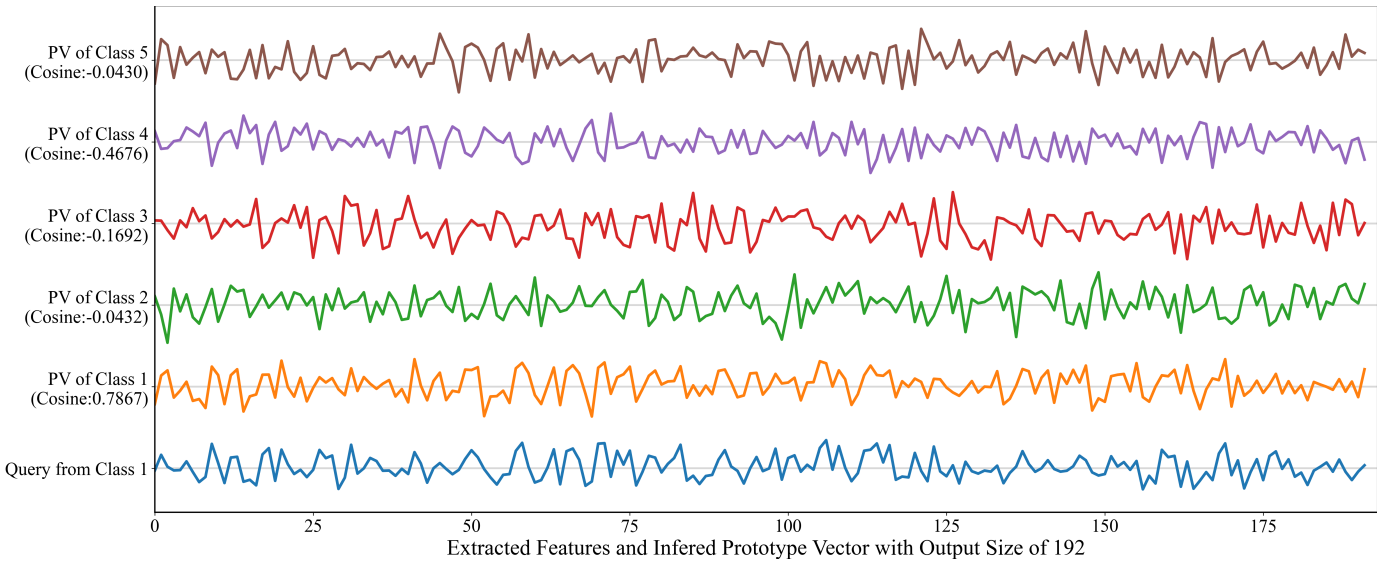


Fig. 11: We conduct a 5-way 5-shot HSIC toy experiment to evaluate the approximate inference performed by the inference network. The feature of a query sample from the first class $f_\theta(x_q^1)$ is supposed to be sampled from the true posterior distribution for the prototype vector of the first class. Prototype vector of each class $\omega_i, i = 1, ..., 5$ is drawn from its corresponding approximate posterior distribution. Note the output size of the extracted feature and inferred prototype vector is 192. To offer direct and easy observing, we add bias $5, 10, 15, 20, 25$ to each prototype vector respectively. It can be observed that the extracted feature of the query sample (Query from Class 1) is almost consistent with the prototype vector of its corresponding class (PV of Class 1) and they show roughly the same change tendency. To account for this, cosine similarity is adapted to measure the similarity between the query feature and each prototype vector. The cosine similarity between the query feature and the prototype vector of the first class is far closer to 1. Therefore, the inference procedure can obtain appropriate posterior distributions over prototype vectors $\boldsymbol{\omega}^{(t)}$.

[7] Y. Li, Q. Li, Y. Liu, and W. Xie, "A spatial-spectral sift for hyperspectral image matching and classification," *Pattern Recognition Letters*, vol. 127, pp. 18–26, 2019. Advances in Visual Correspondence: Models, Algorithms and Applications (AVC-MAA).

[8] Q. Wang, X. He, and X. Li, "Locality and structure regularized low rank representation for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 911–923, 2019.

[9] S. Prasad and L. M. Bruce, "Limitations of principal components analysis for hyperspectral target recognition," *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 4, pp. 625–629, 2008.

[10] L.-Z. Wang, H. Huang, and H.-L. Feng, "Hyperspectral remote sensing image classification based on ssmfa and knns," *Tien Tzu Hsueh Pao/Acta Electronica Sinica*, vol. 40, pp. 780–787, Apr 2012.

[11] B. Pan, Z. Shi, N. Zhang, and S. Xie, "Hyperspectral image classification based on nonlinear spectral–spatial network," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 12, pp. 1782–1786, 2016.

[12] B. Du, L. Zhang, T. Chen, and K. Wu, "A discriminative manifold learning based dimension reduction method for hyperspectral classification," *International Journal of Fuzzy Systems*, vol. 14, pp. 272–277, Jun 2012.

