Accelerating Convergence in Bayesian Few-Shot Classification

Tianjun Ke [1] **Haoqun Cao** [1] **Feng Zhou** [1 2]

Abstract

Bayesian few-shot classification has been a focal point in the field of few-shot learning. This paper seamlessly integrates mirror descent-based variational inference into Gaussian process-based few-shot classification, addressing the challenge of non-conjugate inference. By leveraging non-Euclidean geometry, mirror descent achieves accelerated convergence by providing the steepest descent direction along the corresponding manifold. It also exhibits the parameterization invariance property concerning the variational distribution. Experimental results demonstrate competitive classification accuracy, improved uncertainty quantification, and faster convergence compared to baseline models. Additionally, we investigate the impact of hyperparameters and components. Code is publicly available at https://github.com/keanson/MD-BSFC.

1. Introduction

Humans have the ability to learn new skills and adapt to new environments based on very few instances. In contrast, most machine learning techniques, especially deep learning, require vast amounts of examples to achieve similar performance and may yet struggle to generalize. The reason humans are more advanced than machines in adapting and generalizing is that they can leverage prior experience to solve new tasks. It is thus of great interest to design machine learning algorithms that can generalize to novel tasks with a limited amount of data.

Few-shot classification focuses on the classification of new data given only a limited number of training samples with class labels. It proves particularly useful when collecting training examples is challenging or when annotating data incurs high costs. The scarcity of labeled data in such situations introduces uncertainty over model parameters, commonly referred to as epistemic uncertainty. Effectively managing epistemic uncertainty serves to regularize the model, mitigating the risk of overfitting. Additionally, this uncertainty plays a crucial role in assessing confidence, especially in risk-averse applications like medical diagnosis (Prabhu et al., 2019) and autonomous driving (Bojarski et al., 2016).

The Bayesian framework offers a natural approach to capture epistemic uncertainty by introducing a prior distribution over model parameters and computing the posterior using Bayes' theorem based on observed data. In recent years, there has been a significant surge in research applying Bayesian approaches to few-shot learning (Yoon et al., 2018; Finn et al., 2018; Ravi & Beatson, 2018). Leveraging the advantages of the Bayesian framework, some recent studies have employed Gaussian processes (GPs) in the context of few-shot classification (FSC), demonstrating competitive performance in accuracy and uncertainty quantification (Patacchiola et al., 2020; Snell & Zemel, 2021; Ke et al., 2023).

Bayesian inference poses challenges for Gaussian Process (GP) classification due to the non-Gaussian likelihood being non-conjugate to the GP prior, rendering exact posterior computation infeasible. To address this, Patacchiola et al. (2020) assumes a Gaussian likelihood for class labels, achieving conjugacy in inference. However, this approach is not entirely reasonable since class labels are discrete rather than continuous. Snell & Zemel (2021); Ke et al. (2023) combine Pólya-Gamma augmentation (Polson et al., 2013) with the one-vs-each softmax approximation (Titsias, 2016) and logistic-softmax likelihood separately to establish conditionally conjugate inference. These methods provide improved uncertainty quantification but necessitate the introduction of additional auxiliary variables.

Differing from the approaches mentioned earlier, we incorporate mirror descent (Nemirovskij & Yudin, 1985)-based variational inference (Blei et al., 2017) into GP-based FSC without introducing any auxiliary variables. This approach is consequently named *Mirror Descent based Bayesian Few-Shot Classification* (MD-BFSC). In our method, the optimization within variational inference is accomplished through conjugate computation. Notably, mirror descent leverages non-Euclidean geometry, providing the steepest

[1]Center for Applied Statistics and School of Statistics, Renmin University of China, Beijing, China [2]Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing. Correspondence to: Feng Zhou <feng.zhou@ruc.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

descent direction along the corresponding manifold, thereby enhancing the convergence rate. It also exhibits the parameterization invariance property concerning the variational distribution (Raskutti & Mukherjee, 2015).

Specifically, we make several contributions: **(1)** We introduce variational inference based on mirror descent to GP-based FSC, thereby transforming non-conjugate inference into an optimization problem with conjugate computation. **(2)** As demonstrated later, MD-BFSC provides the steepest descent direction along the corresponding non-Euclidean manifold, enhancing the convergence rate and maintaining invariance to the parameterization of the variational distribution. **(3)** We showcase that our approach achieves competitive classification accuracy and uncertainty quantification, demonstrating a faster convergence rate on standard FSC benchmarks in comparison to few-shot baseline models.

2. Background

Our approach involves concepts in few-shot classification, GP classification, variational inference, exponential family, natural gradient descent and mirror descent. We illustrate the basic concepts in each part. The same notation in different sections refers to the same variable unless specified.

2.1. Few-shot Classification

In FSC (Snell et al., 2017b; Miller et al., 2000; Wang et al., 2020), a classifier must adapt to new classes which are not observed in training, given only a few samples of each class. Consider a L-shot and C-way FSC task s with support set $\mathcal{S}_s = \{\mathbf{x}_l, y_l\}_{l=1}^{L}$ and query set $\mathcal{Q}_s = \{\mathbf{x}_m, y_m\}_{m=1}^{M}$, $\mathbf{x}_m$ is the input feature and $y_m \in \{1, \ldots, C\}$ is the class label. We can sample distinct tasks from a distribution over tasks to form the training dataset $\mathcal{D}_{\text{train}} = \{\mathcal{S}_s, \mathcal{Q}_s\}_{s=1}^{S}$. Similarly, the validation dataset $\mathcal{D}_{\text{validation}}$ and test dataset $\mathcal{D}_{\text{test}}$. Given a new task s^* from the test dataset with support and query set $\{\mathcal{S}_{s^*}, \mathcal{Q}_{s^*}\}$, our goal is to train a classifier on $\mathcal{S}_{s^*}$ based on the regularity extracted from $\mathcal{D}_{\text{train}}$ to predict the label of samples in $\mathcal{Q}_{s^*}$. The validation dataset is used for tuning hyperparameters, e.g., learning rate.

2.2. Gaussian Process Classification with Variational Inference

Consider a multi-class classification task consisting of N samples with the input features $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top$ and the corresponding class labels $\mathbf{y} = [\mathbf{y}_1^\top, \ldots, \mathbf{y}_N^\top]^\top$, where $\mathbf{x}_n \in \mathcal{X} \subset \mathbb{R}^D$ and $\mathbf{y}_n$ is the one-hot encoding for C classes. The multi-class GP classification model (Williams & Rasmussen, 2006) includes latent GP functions for all classes, i.e., $\{f^1, \ldots, f^C\}$ where $f^c(\cdot) : \mathcal{X} \to \mathbb{R}$ is the latent function for c-th class. A GP prior is applied over each latent function $f^c \sim \mathcal{GP}(0, k_{\boldsymbol{\eta}^c})$ where $k_{\boldsymbol{\eta}^c}$ is the GP kernel for c-th class with hyperparameters $\boldsymbol{\eta}^c$. Following tradition, we use the softmax likelihood $p(\mathbf{y}_n|\mathbf{f}_n) = \prod_c \exp(f_n^c \cdot y_n^c) / \sum_c \exp(f_n^c)$ with $f_n^c = f^c(\mathbf{x}_n)$ and $\mathbf{f}_n = [f_n^1, \ldots, f_n^C]^\top$, and the posterior of latent functions can be computed as:

$$p(\mathbf{f}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f})}{p(\mathbf{y})} = \frac{\prod_n p(\mathbf{y}_n|\mathbf{f}_n) \prod_c p(\mathbf{f}^c)}{\int \prod_n p(\mathbf{y}_n|\mathbf{f}_n) \prod_c p(\mathbf{f}^c)\, \mathrm{d}\mathbf{f}}, \tag{1}$$

where $\mathbf{f}^c = [f_1^c, \ldots, f_N^c]^\top$, $\mathbf{f} = [\mathbf{f}^{1\top}, \ldots, \mathbf{f}^{C\top}]^\top$, $p(\mathbf{f}^c) = \mathcal{N}(\mathbf{f}^c|\mathbf{0}, \mathbf{K}^c)$ with $K_{ij}^c = k_{\boldsymbol{\eta}^c}(\mathbf{x}_i, \mathbf{x}_j)$.

