Bayesian Sharpness-Aware Prompt Tuning for Cross-Domain Few-shot Learning

Shuo Fan, Liansheng Zhuang*, Aodi Li
University of Science and Technology of China, Hefei, China
Email:{fs199816@mail.ustc.edu.cn, lszhuang@ustc.edu.cn}

2023 International Joint Conference on Neural Networks (IJCNN) | 978-1-6654-8867-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/IJCNN54540.2023.10191224

***Abstract*—Few-shot learning aims to learn a classifier to recognize novel classes with only few labeled images in each class. Fine-tuning is a promising tool to solve the few-shot learning problem, which pre-trains a large-scale model on source domains and then adapts it to target domains. However, existing methods have poor generalization when encountering the domain-shifting problem in the cross-domain scenario. Inspired by recent advances on domain generalization and prompt-based tuning methods, this paper proposes Bayesian Sharpness-Aware Prompt Tuning (BSAPT) for the cross-domain few-shot learning task. Instead of learning deterministic prompts like existing methods, our BSAPT learns a weight distribution over prompts to model the uncertainty caused by limited training data and resist overfitting. To improve the generalization ability, our BSAPT seeks the prompts which lie in neighborhoods having uniformly low loss by simultaneously minimizing the training loss value and loss sharpness. Benefiting from deterministic pre-trained training and Bayesian inference, our BSAPT has better generalization ability and less overfitting than existing fine-tuning methods. Extensive experiments on public datasets show that our BSAPT outperforms state-of-the-art methods and achieves new state-of-the-art performance in the cross-domain few-shot learning task.**

***Index Terms*—few-shot learning, visual prompt tuning, Bayesian inference, model generalization**

I. Introduction

The past decade has witnessed the great successes in the fields of computer vision and natural language processing, where a large number of labeled datasets are available. Unfortunately, in many application scenarios such as rare diseases or new species [1], it is difficult or even impossible to collect sufficient labeled data. Traditional training approaches for deep neural networks often fail to perform well when they suffer from data hunger and extensively high computation time and resources. This phenomenon inspired researchers to propose few-shot learning (FSL) task [2], which aims to learn a classifier to recognize novel classes with only a few labeled examples. FSL is an essential step towards human-like learning systems that can quickly learn new concepts with limited labeled samples and generalize to many new images.

Many approaches have been proposed to address the FSL problem [3]–[8]. In essence, most existing FSL methods are based on transfer learning [9], which transfers prior knowledge from a source domain to the target domain. Fine-tuning is the most widely used method for transfer learning [7], [8], which pre-trains a large-scale base model (such as ViT [10]) on the source domain and adapts it to the target domain. The most straightforward adaptation strategy is fully fine-tuning the pre-trained model on the labeled data from the target domain. However, since the parameter quantity of large-scale model is too large, fine-tuning all the parameters requires massive computation and storage overheads and often results in overfitting [11]. Alternatively, some works only fine-tune a few layers while freezing the remaining layers with pre-trained values [12], [13]. However, this strategy often underperforms fully fine-tuning in accuracy. Instead of fine-tuning the pre-trained base model itself, prompt-based tuning methods [14] modify the input of the base model. It introduces a small amount of task-specific trainable parameters (namely visual prompts) into the input space while keeping the whole pre-trained backbone frozen. Since only a few parameters (less than 1% of backbone parameters) are required to be updated, prompt-based fine-tuning not only dramatically reduces computation and storage overheads but also obtains competitive performance compared to full fine-tuning [14].

Despite the significant progress, it is still a challenging problem to deploy the fine-tuning methods. First, existing fine-tuning methods are limited to a restrictive setting where base classes and novel classes are sampled from the same domain. When there is a large domain shifting like in the cross-domain scenario, existing fine-tuning methods have poor generalization ability [1], [15]. Secondly, in the fine-tuning stage, existing methods only optimize the training loss value, which provides few guarantees on model generalization ability, and easily leads to sub-optimal model parameters [16]. Third, existing fine-tuning methods tend to build deterministic models while ignoring the uncertainty in the training data, which often leads to overconfident predictions [17].

Inspired by recent advances on domain generalization, Bayesian neural network and prompt-based tuning methods, this paper proposes a novel Bayesian Sharpness-Aware Prompt Tuning (BSAPT) method for cross-domain few-shot learning (CDFSL). Following the framework of Visual Prompt Tuning (VPT) [14], we choose the popular Transformer [18] model ViT [10] as our base model and fine-tune it in the target domain by adding trainable visual prompt tokens in input space. Different from VPT who learns deterministic visual prompts, our BSAPT learns the probability distribution of visual prompts (namely Bayesian Prompts). By combining visual prompts with Bayesian method, our BSAPT provides a

*This work was supported in part to Dr. Liansheng Zhuang by NSFC under contract No.U20B2070 and No.61976199. Dr. Liansheng Zhuang is the corresponding author.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on April 23,2024 at 01:32:05 UTC from IEEE Xplore. Restrictions apply.