[13] L. He, C. Liu, J. Li, Y. Li, S. Li, and Z. Yu, "Hyperspectral image spectral–spatial-range gabor filtering," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 7, pp. 4818–4836, 2020.

[14] X. Kang, C. Li, S. Li, and H. Lin, "Classification of hyperspectral images by gabor filtering based deep network," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 4, pp. 1166–1178, 2018.

[15] B. Pan, Z. Shi, and X. Xu, "Hierarchical guidance filtering-based ensemble classification for hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 4177–4189, 2017.

[16] B. Pan, Z. Shi, and X. Xu, "R-vcanet: A new deep-learning-based hyperspectral image classification method," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 5, pp. 1975–1986, 2017.

[17] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, vol. 7, pp. 2094–2107, Jun 2014.

[18] B. Pan, X. Xu, Z. Shi, N. Zhang, H. Luo, and X. Lan, "Dssnet: A simple dilated semantic segmentation network for hyperspectral imagery classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 11, pp. 1968–1972, 2020.

[19] L. Liu, Z. Shi, B. Pan, N. Zhang, H. Luo, and X. Lan, "Multiscale deep spatial feature extraction using virtual rgb image for hyperspectral imagery classification," *Remote Sensing*, vol. 12, p. 280, Jan 2020.

[20] Y. Liu, Q. Li, Y. Yuan, Q. Du, and Q. Wang, "Abnet: Adaptive balanced network for multiscale object detection in remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.

[21] Q. Wang, X. He, and X. Li, "Locality and structure regularized low rank representation for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 911–923, 2019.

[22] Y. Zhan, D. Hu, Y. Wang, and X. Yu, "Semisupervised hyperspectral image classification based on generative adversarial networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 2, pp. 212–216, 2018.

[23] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, (Cambridge, MA, USA), p. 2672–2680, MIT Press, 2014.

[24] I. Dopido, J. Li, P. R. Marpu, A. Plaza, J. M. Bioucas Dias, and J. A. Benediktsson, "Semisupervised self-learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 7, pp. 4032–4044, 2013.

[25] H. Wu and S. Prasad, "Dirichlet process based active learning and discovery of unknown classes for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 8, pp. 4882–4895, 2016.

[26] X. Cao, J. Yao, Z. Xu, and D. Meng, "Hyperspectral image classification with convolutional neural network and active learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 7, pp. 4604–4616, 2020.