However, the exact computation of posterior is infeasible since the non-Gaussian likelihood is non-conjugate to the Gaussian prior. Variational inference (VI) (Blei et al., 2017; Hoffman et al., 2013) is an approximate inference method in which the exact posterior is approximated by a variational distribution $q(\mathbf{f}|\boldsymbol{\theta})$ with $\boldsymbol{\theta}$ being the variational parameter. The optimal variational distribution is obtained by minimising the Kullback-Leibler (KL) divergence between q and the exact posterior, or equivalently maximizing the evidence lower bound (ELBO) $\mathcal{L}(\boldsymbol{\theta}) : \boldsymbol{\Theta} \to \mathbb{R}$ (Bishop, 2006):

$$\arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_q \left[\log p(\mathbf{y}, \mathbf{f}) - \log q(\mathbf{f}|\boldsymbol{\theta})\right],$$

where $\boldsymbol{\Theta}$ is the set of valid variational parameters. In GP classification, the variational distribution q is assumed to be a Gaussian distribution. The GP kernel hyperparameter $\boldsymbol{\eta}$ can be determined through empirical Bayes by maximizing the marginal likelihood.

2.3. Exponential Family

An exponential family (Wainwright & Jordan, 2008) is a set of distributions whose probability density function can be expressed in the form:

$$q(\mathbf{f}|\boldsymbol{\theta}) = h(\mathbf{f}) \exp(\boldsymbol{\theta}^\top \phi(\mathbf{f}) - A(\boldsymbol{\theta})),$$

where $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subset \mathbb{R}^P$ is a vector of natural parameters, $\phi(\mathbf{f})$ is a vector of sufficient statistics, $A(\boldsymbol{\theta}) = \log \int h(\mathbf{f}) \exp(\boldsymbol{\theta}^\top \phi(\mathbf{f})) d\mathbf{f}$ is the log-partition function that is convex. An exponential family is referred to as minimal if each distribution has a unique natural parameter. A minimal exponential family distribution can also be parameterized by the mean parameter that is the mean of sufficient statistics: $\boldsymbol{\mu} = \mathbb{E}_q[\phi(\mathbf{f})] \in \mathcal{M} \subset \mathbb{R}^P$. The mean parameter also can be obtained by the gradient of log-partition function: $\boldsymbol{\mu} = \nabla A(\boldsymbol{\theta})$. The convex conjugate function (Rockafellar, 2015) of $A(\boldsymbol{\theta})$ is the negative entropy function $H(\boldsymbol{\mu}) = \mathbb{E}_q[\log q(\mathbf{f}|\boldsymbol{\theta})] = \boldsymbol{\theta}^\top \boldsymbol{\mu} - A(\boldsymbol{\theta}) + \text{const}$. Similarly, the natural parameter can be obtained by the gradient of negative entropy function $\boldsymbol{\theta} = \nabla H(\boldsymbol{\mu})$. The pair of $\nabla A(\cdot)$ and $\nabla H(\cdot)$ are inverse functions.

2.4. Variational Inference with Natural Gradient Descent

The variational inference using gradient descent can be generalized to that using natural gradient descent (Amari, 1997) which takes the non-Euclidean geometry into account, because each set of parameters corresponds to a distribution (Salimans & Knowles, 2013; Hoffman et al., 2013; Hensman et al., 2013; Salimbeni et al., 2018). Natural gradient descent assumes the ELBO is maximized over a distribution space and the variational parameters lie on a Riemannian manifold. Given an exponential-family variational distribution $q(\mathbf{f}|\boldsymbol{\theta})$, we can denote a Riemannian metric (Fisher information matrix) $\mathbf{I}(\boldsymbol{\theta}) = -\mathbb{E}_q[\nabla^2_{\boldsymbol{\theta}} \log q(\mathbf{f}|\boldsymbol{\theta})] = \nabla^2 A(\boldsymbol{\theta})$ (Rissanen, 1996; Hoffman et al., 2013) to induce a P-dimensional Riemannian manifold $(\boldsymbol{\Theta}, \mathbf{I}(\boldsymbol{\theta}))$. To maximize the ELBO on the Riemannian manifold, we can use the following natural gradient update:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \rho_t \mathbf{I}^{-1}(\boldsymbol{\theta}_t) \nabla \mathcal{L}(\boldsymbol{\theta}_t), \tag{2}$$

where ρ_t is the step size. The natural gradient selects the steepest descent direction along the Riemannian manifold and the optimization path on the Riemannian manifold with infinitesimally small steps is invariant to the parameterization of variational distribution (Martens, 2020).

2.5. Variational Inference with Mirror Descent

Mirror descent (Nemirovskij & Yudin, 1985) is another generalization of gradient descent. Given a specific parameterized ELBO $\mathcal{L}(\boldsymbol{\mu}) : \mathcal{M} \to \mathbb{R}$ that needs to be maximized, the gradient descent update is $\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t + \rho_t \nabla \mathcal{L}(\boldsymbol{\mu}_t)$ where ρ_t is the step size. The above update is equivalent to maximize a local quadratic approximation of $\mathcal{L}(\boldsymbol{\mu})$: $\boldsymbol{\mu}_{t+1} = \arg\max_{\boldsymbol{\mu} \in \mathcal{M}} \nabla \mathcal{L}(\boldsymbol{\mu}_t)^\top \boldsymbol{\mu} - \frac{1}{2\rho_t} \|\boldsymbol{\mu} - \boldsymbol{\mu}_t\|_2^2$. Mirror descent replaces Euclidean norm with a proximity function $\Psi(\cdot, \cdot) : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+$:

$$\boldsymbol{\mu}_{t+1} = \arg\max_{\boldsymbol{\mu} \in \mathcal{M}} \nabla \mathcal{L}(\boldsymbol{\mu}_t)^\top \boldsymbol{\mu} - \frac{1}{\rho_t} \Psi(\boldsymbol{\mu}, \boldsymbol{\mu}_t).$$

The proximity function characterizes the non-Euclidean geometry and different choices of Ψ correspond to different manifolds that variational parameters lie on.

3. Methodology

The GP-based FSC (Patacchiola et al., 2020; Titsias et al., 2020; Snell & Zemel, 2021) is an emerging topic in recent years, which combines the Bayesian framework with few-shot learning. Given a large number of small but related classification tasks, the main idea of GP-based FSC is to perform GP classification for each task and learn a common GP prior that represents the meta-knowledge. This prior is then used to train a GP classifier on a small support set to predict class labels of a query set for a new unseen task. A bi-level optimization framework, which is similar to model-agnostic meta-learning (MAML) (Finn et al., 2017), is used to learn the common GP prior (Patacchiola et al., 2020; Snell & Zemel, 2021): in the inner loop, VI is performed to estimate the posterior of latent functions for each task (task-specific parameters); in the outer loop, the GP kernel is designed as a deep kernel (Wilson et al., 2016) whose hyperparameters (task-common parameters), including the kernel hyperparameters and neural network weights, are updated by empirical Bayes (Maritz & Lwin, 2018).