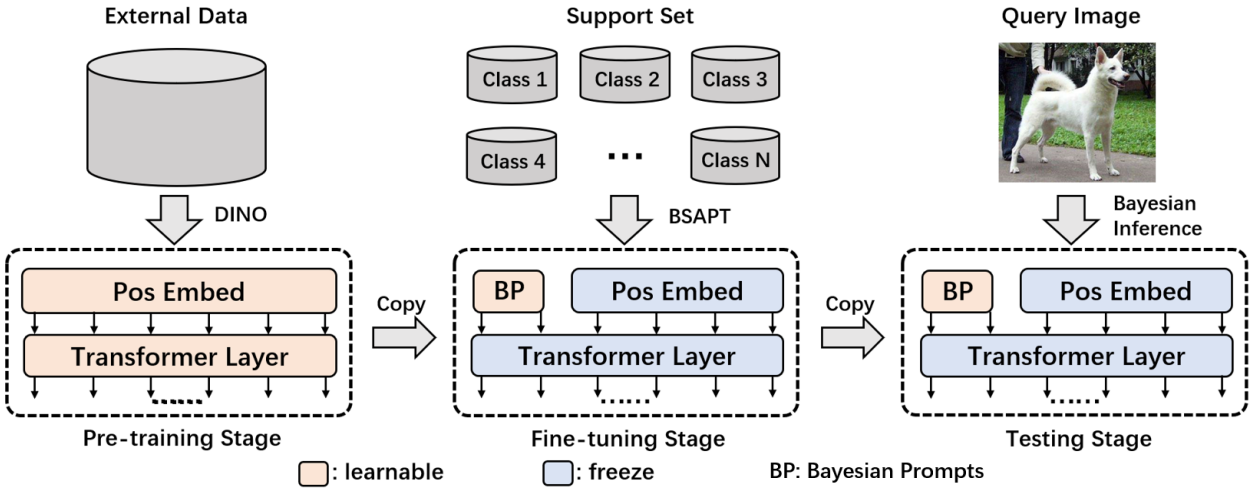

Fig. 1. The framework of our approach to solving the FSL problem with BSAPT. In the pre-training stage, we use the DINO strategy to pre-train our ViT backbone on the external data from the source domain; In the fine-tuning stage, we propose a novel BSAPT method to update Bayesian Prompts and fine-tune the model in the support set of basic task; In the testing stage, we use Bayesian inference and Monte Carlo sampling to classify the image in the query set of basic task.

natural framework to model the inherent uncertainty caused by the limited training dataset and resist both over-fitting and over-confidence on prediction results. To improve the generalization ability, motivated by prior work connecting the geometry of the loss landscape and generalization [16], our BSAPT modifies the training process to seek the Bayesian Prompts that lie in neighborhoods having uniformly low loss by simultaneously minimizing the training loss value and loss sharpness. Extensive experiments on the most popular FSL benchmark (*i.e.*, Meta-Dataset) show that our BSAPT consistently outperforms existing SOTA FSL methods and achieves SOTA performances under the setting of cross-domain few-shot learning.

In summary, our main contributions are as follows:

- A novel fine-tuning method namely BSAPT is introduced for the CDFSL task. In comparison to existing fine-tuning methods, our BSAPT has better generalization and less overfitting. Moreover, it is effective and lightweight.
- A new Bayesian Prompts module is introduced by combing the Visual Prompts with Bayesian method. Inheriting the advantages of Bayesian deep learning, the Bayesian Prompts module provides a natural framework to capture the model uncertainty caused by the limited training data.
- Extensive experiments on Meta-Dataset demonstrate that our BSAPT consistently outperforms existing FSL methods and achieves new SOTA performance.

II. Methodology

A. Architecture Overview

Following the FSL setting in [19], [20], we formulate the FSL problem as an N-way K-shot learning task. Each basic task contains N classes, and each class provides K labeled training examples as support set $\mathcal{D}_{\text{train}} = \{(x_i^j, y_i) | 1 \leq i \leq N,\ 1 \leq j \leq K\}$, where x_i^j is the j-th training sample of the i-th class and y_i is the label of the i-th class. There are also some test samples $\mathcal{D}_{\text{test}} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_m\}$ sampled from the same N classes, and we denote $\mathcal{D}_{\text{test}}$ as the query set. The target of basic task is to learn a classifier to recognize the test samples in query set with limited support set.

As shown in Fig.1, the pipeline of our FSL method consists of three stages: pre-training stage, fine-tuning stage and testing stage. In the pre-training stage, we adopt the popular Transformer [18] architecture ViT [10] as our feature backbone and use DINO [21] to pre-train the ViT on external data (*i.e.,* base classes). In the fine-tuning stage, we propose Bayesian Sharpness-Aware Prompt Tuning (namely BSAPT) method, which introduces Bayesian Prompts and sharpness-aware minimization [16] into the fine-tuning strategy of VPT [14] and thus effectively enhance the generalization ability of the model. In the testing stage, we use the distribution in Bayesian Prompts to construct base learners and use Bayesian inference to give low variance prediction approximated by Monte Carlo sampling.