[27] F. Nie, S. Shi, and X. Li, "Semi-supervised learning with auto-weighting

feature and adaptive graph," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 6, pp. 1167–1178, 2020.
[28] Y. Miao, M. Chen, Y. Yuan, J. Chanussot, and Q. Wang, "Hyperspectral imagery classification via random multigraphs ensemble learning," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 641–653, 2022.
[29] P. Sellars, A. I. Aviles-Rivero, and C.-B. Schönlieb, "Superpixel contracted graph-based learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 6, pp. 4180–4193, 2020.
[30] N. Li, D. Zhou, J. Shi, X. Zheng, T. Wu, and Z. Yang, "Graph-Based Deep Multitask Few-Shot Learning for Hyperspectral Image Classification," *Remote Sensing*, vol. 14, p. 2246, May 2022.
[31] Y. Zhang, W. Li, M. Zhang, S. Wang, R. Tao, and Q. Du, "Graph information aggregation cross-domain few-shot learning for hyperspectral image classification," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2022.
[32] J. Yang, Y. Zhao, J. C.-W. Chan, and C. Yi, "Hyperspectral image classification using two-channel deep convolutional neural network," in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 5079–5082, 2016.
[33] H. Zhang, Y. Li, Y. Jiang, P. Wang, Q. Shen, and C. Shen, "Hyperspectral classification based on lightweight 3-d-cnn with transfer learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 8, pp. 5813–5828, 2019.
[34] Y. Jiang, Y. Li, and H. Zhang, "Hyperspectral image classification based on 3-d separable resnet and transfer learning," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 12, pp. 1949–1953, 2019.
[35] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 594–611, 2006.
[36] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum, "One shot learning of simple visual concepts," in *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, 2011.
[37] R. Salakhutdinov, J. Tenenbaum, and A. Torralba, "One-shot learning with a hierarchical nonparametric bayesian model," in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning* (I. Guyon, G. Dror, V. Lemaire, G. Taylor, and D. Silver, eds.), vol. 27 of *Proceedings of Machine Learning Research*, (Bellevue, Washington, USA), pp. 195–206, PMLR, 02 Jul 2012.
[38] B. Liu, X. Yu, A. Yu, P. Zhang, G. Wan, and R. Wang, "Deep few-shot learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 4, pp. 2290–2304, 2019.
[39] X. Ma, S. Ji, J. Wang, J. Geng, and H. Wang, "Hyperspectral image classification based on two-phase relation learning network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 12, pp. 10398–10409, 2019.
[40] B. Deng and D. Shi, "Relation network for hyperspectral image classification," in *2019 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pp. 483–488, 2019.
[41] Z. Li, M. Liu, Y. Chen, Y. Xu, W. Li, and Q. Du, "Deep cross-domain few-shot learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–18, 2021.
[42] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), p. 4080–4090, Curran Associates Inc., 2017.
[43] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1199–1208, 2018.
[44] M. Rao, P. Tang, and Z. Zhang, "Spatial–spectral relation network for hyperspectral image classification with limited training samples," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 12, pp. 5086–5100, 2019.
[45] F. Zhou, L. Zhang, W. Wei, Z. Bai, and Y. Zhang, "Meta transfer learning for few-shot hyperspectral image classification," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pp. 3681–3684, 2021.
[46] J. Bai, S. Huang, Z. Xiao, X. Li, Y. Zhu, A. C. Regan, and L. Jiao, "Few-shot hyperspectral image classification based on adaptive subspaces and feature transformation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–17, 2022.
[47] M. Steyvers, T. L. Griffiths, and S. Dennis, "Probabilistic inference in human semantic memory," *Trends in Cognitive Sciences*, vol. 10, no. 7, pp. 327–334, 2006. Special issue: Probabilistic models of cognition.
[48] J. Tenenbaum, C. Kemp, T. Griffiths, and N. Goodman, "How to grow a mind: Statistics, structure, and abstraction," *Science (New York, N.Y.)*, vol. 331, pp. 1279–1285, Mar 2011.
[49] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, pp. 859 – 877, 2016.
[50] S. Müller, N. Hollmann, S. P. Arango, J. Grabocka, and F. Hutter, "Transformers can do bayesian inference," 2022.
[51] J. Yoon, T. Kim, O. Dia, S. Kim, Y. Bengio, and S. Ahn, "Bayesian model-agnostic meta-learning," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, (Red Hook, NY, USA), p. 7343–7353, Curran Associates Inc., 2018.
[52] W. K. Hastings, "Monte Carlo Sampling Methods using Markov Chains and their Applications," *Biometrika*, vol. 57, pp. 97–109, Apr. 1970.
[53] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine Learning*, vol. 37, pp. 183–233, Nov 1999.
[54] C. Finn, *Learning to Learn with Gradients*. PhD thesis, EECS Department, University of California, Berkeley, Aug 2018.
[55] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2014.
[56] Y. Li, H. Zhang, and Q. Shen, "Spectral-Spatial Classification of Hyperspectral Imagery with 3D Convolutional Neural Network," *Remote Sensing*, vol. 9, p. 67, Jan. 2017.
[57] N. Yokoya and A. Iwasaki, "Airborne hyperspectral data over chikusei," Tech. Rep. SAL-2016-05-27, Space Application Laboratory, University of Tokyo, Japan, May 2016.
[58] Y. Cai, X. Liu, and Z. Cai, "Bs-nets: An end-to-end framework for band selection of hyperspectral image," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 3, pp. 1969–1984, 2020.
[59] B. Xi, J. Li, Y. Li, R. Song, D. Hong, and J. Chanussot, "Few-shot learning with class-covariance metric for hyperspectral image classification," *IEEE Transactions on Image Processing*, vol. 31, pp. 5079–5092, 2022.

Jing Zhang received her B.S. degree from Computer Science and Technology from Xidian University, Xi'an, China in 2020. She is currently working toward her master's degree in the Image Processing Center, School of Astronautics, Beihang University. Her research interests include deep learning image processing, and pattern recognition.

Liqin Liu received her B.S. degree from Beihang University, Beijing, China in 2018. She is currently working toward her doctorate degree in the Image Processing Center, School of Astronautics, Beihang University.

Rui Zhao received his B.S. degree and M.S. degree from the Image Processing Center, School of Astronautics, Beihang University in 2019 and 2022, respectively. He is currently a researcher in Netease Fuxi AI Lab. His research interests include computer vision, deep learning, and related problems in remote sensing and video games.

Zhenwei Shi (Member IEEE) received the Ph.D. degree in mathematics from Dalian University of Technology, Dalian, China, in 2005.

He was a Post-Doctoral Researcher with the Department of Automation, Tsinghua University, Beijing, China, from 2005 to 2007. He was Visiting Scholar with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, USA., from 2013 to 2014. He is currently a Professor and Dean of the Image Processing Center, School of Astronautics, Beihang University, Beijing. He has authored or coauthored over 200 scientific articles in refereed journals and proceedings, including the IEEE Transactions on Pattern Analysis and Machine Intelligence, the IEEE Transactions on Image Processing, the IEEE Transactions on Geoscience and Remote Sensing, the IEEE Geoscience and Remote Sensing Letters, the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) and the IEEE International Conference on Computer Vision (ICCV). His current research interests include remote sensing image processing and analysis, computer vision, pattern recognition, and machine learning.

Dr. Shi serves as an Editor for the Pattern Recognition, the ISPRS Journal of Photogrammetry and Remote Sensing, and the Infrared Physics and Technology, etc. His personal website is http://levir.buaa.edu.cn/.