In the bi-level optimization framework, both the inner and outer loops consistently employ first-order optimization methods, leading to sluggish convergence in the search for task-common parameters. Moreover, the convergence rate of the inner loop is dependent on the parameterization of the variational distribution, and an arbitrary parameterization may impede faster convergence. In this study, our objective is to enhance the convergence rate of the inner loop. Importantly, we do not anticipate the convergence rate of the inner loop to be dependent on the parameterization of the variational distribution since in practice we often lack knowledge about which parameterization is superior. Consequently, the accelerated convergence of the inner loop also facilitates improvements in the convergence of the outer loop.

3.1. From Natural Gradient Descent to Mirror Descent

To address the above issues, an intuitive approach is to optimize the ELBO with natural gradient descent (Amari, 1997) which is a second-order optimization method and possesses the parameterization invariance property in the meanwhile. A large number of existing work applied natural gradient descent to the variational inference of GP (Hensman et al., 2013; Malagò & Pistone, 2015; Salimbeni et al., 2018). However, a serious disadvantage hindering the application of natural gradient descent is the complex computation due to the requirement of second information (Fisher information matrix). To facilitate the implementation of natural gradient descent, we propose to use mirror descent in place of natural gradient descent. The following theorem establishes the equivalence between natural gradient descent and mirror descent. A detailed proof is provided in Appendix B.

Theorem 3.1 (Raskutti & Mukherjee 2015)**.** *Given two parameterized ELBO with mean parameter and natural parameter $\widetilde{\mathcal{L}}(\boldsymbol{\mu}) = \mathcal{L}(\boldsymbol{\theta})$, to maximize the ELBO, the mirror descent over the mean parameter*

$$\boldsymbol{\mu}_{t+1} = \arg\max_{\boldsymbol{\mu} \in \mathcal{M}} \nabla \widetilde{\mathcal{L}}(\boldsymbol{\mu}_t)^\top \boldsymbol{\mu} - \frac{1}{\rho_t} B_H(\boldsymbol{\mu}, \boldsymbol{\mu}_t), \tag{3}$$

using the Bregman divergence (Bregman, 1967) $B_H(\boldsymbol{\mu}, \boldsymbol{\mu}_t) = H(\boldsymbol{\mu}) - H(\boldsymbol{\mu}_t) - \nabla H(\boldsymbol{\mu}_t)^\top (\boldsymbol{\mu} - \boldsymbol{\mu}_t)$ induced by the negative entropy function $H(\cdot)$ is equivalent

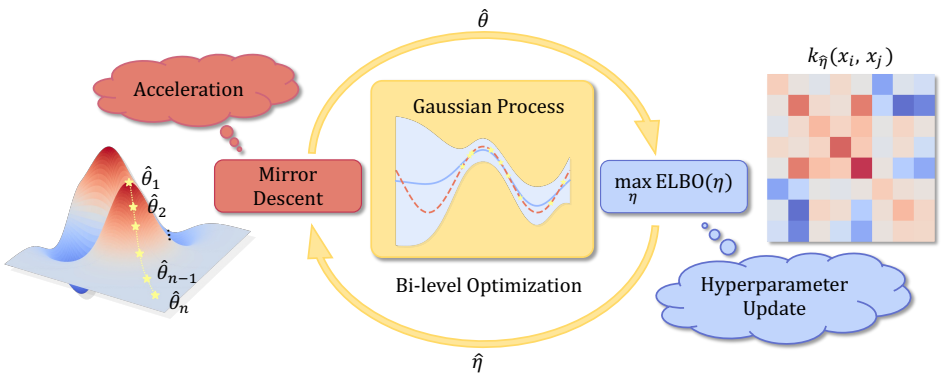


Figure 1. The overview of the training process of MD-BSFC. The diagram illustrates the bi-level optimization process involving an iterative application of mirror descent for VI and hyperparameter tuning.

to the natural gradient descent over the natural parameter

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \rho_t [\nabla^2_{\boldsymbol{\theta}} A(\boldsymbol{\theta}_t)]^{-1} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_t),$$

using the Riemannian metric induced by the log-partition function $A(\cdot)$.

The theorem above asserts that employing natural gradient descent over the natural parameter using Fisher information, induced by the log-partition function, is equivalent to implementing mirror descent over the mean parameter using Bregman divergence induced by the negative entropy function. The notable advantage of mirror descent over natural gradient descent lies in its reliance on only the first-order gradient, as opposed to the second-order gradient required by the latter. Consequently, the implementation of natural gradient descent can be reformulated as a computationally more efficient mirror descent.

3.2. From Mirror Descent to Conjugate Bayesian Inference

As demonstrated by Khan & Lin (2017), Equation (3) can be additionally streamlined into a Bayesian inference within a conjugate model. The subsequent theorem establishes the equivalence between Equation (3) and Bayesian inference within a conjugate model. We refer to Khan & Lin (2017) for the complete proof.

Theorem 3.2 (Khan & Lin 2017)**.** *The mirror descent in Equation (3) is equivalent to the following conjugate Bayesian inference:*

$$q(\mathbf{f}|\boldsymbol{\theta}_{t+1}) \propto \exp\left(\widetilde{\boldsymbol{\theta}}_t^\top \phi(\mathbf{f})\right) p(\mathbf{f}|\boldsymbol{\eta}),$$
$$\widetilde{\boldsymbol{\theta}}_t = (1-\rho_t)\widetilde{\boldsymbol{\theta}}_{t-1} + \rho_t \nabla_{\boldsymbol{\mu}} \mathbb{E}_q[\log p(\mathbf{y}|\mathbf{f})]|_{\boldsymbol{\mu}=\boldsymbol{\mu}_t}, \quad (4)$$

with $\widetilde{\boldsymbol{\theta}}_0 = \mathbf{0}$ *and* $\boldsymbol{\theta}_1 = \boldsymbol{\eta}$*. Because* $p(\mathbf{f}|\boldsymbol{\eta})$ *is the Gaussian prior in Equation (1) that is also a exponential-family distribution,* $\boldsymbol{\theta}_{t+1}$ *can be obtained by conjugate computation* $\boldsymbol{\theta}_{t+1} = \widetilde{\boldsymbol{\theta}}_t + \boldsymbol{\eta}$.

The aforementioned theorem asserts that the original softmax likelihood in Equation (1) is approximated by a Gaussian distribution, with its natural parameter iteratively updated by the gradient of the expectation of the log-likelihood. Subsequently, the variational parameter is updated by adding the natural parameters of the approximated Gaussian distribution and the prior.

The motivation for using mirror descent should now be clear: for the current problem, mirror descent over the mean parameter is an equivalent approach to natural gradient descent over the natural parameter. Therefore, it also exhibits a second-order convergence rate and possesses the parameterization invariance property. Moreover, mirror descent streamlines the optimization process by requiring only first-order information, and the optimization step can be simplified as a conjugate Bayesian inference computation.