In this paper, we focus on the fine-tuning stage and propose a novel BSAPT method to fine-tune the pre-trained model from base classes to novel classes, aiming to improve the generalization ability of the fine-tuned model. From a high-level perspective, our BSAPT combines the advantages of VPT and Bayesian deep learning and seeks the weights distribution of those prompts with solid generalization ability, according to recent advances on connecting the geometry of the loss landscape and generalization ability.

B. Bayesian Prompt Tuning

The shortage of labeled data often leads to the uncertainty of model parameters (namely model uncertainty) and raises a significant risk of overfitting. Bayesian methods maintain a distribution over model parameters and thus provide a natural framework for capturing this inherent model uncertainty. Motivated by the insights, this paper extends the Visual

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on April 23,2024 at 01:32:05 UTC from IEEE Xplore. Restrictions apply.

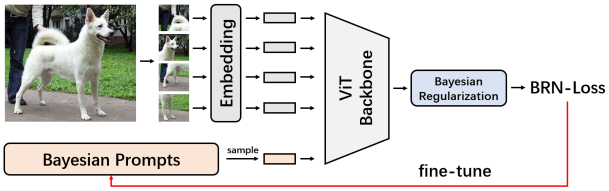

Fig. 2. The illustration of the training process of BPT method in the fine-tuning stage. BPT uses Bayesian Prompts to generate prompts and model the uncertainty of the limited training dataset.

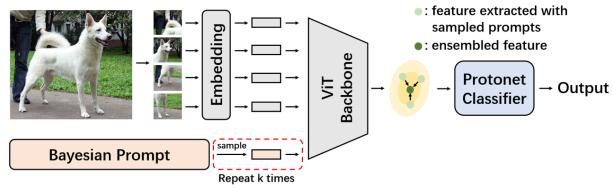

Fig. 3. The illustration of the inference process of the BPT method. BPT use ensemble learning to classify query sample with Bayesian inference and Monte Carlo sampling.

Prompt Tuning (VPT) to Bayesian Prompt Tuning (BPT) and learns the weights distribution of the Visual Prompts (namely Bayesian Prompts).

Bayesian Prompts. Following the framework of VPT [14], for an image $x \in \mathbb{R}^{3\times H\times W}$, we divide it into small patches $\{\mathbf{I}_j \in \mathbb{R}^{3\times h\times w} \in \mathbb{N}, 1 \leq j \leq m\}$, where h and w are the height and width of the image patches. Then we use the position embedding layer to embed each of the image patches into a d-dimensional feature:

$$\mathbf{e}_0^j = \text{Embed}(\mathbf{I}_j), \qquad \mathbf{e}_0^j \in \mathbb{R}^d, j = 1, 2, \ldots, m. \quad (1)$$

Let L_i be the i-th Transformer layer and $\mathbf{E}_i$ be the output of the i-th Transformer layer (together with the input of the first Transformer layer as $\mathbf{E}_0$). BSAPT aims to provide learnable prompt tokens T_i as parts of the inputs to the (i+1)-th Transformer layer as VPT does:

$$\mathbf{T}_i = \{\mathbf{t}_i^j \in \mathbb{R}^d |\ j = 0, 1, \cdots, N\} \quad (2)$$

$$[\mathbf{c}_{i+1}, \mathbf{E}_{i+1}] = \mathrm{L}_{i+1}([\mathbf{c}_i, \mathbf{T}_i, \mathbf{E}_i]), \quad (3)$$

where $\mathbf{c} \in \mathbb{R}^d$ denotes the embedding of classification token. We denote all of the prompt tokens as $\mathbf{w}_\mathrm{t} = \{\mathbf{T}_i, 0 \leq i \leq N_{\text{layer}}\}$, and the remaining frozen parameters are not mentioned in following formulas.

To capture the model uncertainty, we assume that each parameter of the prompt tokens can be described by a distribution function $\mathcal{F}$ with the parameters θ:

$$\mathbf{t}_i^j[k] \sim \mathcal{F}(\theta(i, j, k)), \quad k \in \mathbb{N}, 1 \leq k \leq d. \quad (4)$$

Previous studies have proven that Gaussian-based prior distribution yields L_2 regularization for the weights [17], which can enhance the model generalization ability. Therefore, this paper chooses the diagonal two-dimensional Gaussian distribution $q(\mathbf{w}_\mathrm{t}|\theta)$ as the distribution function $\mathcal{F}(\theta)$ to approximate the real posterior distribution:

$$q(\mathbf{w}_\mathrm{t}|\theta) = \prod_j (\alpha\mathcal{N}(\mathbf{w}_\mathrm{t}^j|\phi_1) + \beta\mathcal{N}(\mathbf{w}_\mathrm{t}^j|\phi_2)) \quad (5)$$

$$\log q(\mathbf{w}_\mathrm{t}|\theta) = \sum_j \log(\alpha\mathcal{N}(\mathbf{w}_\mathrm{t}^j|\phi_1) + \beta\mathcal{N}(\mathbf{w}_\mathrm{t}^j|\phi_2)), \quad (6)$$

where $\phi_1 = \{\mu_1, \sigma_1\}$ and $\phi_2 = \{\mu_2, \sigma_2\}$ are the parameters θ of the Gaussian distribution, μ_i and σ_i are the mean and variance of the Gaussian distribution, $0 \leq \alpha, \beta \leq 1$ and $\alpha + \beta = 1$, $\mathcal{N}$ is the Gaussian distribution, and $\mathbf{w}_\mathrm{t}^j$ represents the j-th parameter in $\mathbf{w}_\mathrm{t}$.

To initialize our prior distribution, we set $\mu_1 = \mu_2 = 0$ and $\sigma_1 >> \sigma_2 > 0$, where σ_2 is very close to 0. We can regard the second component of the Gaussian mixture distribution with zero mean value and minimal variance σ_2 as a dropout-like module, for the weight could be sampled as a value very close to 0 with a probability $(1 - \pi)$. This distribution provides the ability to drop the output of this weight with a probability of $(1 - \pi)$, which can increase the generalization ability of our model and reduce overfitting.

Training & Inference. To learn the optimal parameter θ^* of Bayesian Prompts, we minimize the Kullback-Leibler (KL) divergence between $q(\mathbf{w}_\mathrm{t}|\theta)$ and the true posterior distribution on the weights:

$$\begin{aligned}\theta^* &= \arg\min_\theta \mathrm{KL}[q(\mathbf{w}_\mathrm{t}|\theta) \| P(\mathbf{w}_\mathrm{t}|\mathcal{D}_{\text{train}})] \\ &= \arg\min_\theta \Big(\int q(\mathbf{w}_\mathrm{t}|\theta) \log \frac{q(\mathbf{w}_\mathrm{t}|\theta)}{P(\mathbf{w}_\mathrm{t})} \mathrm{d}\mathbf{w}_\mathrm{t} \\ &\qquad - \mathbb{E}_{q(\mathbf{w}_\mathrm{t}|\theta)}[\log P(\mathcal{D}_{\text{train}}|\mathbf{w}_\mathrm{t})] \Big). \quad (7)\end{aligned}$$

We can denote (7) as a loss optimization problem:

$$\begin{aligned}\mathcal{L}(\mathcal{D}_{\text{train}}, \theta) &= \int q(\mathbf{w}_\mathrm{t}|\theta) \log \frac{q(\mathbf{w}_\mathrm{t}|\theta)}{P(\mathbf{w}_\mathrm{t})} \mathrm{d}\mathbf{w}_\mathrm{t} \\ &\qquad - \mathbb{E}_{q(\mathbf{w}_\mathrm{t}|\theta)}[\log P(\mathcal{D}_{\text{train}}|\mathbf{w}_\mathrm{t})]. \quad (8)\end{aligned}$$

The integral in (8) can hardly be calculated directly, but it can be approximated by Monte Carlo sampling. With the proposition of [17] about the commutativity of integrals and derivatives, we have our Bayesian Regularized NLL-Loss (BRN-Loss) $\mathcal{L}_{\text{BRN}}$ as:

$$\begin{aligned}\mathcal{L}_{\text{BRN}}(\mathcal{D}_{\text{train}}, \theta) &\approx \sum_{i=1}^{n} (\log q(\mathbf{w}_\mathrm{t}^i|\theta) - \log P(\mathbf{w}_\mathrm{t}^i) \\ &\qquad - \log P(\mathcal{D}_{\text{train}}|\mathbf{w}_\mathrm{t}^i)), \quad (9)\end{aligned}$$

where $q(\mathbf{w}_\mathrm{t}^i|\theta)$, $P(\mathbf{w}_\mathrm{t}^i)$ and $-\log P(\mathcal{D}_{\text{train}}|\mathbf{w}_\mathrm{t}^i)$ denote the variational approximation of the Bayesian posterior distribution of $\mathbf{w}_\mathrm{t}$, the prior probability of $\mathbf{w}_\mathrm{t}$, and the Negative Log Likelihood Loss (NLL-Loss) of $\mathcal{D}_{\text{train}}$. $\mathbf{w}_\mathrm{t}^i$ represents the i-th Monte Carlo sampling results from $q(\mathbf{w}_\mathrm{t}|\theta)$. The complete training process of BPT is shown in Fig.2.