3.3. Algorithm

The bi-level optimization framework of MD-BFSC is depicted in Figure 1 and summarized as follows. In the inner loop, the variational parameters $\boldsymbol{\theta}$ (task-specific parameters) are updated by using Theorem 3.2 and Equation (4). For $\nabla_{\boldsymbol{\mu}} \mathbb{E}_q[\log p(\mathbf{y}|\mathbf{f})]$ in Equation (4), we assume the variaitonal distribution $q(\mathbf{f}|\boldsymbol{\theta}) = \prod_c q(\mathbf{f}^c|\boldsymbol{\theta}^c)$. Since $p(\mathbf{y}|\mathbf{f}) = \prod_n p(\mathbf{y}_n|\mathbf{f}_n)$, we can compute the gradient of each point $\nabla_{\boldsymbol{\mu}_n} \mathbb{E}_{q(\mathbf{f}_n)}[\log p(\mathbf{y}_n|\mathbf{f}_n)]$ separately. If we define $\mathbf{m}_n$ and $\mathbf{v}_n$ to be the mean and covariance diagonal (non-diagonal entries are 0) of the marginal distribution $q(\mathbf{f}_n)$, then $\nabla_{\boldsymbol{\mu}_n} \mathbb{E}_{q(\mathbf{f}_n)}[\log p(\mathbf{y}_n|\mathbf{f}_n)]$ can be expressed as:

$$\nabla_{\boldsymbol{\mu}_n^{(1)}} \mathbb{E}_{q(\mathbf{f}_n)}[\log p(\mathbf{y}_n|\mathbf{f}_n)] = \mathbf{g}_{\mathbf{m}_n} - 2\mathbf{g}_{\mathbf{v}_n} \odot \mathbf{m}_n,$$
$$\nabla_{\boldsymbol{\mu}_n^{(2)}} \mathbb{E}_{q(\mathbf{f}_n)}[\log p(\mathbf{y}_n|\mathbf{f}_n)] = \mathbf{g}_{\mathbf{v}_n},$$

where $\boldsymbol{\mu}_n^{(1)}$ and $\boldsymbol{\mu}_n^{(2)}$ are the two mean parameters of $q(\mathbf{f}_n)$, $\mathbf{g}_{\mathbf{m}_n} = \nabla_{\mathbf{m}_n} \mathbb{E}_q[\log p(\mathbf{y}_n|\mathbf{f}_n)] = \mathbb{E}_q\left[\mathbf{y}_n - \frac{\exp(\mathbf{f}_n)}{\sum_c \exp(\mathbf{f}_n)}\right]$, $\mathbf{g}_{\mathbf{v}_n} = \nabla_{\mathbf{v}_n} \mathbb{E}_q[\log p(\mathbf{y}_n|\mathbf{f}_n)] =$

$\frac{1}{2}\mathbb{E}_q\left[\frac{\exp(2\mathbf{f}_n)}{(\sum_c \exp(\mathbf{f}_n))^2} - \frac{\exp(\mathbf{f}_n)}{\sum_c \exp \mathbf{f}_n}\right]$, $\odot$ is the element-wise product (Opper & Archambeau, 2009). A proof of Section 3.3 is provided in Appendix C.

In the outer loop, the GP kernel is designed as a deep kernel (Wilson et al., 2016) and the kernel hyperparameter $\boldsymbol{\eta}$ includes the traditional kernel hyperparameters and neural network weights. The hyperparameter $\boldsymbol{\eta}$ (task-common parameters) is updated by maximizing the ELBO, that is an approximation of log-marginal likelihood:

$$\arg\max_{\boldsymbol{\eta}} \mathcal{L}(\boldsymbol{\eta}) = \mathbb{E}_q\left[\log p(\mathbf{y}|\mathbf{f}) + \log p(\mathbf{f}|\boldsymbol{\eta}) - \log q(\mathbf{f})\right]. \tag{5}$$

For prediction in a new test task with support set: $\mathcal{S} = \{\mathbf{X}, \mathbf{y}\}$ and query set without labels $\mathcal{Q} = \mathbf{X}_*$, the label predictive distribution of one sample $\mathbf{x}_*$ in the query set is:

$$p(y_* = r \mid \mathbf{x}_*, \mathbf{X}, \mathbf{y}, \hat{\boldsymbol{\eta}}) \tag{6a}$$
$$= \int p(y_* = r \mid \mathbf{f}_*) \prod_{c=1}^{C} q(f_*^c \mid \mathbf{X}, \mathbf{Y}, \hat{\boldsymbol{\eta}}) d\mathbf{f}_*,$$
$$q(f_*^c \mid \mathbf{X}, \mathbf{y}, \hat{\boldsymbol{\eta}}) \tag{6b}$$
$$= \int p(f_*^c \mid \mathbf{f}^c) q(\mathbf{f}^c \mid \mathbf{X}, \mathbf{y}, \hat{\boldsymbol{\eta}}) d\mathbf{f}^c = \mathcal{N}(f_*^c \mid \mu_*^c, \sigma_*^{2^c}),$$

where $\hat{\boldsymbol{\eta}}$ is the estimated kernel hyperparameter from the training dataset, $\mathbf{f}_* = [f^1(\mathbf{x}_*), \ldots, f^C(\mathbf{x}_*)]^\top$, $q(\mathbf{f}^c \mid \mathbf{X}, \mathbf{y}, \hat{\boldsymbol{\eta}})$ is a Gaussian distribution computed by Theorem 3.2 and Equation (4). By converting the natural parameters into traditional parameters, $\mathcal{N}(\mathbf{f}^c|\boldsymbol{\theta}^c) \rightarrow \mathcal{N}(\mathbf{f}^c|\mathbf{m}^c, \boldsymbol{\Sigma}^c)$, we can derive $\mu_*^c = \mathbf{k}_{*L}^c \mathbf{K}_{LL}^{c^{-1}} \mathbf{m}^c$ and $\sigma_*^{2^c} = k_{**}^c - \mathbf{k}_{*L}^c \mathbf{K}_{LL}^{c^{-1}} \mathbf{k}_{L*}^c + \mathbf{k}_{*L}^c \mathbf{K}_{LL}^{c^{-1}} \boldsymbol{\Sigma}^c \mathbf{K}_{LL}^{c^{-1}} \mathbf{k}_{L*}^c$ where L is the number of samples in the support set of the test dataset, and all the $\mathbf{k}$'s are the respective kernel matrices. The integral in Equation (6a) is intractable, necessitating the use of Monte Carlo for computation. The training and test process of MD-BFSC is summarized in Algorithm 1.

Algorithm 1: Mirror Descent based Bayesian Few-Shot Classification

Training:
Input: Input feature and class labels for S tasks: $\{\mathbf{X}_s\}_{s=1}^S$, $\{\mathbf{y}_s\}_{s=1}^S$
Output: GP kernel hyperparameter $\boldsymbol{\eta}$
Initialize GP kernel hyperparameter $\boldsymbol{\eta}$ and variational parameters $\widetilde{\boldsymbol{\theta}}_0 = \mathbf{0}$ and $\boldsymbol{\theta}_1 = \boldsymbol{\eta}$;
for *Iteration* **do**
  for *Task* s **do**
    # Update task-specific parameters
    for *Step* t **do**
      Update $\widetilde{\boldsymbol{\theta}}_t^s$ by Equation (4) and Section 3.3;
      Update $\boldsymbol{\theta}_{t+1}^s = \widetilde{\boldsymbol{\theta}}_t^s + \boldsymbol{\eta}$;
    end
    # Update task-common parameters
    Update $\boldsymbol{\eta}$ by Equation (5).
  end
end
Test:
Input: Support set $\mathcal{S} = \{\mathbf{X}, \mathbf{y}\}$; query set $\mathcal{Q} = \mathbf{X}_*$; learned hyperparameter $\hat{\boldsymbol{\eta}}$
Output: Predicted labels
Initialize variational parameters $\widetilde{\boldsymbol{\theta}}_0 = \mathbf{0}$ and $\boldsymbol{\theta}_1 = \hat{\boldsymbol{\eta}}$;
# Update task-specific parameters
for *Step* t **do**
  Update $\widetilde{\boldsymbol{\theta}}_t$ by Equation (4) and Section 3.3;
  Update $\boldsymbol{\theta}_{t+1} = \widetilde{\boldsymbol{\theta}}_t + \hat{\boldsymbol{\eta}}$;
end
# Predict labels
for $\mathbf{x}_* \in \mathbf{X}_*$ **do**
  Predict y_* by Equation (6).
end

4. Experiments

In this section, we present the performance of MD-BFSC on few-shot classification tasks, encompassing accuracy, uncertainty quantification, and convergence rate. We address three challenging tasks using benchmark datasets, including Caltech-UCSD Birds (Wah et al., 2011), mini-ImageNet (Ravi & Larochelle, 2017), Omniglot (Lake et al., 2011), and EMNIST (Cohen et al., 2017).

4.1. Accuracy

In this section, we present the experimental arrangement and disclose the findings of our few-shot classification tasks. While we acknowledge the existence of various configurations for few-shot classification, we opt for the basic setup in Bayesian meta-learning for a fair comparison. Consistent with the methodology employed in previous studies (Patacchiola et al., 2020), we utilize a standard Conv4 architecture (Vinyals et al., 2016) as the underlying backbone, and evaluate our model across six distinct scenarios, encompassing 1-shot and 5-shot situations for both in-domain and cross-domain tasks. We subject our model to a comprehensive assessment against a range of baselines and state-of-the-art models, including Feature Transfer (Chen et al., 2019), Baseline++ (Chen et al., 2019), MatchingNet (Vinyals et al., 2016), ProtoNet (Snell et al., 2017a), RelationNet (Sung et al., 2018), MAML (Finn et al., 2017), DKT (Patacchiola et al., 2020), Bayesian MAML (Yoon et al., 2018), ABML (Ravi & Beatson, 2019), OVE (Snell & Zemel, 2021), and LS (Ke et al., 2023). It is worth noting that DKT, LS, and OVE share similarities with our proposed approach as they are all GP-based models with different likelihood functions

Table 1. Accuracy performance for all models in 1-shot and 5-shot 5-way few-shot classification tasks. Baseline results are from Snell & Zemel (2021) and Ke et al. (2023). Results are evaluated over 5 batches of 600 episodes with different random seeds.

	CUB		mini-ImageNet → CUB		Omniglot → EMNIST	
Method	**1-shot**	**5-shot**	**1-shot**	**5-shot**	**1-shot**	**5-shot**
Feature Transfer	46.19 ± 0.64	68.40 ± 0.79	32.77 ± 0.35	50.34 ± 0.27	64.22 ± 1.24	86.10 ± 0.84
Baseline++	61.75 ± 0.95	78.51 ± 0.59	39.19 ± 0.12	$\mathbf{57.31 \pm 0.11}$	56.84 ± 0.91	80.01 ± 0.92
MatchingNet	60.19 ± 1.02	75.11 ± 0.35	36.98 ± 0.06	50.72 ± 0.36	75.01 ± 2.09	87.41 ± 1.79
ProtoNet	52.52 ± 1.90	75.93 ± 0.46	33.27 ± 1.09	52.16 ± 0.17	72.04 ± 0.82	87.22 ± 1.01
RelationNet	62.52 ± 0.34	78.22 ± 0.07	37.13 ± 0.20	51.76 ± 1.48	75.62 ± 1.00	87.84 ± 0.27
MAML	56.11 ± 0.69	74.84 ± 0.62	34.01 ± 1.25	48.83 ± 0.62	72.68 ± 1.85	83.54 ± 1.79
DKT	63.37 ± 0.19	77.73 ± 0.26	40.22 ± 0.54	55.65 ± 0.05	73.06 ± 2.36	$\mathbf{88.10 \pm 0.78}$
Bayesian MAML	55.93 ± 0.71	72.87 ± 0.26	33.52 ± 0.36	51.35 ± 0.16	63.94 ± 0.47	65.26 ± 0.30
Bayesian MAML (Chaser)	53.93 ± 0.72	71.16 ± 0.32	36.22 ± 0.50	51.53 ± 0.43	55.04 ± 0.34	54.19 ± 0.32
ABML	49.57 ± 0.42	68.94 ± 0.16	29.35 ± 0.26	45.74 ± 0.33	73.89 ± 0.24	87.28 ± 0.40
OVE PG GP (ML)	63.98 ± 0.43	77.44 ± 0.18	39.66 ± 0.18	55.71 ± 0.31	68.43 ± 0.67	86.22 ± 0.20
OVE PG GP (PL)	60.11 ± 0.26	79.07 ± 0.05	37.49 ± 0.11	57.23 ± 0.31	$\mathbf{77.00 \pm 0.50}$	87.52 ± 0.19
CDKT (ML) ($\tau < 1$)	65.21 ± 0.45	$\mathbf{79.10 \pm 0.33}$	40.43 ± 0.43	55.72 ± 0.45	-	-
CDKT (ML) ($\tau = 1$)	60.85 ± 0.38	75.98 ± 0.33	35.57 ± 0.30	52.42 ± 0.50	-	-
CDKT (PL) ($\tau < 1$)	59.49 ± 0.35	76.95 ± 0.28	39.18 ± 0.34	56.18 ± 0.28	-	-
CDKT (PL) ($\tau = 1$)	52.91 ± 0.29	73.34 ± 0.40	37.62 ± 0.32	54.32 ± 0.19	-	-
MD-BFSC	$\mathbf{65.45 \pm 0.42}$	78.38 ± 0.21	$\mathbf{40.75 \pm 0.31}$	56.98 ± 0.30	74.02 ± 0.49	87.05 ± 0.23

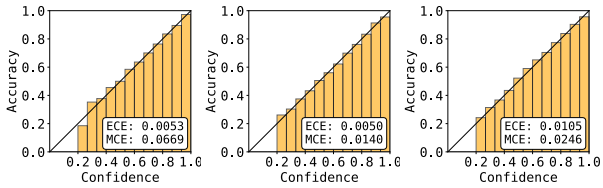
and inference methods. The default number of epochs from Ke et al. (2023) is employed for training MD-BSFC. During training, we run 3 steps for the inner loop updates in all experiments due to its fast convergence and then conduct a 1 step update for the hyperparameters with Adam. For further experimental details, please refer to Appendix E.

We report the average accuracy and standard deviation of our model evaluated on 5 batches of 600 episodes with different random seeds in Table 1. Our model achieves the highest accuracy in 1-shot experiments on the CUB dataset (65.45%, in-domain) and the CUB → mini-ImageNet dataset (40.75%, cross-domain). As for other datasets and settings, our model also achieves near-optimal or comparable results. While the main strength of our method is its theoretically fast convergence, we find that its classification accuracy is comparable to state-of-the-art models, demonstrating its utility beyond rapid learning.

4.2. Uncertainty Quantification

In the realm of FSC, the quantification of uncertainty holds great significance due to the potential high uncertainty in predictions made by classifiers trained with limited data. Particularly in high-risk domains, it is crucial for a robust model to effectively characterize such uncertainty.