After learning Bayesian Prompts, we can use it to predict the class of a query example $\hat{x}$ by

$$P(\hat{y}|\mathcal{D}_{\text{train}}, \hat{x}) = \int P(\hat{y}|\mathbf{w}_\mathrm{t}) P(\mathbf{w}_\mathrm{t}|\mathcal{D}_{\text{train}}, \hat{x}) \mathrm{d}\mathbf{w}_\mathrm{t}. \quad (10)$$

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on April 23,2024 at 01:32:05 UTC from IEEE Xplore. Restrictions apply.

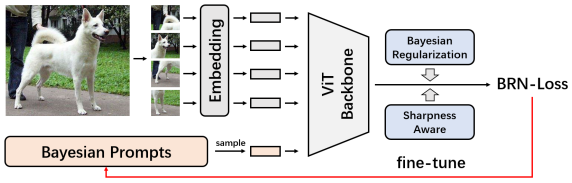

Fig. 4. The illustration of the training process of BSAPT method in the fine-tuning stage.

Since the hypothesis space of posterior distribution is too large, the integral in (10) is difficult to calculate directly. Fortunately, it can be approximated by the density distribution provided by Bayesian Prompts. Specifically, we can use Monte Carlo sampling to approximate the Bayesian inference by using discrete sampling to approximate continuous integral:

$$P(\hat{y}|\mathcal{D}_{\text{train}}, \hat{x}) \approx \frac{1}{N}\sum_{i=1}^{N} P(\hat{y}|\mathbf{w}_{\text{t}}^{i}) P(\mathbf{w}_{\text{t}}^{i}|\mathcal{D}_{\text{train}}, \hat{x}). \tag{11}$$

The complete inference process of BPT is shown in Fig.3. Note here that, to avoid bias and increase the generalization ability, we should not use the mean value of the Gaussian distribution as a fixed weight directly in the testing stage. This is because VPT fine-tunes the model by modifying its input, and the mean value of prompt tokens (as input) does not guarantee the output being the expectation of the output with Monte Carlo sampling.

C. Bayesian Sharpness-Aware Prompt Tuning.

Obviously, the landscape of the BPT training loss $\mathcal{L}_{\text{BRN}}$ is very complex and non-convex with multiple local and global minima. There are numerous recent studies [22], [23] on the connecting between the geometry of the loss landscape and generalization, and it has been shown that simply minimizing the training loss value on the training set is typically not sufficient to achieve satisfactory generalization.

To improve the generalization ability of the fine-tuned models, following the work of [16], we simultaneously minimize the training loss value and loss sharpness. Specifically, we modify the gradient of $\mathcal{L}_{\text{BRN}}$ and obtain a new loss namely Bayesian Sharpness-aware Regularized NLL-Loss (denoted as BSRN-Loss or $\mathcal{L}_{\text{BSRN}}$). The gradient can be approximated by

$$\nabla_{\mathbf{w}_{\text{t}}}\mathcal{L}_{\text{BSRN}}(\mathbf{w}_{\text{t}}, \mathcal{D}_{\text{train}}) \approx \nabla_{\mathbf{w}_{\text{t}}}\mathcal{L}_{\text{BRN}}(\mathbf{w}_{\text{t}}, \mathcal{D}_{\text{train}})|_{\mathbf{w}_{\text{t}}+\hat{\epsilon}(\mathbf{w}_{\text{t}})}, \tag{12}$$

where $\hat{\epsilon}(\mathbf{w}_{\text{t}})$ is the adversarial weight perturbations. It helps the model to find the worst-case point in the neighborhood of model parameter, so as to carry out further optimization. Following the work of [16], $\hat{\epsilon}(\mathbf{w}_{\text{t}})$ can be denoted as

$$\hat{\epsilon}(\mathbf{w}_{\text{t}}) = \frac{\rho \text{sign}(\nabla_{\mathbf{w}_{\text{t}}}\mathcal{L}_{\text{BRN}}(\mathbf{w}_{\text{t}}))|\nabla_{\mathbf{w}_{\text{t}}}\mathcal{L}_{\text{BRN}}(\mathbf{w}_{\text{t}})|^{q-1}}{(\|\nabla_{\mathbf{w}_{\text{t}}}\mathcal{L}_{\text{BRN}}(\mathbf{w}_{\text{t}})\|_{q}^{q})^{1/p}}, \tag{13}$$

where $\rho \geq 0$ is a hyperparameter set according to [16]. The complete training process of BSAPT is shown in Fig.4.

III. Experiments

In this section, we first present the experiment settings in details, and give the experimental results.

TABLE I
PERFORMANCES OF BSAPT WITH DIFFERENT PROMPT TOKEN NUMBERS ON QUICKDRAW DATASET.

num	2	4	6	8	10	12	14	16	18
acc	73.1	73.3	72.8	73.1	74.2	74.0	**74.7**	74.0	74.1

TABLE II
COMPARISON OF THE PARAMETER NUMBER OF DIFFERENT METHODS.

method	backbone	parameters	ratio
AF-MAML [25]	ResNet12	≈12M	≈55%
ProtoNet [3]	ResNet18	≈11.7M	≈54%
BOHB [26]			
CTX [27]	ResNet34	≈21.5M	≈99%
PMF [7]	ViT-small	21665667	100%
BSAPT(ours)	ViT-small	**129026**	**0.59%**

A. Experimental Setting

Datasets. To verify the effectiveness of our proposed BSAPT, we conduct extensive experiments on the large-scale benchmark, Meta-Dataset [24]. It consists of 10 public and widely used image datasets from a diverse range of domains: ILSVRC-2012 (ImageNet-1k), Omniglot, Aircraft, CUB-200-2011, Describable Textures, QuickDraw, Fungi, VGG Flower, Traffic Signs, and MSCOCO. Each dataset has train/validation/test splits with about 70%/15%/15% proportions.

Evaluation Metrics. To evaluate the effectiveness of our proposed BSAPT, this paper adopts two evaluation metrics: Average Accuracy and Average Rank. Both of them evaluate the comprehensive performances of the method on all datasets. The Average Accuracy is calculated by averaging the method accuracy on each dataset. The Average Rank is calculated by averaging the accuracy rank of the method in all methods on each dataset. Note here that, we give the same rank when the accuracy is very close to each other.

Implementation Details. We use PyTorch [28] to implement all experiments on a Tesla V100-32GB GPU. In the fine-tuning stage, we use Adam optimizer to train our model with one of the learning rates in $\{0.01, 0.001, 0.0001\}$ selected according to [7] and the betas are $(0.9, 0.999)$. To give a convincing experimental results, we simulate 50 basic tasks from the test-split of each domain. Following the setting of [7], [24], each task has about 275 support images and 150 query images on average. The total number of query images of each domain is about 7562, which is sufficient to give a low-variance performance.

except for the main experience, for other experience like setting the hyperparameters and ablation study, we randomly choose 4 datasets: Omniglot, DTD, QuickDraw,MSCOCO datasets.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on April 23,2024 at 01:32:05 UTC from IEEE Xplore. Restrictions apply.

TABLE III
EXPERIMENTAL RESULTS OF BSAPT METHOD WITH DIFFERENT PRIOR DISTRIBUTIONS ON THE DIFFERENT SPLIT.

Data Split	Method	Omglot	Acraft	CUB	DTD	QDraw	Fungi	Flower	Sign	COCO	Avg
Support	vanilla	89.13	77.2	94.47	90.49	70.95	86.81	99.23	72.28	74.32	83.88
	BSAPT(single)	99.52	95.06	98.69	98.16	89.55	92.84	99.71	97.48	92.34	95.92
	BSAPT(mixed)	99.74	95.13	97.53	98.72	91.2	92.93	99.23	98.47	89.86	95.86
Query	vanilla	54.84	62.39	84.03	79.68	58.99	56.11	96.26	55.29	57.5	67.23
	BSAPT(single)	79.21	**77.03**	86.93	82.14	74.37	57.91	96.68	89.64	65.04	78.78
	BSAPT(mixed)	**80.60**	76.39	**88.61**	**82.80**	**74.68**	**59.38**	**96.87**	**91.22**	**65.83**	**79.59**

TABLE IV
EXPERIMENTAL RESULTS OF BSAPT ON META-DATASET.

Method	Omglot	Acraft	CUB	DTD	QDraw	Fungi	Flower	Sign	COCO	Avg.Acc	Avg.Rank
ProtoNet [3]	59.98	53.1	68.79	66.56	48.96	39.71	85.27	47.12	41	56.72	6.0
AF-MAML [25]	61.87	63.43	69.75	70.78	59.17	41.49	85.96	60.78	48.11	62.37	4.5
BOHB [26]	67.57	54.12	70.69	68.34	50.33	41.38	87.34	51.8	48.03	59.95	4.4
CTX [27]	**82.21**	**79.49**	80.63	75.57	72.68	51.58	95.34	82.65	59.9	75.56	2.4
PMF [7]	80.68	76.78	85.04	**86.63**	71.25	54.78	94.57	88.33	62.57	77.84	2.0
BSAPT(ours)	80.60	76.39	**88.61**	82.80	**74.68**	**59.38**	**96.87**	**91.22**	**65.83**	**79.59**	**1.4**

B. Main Results

Determine Hyperparameters. To determine the optimal number of prompt tokens, we evaluated the performance of our BSAPT method with different numbers of prompt tokens on QuickDraw. This dataset is widely recognized for its moderate difficulty, making it a reliable representative dataset. As shown in Table I, our BSAPT method achieves the best performance when the number of prompt tokens is 14. To choose the appropriate probability distribution for Bayesian Prompts, we design experiments to compare the accuracy of the BSAPT method with different probability distributions on the support set and query set. As shown in Table III, the BSAPT with both single and Gaussian mixture distributions significantly improve the accuracy of models in the support set, while the BSAPT with Gaussian mixture distribution can further improve the model accuracy in the query set. Above experiments show that the Gaussian mixture distribution does alleviate the overfitting problem and improve the generalization ability of the model. The hyperparameter settings not mentioned are the same as our backbone [7].