To quantify uncertainty, we employ two widely-used metrics: the expected calibration error (ECE) (Guo et al., 2017) and the maximum calibration error (MCE) (Ovadia et al., 2019). ECE measures the average difference between confidence (probability outputs) and accuracy within predefined bins, while MCE is similar but focuses on the maximum difference. Following the evaluation protocol outlined by Patacchiola et al. (2020), we compute the ECE and MCE on the test set. The summarized results of the 5-shot experiments can be found in Table 2. Specifically, we achieve the lowest ECE values on CUB (0.005, in-domain) and mini-ImageNet → CUB (0.005, cross-domain), as well as the lowest MCE values on mini-ImageNet → CUB (0.014, cross-domain) and Omniglot → EMNIST (0.024, cross-domain). While our model slightly lags behind the state-of-the-art models in other scenarios, the overall outcome highlights its remarkable reliability in uncertainty calibration, thereby demonstrating promising robustness in few-shot scenarios.



(a) CUB (b) MI → CUB (c) Omni → EMNIST

Figure 2. Reliability diagrams on 5-shot classification with ECE and MCE metrics. MI denotes mini-ImageNet and Omni denotes Omniglot. Results are computed on 3,000 test tasks.

The reliability diagrams displayed in Figure 2 illustrate the performance of our model in terms of reliability. A robust uncertainty quantification model should exhibit a confidence barplot that aligns with the diagonal line. Our model closely adheres to the diagonal line.

Table 2. ECE and MCE performance for all models in 5-shot 5-way tasks on CUB, mini-ImageNet → CUB, and Omniglot → EMNIST. Baseline results are from Snell & Zemel (2021) and Ke et al. (2023). All metrics are computed on 3,000 random tasks from the test set.

	CUB		mini-ImageNet → CUB		Omniglot → EMNIST	
Method	**ECE**	**MCE**	**ECE**	**MCE**	**ECE**	**MCE**
Feature Transfer	0.187	0.250	0.275	0.646	**0.004**	0.092
Baseline++	0.421	0.502	0.315	0.537	0.390	0.475
MatchingNet	0.023	0.031	0.030	0.079	0.057	0.259
ProtoNet	0.034	0.059	0.009	0.025	0.010	0.243
RelationNet	0.438	0.593	0.234	0.554	0.552	0.594
DKT	0.187	0.250	0.236	0.426	0.413	0.483
Bayesian MAML	0.018	0.047	0.048	0.077	0.090	0.124
Bayesian MAML (Chaser)	0.047	0.104	0.066	0.260	0.235	0.306
OVE PG GP (ML)	0.026	0.043	0.049	0.066	0.112	0.183
OVE PG GP (PL)	**0.005**	**0.023**	0.020	0.032	0.064	0.240
CDKT (ML)	**0.005**	0.036	0.007	0.020	-	-
CDKT (PL)	0.018	0.223	0.010	0.029	-	-
MD-BFSC	**0.005**	0.067	**0.005**	**0.014**	0.010	**0.025**

4.3. Convergence Rate

In this section, we would like to investigate the empirical convergence rate of our proposed method. Therefore, we compare MD-BSFC to the method that employs the exact same variational distribution, but its inner-loop VI is implemented through vanilla gradient descent. We disentangle the bi-level optimization into two parts: fixing the hyperparameters of the kernel and neural network to check inner-loop convergence and checking the outer-loop convergence with full bi-level optimization.

Inner-loop convergence For simplicity, we fix a random batch of data from a random episode for the CUB dataset and the mini-ImageNet dataset and observe the learning curve for ELBO. To ensure a fair comparison, we employ the same initialization scheme for the variational parameters and a comparable learning rate of 0.005 for the inner loop and run each method for 30 steps. Notice that in this empirical study, we employ a relatively small learning rate to better observe the alteration of the ELBO curve. The results can be seen in Figure 3. The empirical observation aligns with our theoretical analysis: the ELBO of MD-BSFC increases at a faster rate than the vanilla method for both 1-shot and 5-shot scenarios. This is attributed to the fact that MD-BSFC employs second-order optimization in the inner loop, whereas the vanilla method utilizes first-order optimization.

Outer-loop convergence For the full bi-level convergence experiment, we use a larger learning rate (0.1 for 1-shot and 0.02 for 5-shot) with 2 steps for the inner loop. The outer loop is updated by Adam with a learning rate of 0.001. We empirically find that gradient descent displays high numerical instability for ELBO when a larger learning rate is applied. Thus, we use predictive likelihood for the outer-loop loss (Ke et al., 2023), which is a plausible alternative to reflect the performance of convergence. Specifically, we perform inner-loop updates on the support samples and compute the cross-entropy loss on the validation samples. We run each method for 30 iterations for the outer-loop updates as well and plot the cross-entropy loss. The results can be seen in Figure 4. We observe that MD-BSFC takes a great lead in the 5-shot classification scenario and learns at a slightly higher pace in the 1-shot classification, demonstrating an improvement in convergence speed.

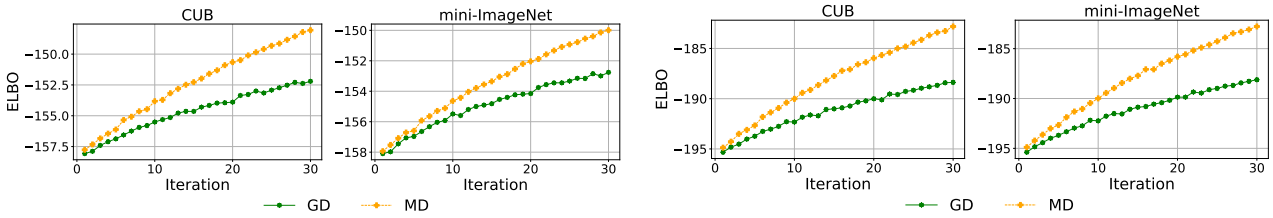
Table 3. Accuracy of MD-BSFC in 1-shot 5-way tasks on the CUB dataset with different hyperparameters. Results are evaluated over 5 batches of 600 episodes with different random seeds.

ρ	**MD-BSFC**	**kernel**	**MD-BSFC**	**step**	**MD-BSFC**
0.01	32.22 ± 0.41	COS	$\mathbf{65.45 \pm 0.42}$	1	62.63 ± 0.26
0.1	61.54 ± 0.26	RBF	62.75 ± 0.39	3	$\mathbf{65.45 \pm 0.42}$
0.5	63.86 ± 0.38	POL1	64.69 ± 0.36	5	64.18 ± 0.52
1	$\mathbf{65.45 \pm 0.42}$	POL2	64.74 ± 0.25	7	62.55 ± 0.46

4.4. Hyperparameter Settings

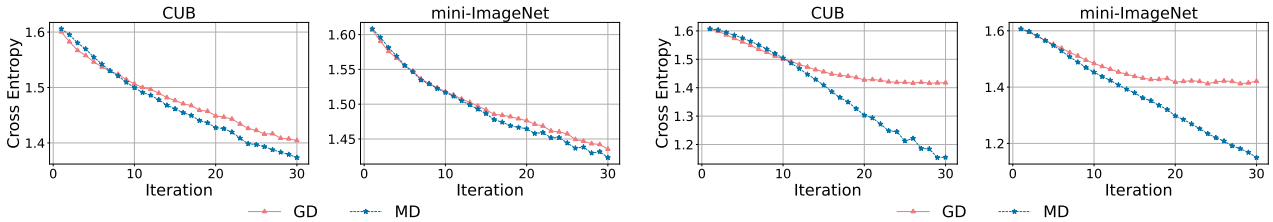
In this section, we investigate the robustness of our method in different hyperparameter settings, including various inner-loop step size ρ, various kernels, and different number of steps for the inner loop. For simplicity, we present the results on the CUB dataset for the 1 shot scenario.