Comprehensive Comparative Experiment. To make a broader comparison, we collect many popular CDFSL methods such as ProtoNet [3], ALFA+FP-MAML [25], BOHB [26], CTX [27] and PMF [7], where PMF is a fine-tuning FSL method and outperforms the previous SOTA in CDFSL conditions. For fairness, we choose ViT-small [10] as the backbone of PMF and our proposed BSAPT. The backbone and trainable parameter number of all mentioned methods including our proposed BSAPT method are shown in Table II. We calculate the parameter number of ResNet18 and ResNet34 according to [29], and the parameter number of ResNet12 according to [11]. Our proposed BSAPT method has far fewer trainable parameters than other algorithms, which is beneficial to save storage space for fine-tuning models.

Then we conduct all mentioned methods on Meta-Dataset in CDFSL scenario. Following the CDFSL training and testing protocol proposed by [7], we pre-train the model on the training split of ImageNet and test the performance on the test splits of the remaining nine datasets, and the corresponding experimental results are shown in Table IV. The experimental results show that our BSAPT method achieves SOTA under the evaluation metrics of both Average Accuracy and Average Rank.

Ablation Analysis. Further ablation experiments are conducted to prove the necessity of each module in our BSAPT method. To let our experiments cover situations as much as possible, we randomly chose 4 datasets: MSCOCO, DTD, Omniglot and QuickDraw from Meta Dataset to conduct detailed ablation experiments. To demonstrate the necessity of each module in BSAPT, we test the performance of VPT, VPT+Bayesian Prompts (our BPT method), and VPT+Bayesian Prompt+SAM (our BSAPT method). The experimental results shown in Fig.5 not only prove that our proposed BSAPT method achieves the best performance, but also prove the necessity of each module in our BSAPT method. In addition, we follow the work of [30] to visualize the loss function of two randomly selected meta-tasks from the test phase. As shown in Fig.6, using SAM regularized can help the model to find optimal parameters lying in neighborhoods with flattened low loss.

IV. CONCLUSION

This paper proposes a novel Bayesian Sharpness-Aware Prompt Tuning (namely BSAPT) method to deal with the FSL problem. Unlike existing prompt-based tuning methods, BSAPT fine-tunes the popular Transformer model to adapt to novel classes by learning Bayesian Prompts to capture the uncertainty of limited training data and prevent overfitting. To

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on April 23,2024 at 01:32:05 UTC from IEEE Xplore. Restrictions apply.

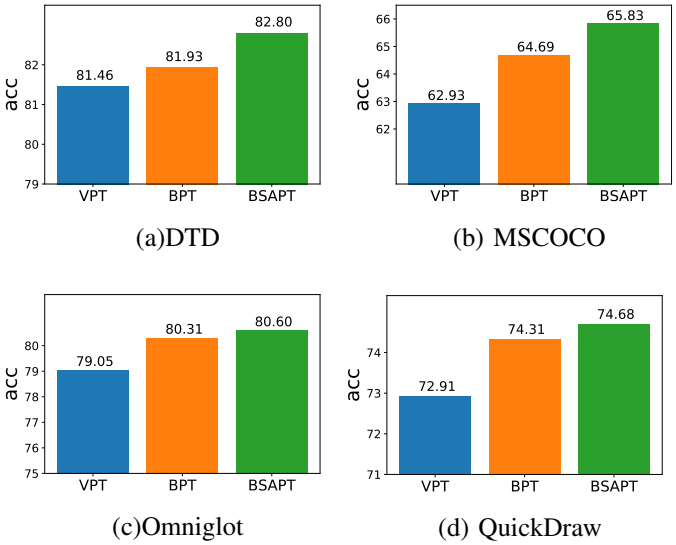

Fig. 5. Ablation Experiments of BSAPT on different datasets.

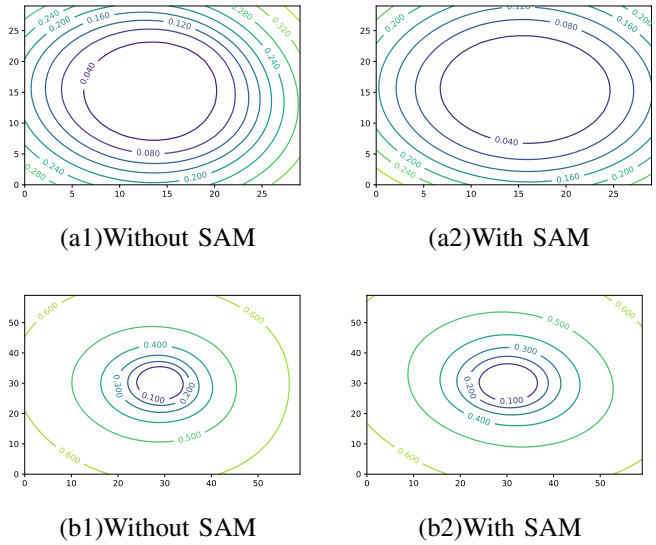

Fig. 6. Loss visualization of BSAPT with/without SAM regularization technique.

further improve the generalization ability, our BSAPT seeks the Bayesian Prompts whose weights lie in neighborhoods with uniformly low loss. In the testing stage, we use Bayesian inference and Monte Carlo sampling to sample multiple prompt token weights from Bayesian Prompts and then use the ensemble learning strategy to give the final prediction with low variance and high confidence. Our BSAPT has better generalization with less overfitting compared with existing fine-tuning methods. Extensive experiments on the most popular CDFSL benchmark (Meta-Dataset) also justify the effectiveness and superiority of our BSAPT, which outperforms SOTA FSL methods and achieves new SOTA performance.

References

[1] Yunhui Guo, Noel C Codella, Leonid Karlinsky, James V Codella, John R Smith, Kate Saenko, Tajana Rosing, and Rogerio Feris, "A broader study of cross-domain few-shot learning," in *European conference on computer vision*. Springer, 2020, pp. 124–141.

[2] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang, "A closer look at few-shot classification," *arXiv preprint arXiv:1904.04232*, 2019.

[3] Jake Snell, Kevin Swersky, and Richard Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[4] Chelsea Finn, Pieter Abbeel, and Sergey Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 1126–1135.

[5] Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn, "Bayesian model-agnostic meta-learning," *Advances in neural information processing systems*, vol. 31, 2018.

[6] Massimiliano Patacchiola, Jack Turner, Elliot J Crowley, Michael O'Boyle, and Amos J Storkey, "Bayesian meta-learning for the few-shot setting via deep kernels," *Advances in Neural Information Processing Systems*, vol. 33, pp. 16108–16118, 2020.

[7] Shell Xu Hu, Da Li, Jan Stühmer, Minyoung Kim, and Timothy M Hospedales, "Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9068–9077.

[8] Yanbin Liu, Juho Lee, Linchao Zhu, Ling Chen, Humphrey Shi, and Yi Yang, "A multi-mode modulator for multi-domain few-shot classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8453–8462.

[9] Sinno Jialin Pan and Qiang Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[11] Yinbo Chen, Zhuang Liu, Huijuan Xu, Trevor Darrell, and Xiaolong Wang, "Meta-baseline: Exploring simple meta-learning for few-shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9062–9071.

[12] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson, "How transferable are features in deep neural networks?," *Advances in neural information processing systems*, vol. 27, 2014.

[13] Yunhui Guo, Honghui Shi, Abhishek Kumar, Kristen Grauman, Tajana Rosing, and Rogerio Feris, "Spottune: transfer learning through adaptive fine-tuning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4805–4814.

[14] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim, "Visual prompt tuning," *arXiv preprint arXiv:2203.12119*, 2022.

[15] Hung-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, and Ming-Hsuan Yang, "Cross-domain few-shot classification via learned feature-wise transformation," *arXiv preprint arXiv:2001.08735*, 2020.

[16] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur, "Sharpness-aware minimization for efficiently improving generalization," *arXiv preprint arXiv:2010.01412*, 2020.

[17] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra, "Weight uncertainty in neural network," in *International conference on machine learning*. PMLR, 2015, pp. 1613–1622.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[19] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al., "Matching networks for one shot learning," *Advances in neural information processing systems*, vol. 29, 2016.

[20] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM computing surveys (csur)*, vol. 53, no. 3, pp. 1–34, 2020.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on April 23,2024 at 01:32:05 UTC from IEEE Xplore. Restrictions apply.

[21] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9650–9660.

[22] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina, "Entropy-sgd: Biasing gradient descent into wide valleys," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2019, no. 12, pp. 124018, 2019.

[23] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," *arXiv preprint arXiv:1609.04836*, 2016.

[24] Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, et al., "Meta-dataset: A dataset of datasets for learning to learn from few examples," *arXiv preprint arXiv:1903.03096*, 2019.

[25] Sungyong Baik, Myungsub Choi, Janghoon Choi, Heewon Kim, and Kyoung Mu Lee, "Meta-learning with adaptive hyperparameters," *Advances in Neural Information Processing Systems*, vol. 33, pp. 20755–20765, 2020.

[26] Tonmoy Saikia, Thomas Brox, and Cordelia Schmid, "Optimized generic feature learning for few-shot classification across domains," *arXiv preprint arXiv:2001.07926*, 2020.

[27] Carl Doersch, Ankush Gupta, and Andrew Zisserman, "Crosstransformers: spatially-aware few-shot transfer," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21981–21993, 2020.

[28] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in pytorch," 2017.

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

[30] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein, "Visualizing the loss landscape of neural nets," *Advances in neural information processing systems*, vol. 31, 2018.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on April 23,2024 at 01:32:05 UTC from IEEE Xplore. Restrictions apply.