Inner-loop step size Notice that in Equation (4), the inner-loop step size ρ controls the update rate of a single natural gradient step, where a larger ρ may result in faster convergence but greater variance in general. However, in a bi-level optimization setting, the potential impact of varying step sizes combined with the highly nonconvex loss landscape of neural networks is not clear as opposed to single-level optimization. Therefore, we present some empirical results



(a) 1-shot classification (b) 5-shot classification

Figure 3. The ELBO curve of 30 iterations on the CUB dataset and the mini-ImageNet dataset for 1-shot and 5-shot classification. Our method with mirror descent (MD) learns at a faster rate than the vanilla method with gradient descent (GD) in both scenarios.



(a) 1-shot classification (b) 5-shot classification

Figure 4. The loss curve of 30 iterations on the CUB dataset and the mini-ImageNet dataset for 1-shot and 5-shot classification. Our method with mirror descent (MD) learns at a faster rate than the vanilla method with gradient descent (GD) in both scenarios.

to shed some light on this issue in Table 3. Empirically, a larger ρ leads to better performance, while for an extremely small ρ the model tends to collapse. We suppose the phenomenon is due to a larger step size assist in escaping the saddle point of the highly nonconvex loss function.

Base kernel Patacchiola et al. (2020); Snell & Zemel (2021); Ke et al. (2023) illustrated the performance of various base kernels being close and the tendency of performance decay for smoother kernels, e.g., RBF kernel. We follow their protocol and test the results on similar candidate base kernels in Table 3, including cosine kernel with normalization (COS), RBF kernel, and polynomial kernel of order 1 (POL1) and 2 (POL2) (Patacchiola et al., 2020). More details about the kernels are provided in Appendix E.4. Our empirical result conforms with the previous insights, where finite-rank kernels (POL1, POL2, COS) generally obtain a superior performance on the CUB dataset.

Inner-loop steps We explore the inherent repercussions of varying inner-loop update steps. For single-level optimization where the outer-loop hyperparameters are fixed, the inner-loop parameters should be optimized until convergence. However, due to the complex training dynamics caused by bi-level optimization, it is not clear how we should select the inner-loop update steps. Snell et al. (2017b) employed a 1-step Gibbs sampling for training while Ke et al. (2023) found a step number of 2 is optimal for various datasets for their method. Here we display some empirical results on this hyperparameter in Table 3. Similar to Ke et al. (2023), we also find that a relatively small step number is optimal for our method. We suppose the reason behind this phenomenon is that only 1 step of update proffers an inferior approximation of the optimal task-specific parameters, while larger step numbers may affect the gradient flow.

5. Limitation

In this work, we focus on upgrading inner-loop optimization from a first-order to a second-order method, leading to enhanced overall convergence. However, the updates of the neural network (outer-loop) rely on a first-order method due to computational and memory constraints. Future work will explore second-order methods in outer-loop optimization.

6. Conclusion

In conclusion, our approach has contributed to Bayesian few-shot classification by addressing the non-conjugate inference challenge in GP classification. Moreover, the integration of non-Euclidean geometry through mirror descent has proven effective in enhancing the convergence rate theoretically and empirically. Experimental results robustly underscore the effectiveness of our proposed method, achieving competitive classification accuracy and uncertainty quantification on several benchmark datasets compared to baseline models. Additionally, our systematic exploration of various hyperparameters and components within the model provides valuable insights into their impact on performance.

Acknowledgments

This work was supported by NSFC Project (No. 62106121), the MOE Project of Key Research Institute of Humanities and Social Sciences (22JJD110001), and the Public Computing Cloud, Renmin University of China.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Amari, S. Neural learning in structured parameter spaces-natural riemannian gradient. *Advances in neural information processing systems*, pp. 127–133, 1997.

Bishop, C. M. *Pattern Recognition and Machine Learning*. springer, 2006.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

Bregman, L. M. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3): 200–217, 1967.

Chen, W., Liu, Y., Kira, Z., Wang, Y. F., and Huang, J. A closer look at few-shot classification. In *International Conference on Learning Representations*. OpenReview.net, 2019.

Cohen, G., Afshar, S., Tapson, J., and Van Schaik, A. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pp. 2921–2926. IEEE, 2017.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135. PMLR, 2017.

Finn, C., Xu, K., and Levine, S. Probabilistic model-agnostic meta-learning. *arXiv preprint arXiv:1806.02817*, 2018.

Galy-Fajou, T., Wenzel, F., Donner, C., and Opper, M. Multi-class gaussian process classification made conjugate: Efficient inference via data augmentation. In *Uncertainty in Artificial Intelligence*, pp. 755–765. PMLR, 2020.

Grant, E., Finn, C., Levine, S., Darrell, T., and Griffiths, T. Recasting gradient-based meta-learning as hierarchical bayes. *arXiv preprint arXiv:1801.08930*, 2018.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017.

Hensman, J., Fusi, N., and Lawrence, N. D. Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, UAI 2013, Bellevue, WA, USA, August 11-15, 2013*, 2013.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. Stochastic variational inference. *Journal of Machine Learning Research*, 14(5), 2013.

Ke, T., Cao, H., Ling, Z., and Zhou, F. Revisiting logistic-softmax likelihood in bayesian meta-learning for few-shot classification. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Khan, M. and Lin, W. Conjugate-computation variational inference: Converting variational inference in non-conjugate models to inferences in conjugate models. In *Artificial Intelligence and Statistics*, pp. 878–887. PMLR, 2017.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Lake, B., Salakhutdinov, R., Gross, J., and Tenenbaum, J. One shot learning of simple visual concepts. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011.

Malagò, L. and Pistone, G. Information geometry of the gaussian distribution in view of stochastic optimization. In *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII*, pp. 150–162, 2015.

Maritz, J. S. and Lwin, T. *Empirical bayes methods*. Chapman and Hall/CRC, 2018.

Martens, J. New insights and perspectives on the natural gradient method. *J. Mach. Learn. Res.*, 21:146:1–146:76, 2020.

Miller, E. G., Matsakis, N. E., and Viola, P. A. Learning from one example through shared densities on transforms. In *Proceedings IEEE Conference on Computer Vision and*

Pattern Recognition. CVPR 2000 (Cat. No. PR00662), volume 1, pp. 464–471. IEEE, 2000.

Nemirovskij, A. S. and Yudin, D. B. Problem complexity and method efficiency in optimization. *SIAM Review*, 27 (2):264–265, 1985.

Opper, M. and Archambeau, C. The variational gaussian approximation revisited. *Neural computation*, 21(3):786–792, 2009.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.

Patacchiola, M., Turner, J., Crowley, E. J., O'Boyle, M. F. P., and Storkey, A. J. Bayesian meta-learning for the few-shot setting via deep kernels. In *Advances in Neural Information Processing Systems*, 2020.

Polson, N. G., Scott, J. G., and Windle, J. Bayesian inference for logistic models using Pólya-Gamma latent variables. *Journal of the American statistical Association*, 108(504):1339–1349, 2013.

Prabhu, V., Kannan, A., Ravuri, M., Chaplain, M., Sontag, D., and Amatriain, X. Few-shot learning for dermatological disease diagnosis. In *Machine Learning for Healthcare Conference*, pp. 532–552. PMLR, 2019.

Raskutti, G. and Mukherjee, S. The information geometry of mirror descent. *IEEE Transactions on Information Theory*, 61(3):1451–1457, 2015.

Ravi, S. and Beatson, A. Amortized bayesian meta-learning. In *International Conference on Learning Representations*, 2018.

Ravi, S. and Beatson, A. Amortized bayesian meta-learning. In *International Conference on Learning Representations*, 2019.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. In *International conference on learning representations*, 2017.

Rissanen, J. J. Fisher information and stochastic complexity. *IEEE transactions on information theory*, 42(1):40–47, 1996.

Rockafellar, R. T. *Convex analysis*. Princeton university press, 2015.

Salimans, T. and Knowles, D. A. Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 8(4):837–882, 2013.

Salimbeni, H., Eleftheriadis, S., and Hensman, J. Natural gradients in practice: Non-conjugate variational inference in gaussian process models. In *International Conference on Artificial Intelligence and Statistics*, pp. 689–697. PMLR, 2018.

Snell, J. and Zemel, R. S. Bayesian few-shot classification with one-vs-each pólya-gamma augmented gaussian processes. In *International Conference on Learning Representations*. OpenReview.net, 2021.

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017a.

Snell, J., Swersky, K., and Zemel, R. S. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 4077–4087, 2017b.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., and Hospedales, T. M. 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.

Titsias, M. K. One-vs-each approximation to softmax for scalable estimation of probabilities. In *Advances in Neural Information Processing Systems*, pp. 4161–4169, 2016.

Titsias, M. K., Nikoloutsopoulos, S., and Galashov, A. Information theoretic meta learning with gaussian processes. *arXiv preprint arXiv:2009.03228*, 2020.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The caltech-ucsd birds-200-2011 dataset. 2011.

Wainwright, M. J. and Jordan, M. I. *Graphical models, exponential families, and variational inference*. Now Publishers Inc, 2008.

Wang, Y., Yao, Q., Kwok, J. T., and Ni, L. M. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 53(3):1–34, 2020.

Williams, C. K. and Rasmussen, C. E. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P. Deep kernel learning. In *Artificial intelligence and statistics*, pp. 370–378. PMLR, 2016.

Yoon, J., Kim, T., Dia, O., Kim, S., Bengio, Y., and Ahn, S. Bayesian model-agnostic meta-learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 7343–7353, 2018.

A. Conjugate Variational Inference for Multi-class GP Classification

Consider a multi-class classification task consisting of N observations with the input features $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^\top$ and the corresponding class labels $\mathbf{y} = [\mathbf{y}_1^\top, \dots, \mathbf{y}_N^\top]^\top$, where $\mathbf{x}_i$ is a D-dimensional vector $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^D$ and $\mathbf{y}_i$ is the one-hot encoding for C classes. The multi-class GP classification model includes latent GP functions for each class, i.e., $f^1, \dots, f^C$ where $f^c(\cdot) : \mathcal{X} \to \mathbb{R}$ is the corresponding latent function for c-th class. In order to address the multi-class classification with GPs, a GP prior is assumed for each latent function (Williams & Rasmussen, 2006), i.e., $f^c \sim \mathcal{GP}(0, k_{\boldsymbol{\eta}_c})$ where $k_{\boldsymbol{\eta}_c}$ is the GP kernel for c-th class with hyperparameters $\boldsymbol{\eta}_c$. In practice, the hyperparameters are different for each class.

Following tradition, we utilize the softmax link function to model the data likelihood (categorical distribution):

$$p(\mathbf{y}_n|\mathbf{f}_n) = \frac{\prod_c \exp\left(f_n^c \cdot y_n^c\right)}{\sum_c \exp\left(f_n^c\right)}, \tag{7}$$

where $f_n^c = f^c(\mathbf{x}_n)$ and $\mathbf{f}_n = [f_n^1, \dots, f_n^C]^\top$. In order to predict the class label of a new data point $\mathbf{x}^*$, we need to compute the posterior of latent functions using Bayesian framework:

$$p(\mathbf{f}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f})}{p(\mathbf{y})} = \frac{\prod_n p(\mathbf{y}_n|\mathbf{f}_n) \prod_c p(\mathbf{f}^c)}{\int \prod_n p(\mathbf{y}_n|\mathbf{f}_n) \prod_c p(\mathbf{f}^c)\, \mathrm{d}\mathbf{f}}, \tag{8}$$

where $\mathbf{f}^c = [f_1^c, \dots, f_N^c]^\top$, $\mathbf{f} = [\mathbf{f}^{1\top}, \dots, \mathbf{f}^{C\top}]^\top$, $p(\mathbf{f}^c) = \mathcal{N}(\mathbf{f}^c|\mathbf{0}, \mathbf{K}^c)$ with $K_{ij}^c = k_{\boldsymbol{\eta}^c}(\mathbf{x}_i, \mathbf{x}_j)$.

However, an issue is that the non-Gaussian likelihood in Equation (7) is non-conjugate to the Gaussian prior making the exact computation of posterior infeasible, so some approximate inference methods have to be utilized to approximate the posterior in Equation (8). Variational inference (VI) (Blei et al., 2017) is one of the most popular methods for approximate inference. VI can transform the inference problem into an optimization one which can be addressed by optimization techniques. Specifically, in VI, Equation (8) is approximated by a Gaussian distribution q:

$$q(\mathbf{f}|\boldsymbol{\theta}) = \prod_c \mathcal{N}(\mathbf{f}^c|\boldsymbol{\theta}^c),$$

where $\boldsymbol{\theta} = \{\boldsymbol{\theta}^c\}_{c=1}^C$ is the natural parameter for each class. This posterior approximation assumes independence among different latent functions of the model. The optimal Gaussian distribution is obtained by minimising the Kullback-Leibler (KL) divergence between q and the exact posterior or equivalently maximizing the evidence lower bound (ELBO) (Bishop, 2006):

$$\begin{aligned} \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{\theta}) &\coloneqq \mathbb{E}_q \left[\log p(\mathbf{y}|\mathbf{f})p(\mathbf{f}) - \log q(\mathbf{f})\right] \\ &= \sum_n \mathbb{E}_q[\log p(\mathbf{y}_n|\mathbf{f}_n)] - \sum_c \mathrm{KL}[\mathcal{N}(\mathbf{f}^c|\boldsymbol{\theta}^c) \| \mathcal{N}(\mathbf{f}^c|\mathbf{0}, \mathbf{K}^c)], \end{aligned} \tag{9}$$

where $\boldsymbol{\Theta}$ is the set of valid variational parameters and KL is the KL divergence.

One problem that arises when computing the ELBO in Equation (9) is that the expectation of log-likelihood, i.e., $\mathbb{E}_q[\log p(\mathbf{y}_n|\mathbf{f}_n)]$, does not have a closed-form expression. Despite its intractability, the ELBO can still be optimized by Monte Carlo combined with the reparametrization trick (Kingma & Welling, 2014), but the major issues are: **(1)** A naive implementation of numerical optimization incurs low efficiency due to the complicated and time-consuming non-conjugate computation. **(2)** The rate of convergence depends on the parameterization of the variational distribution. The ordinary gradient in the Euclidean space is an unnatural direction to follow for VI because the parameters correspond to distributions and our goal is to optimize a distribution rather than its parameters (Hoffman et al., 2013; Salimbeni et al., 2018).

Inspired by Khan & Lin (2017), to address these two issues mentioned above, we propose an algorithm to optimize the ELBO with mirror descent developed by Nemirovskij & Yudin (1985) that combines the best of both worlds: on the one hand, each gradient step in our method can be implemented in a conjugate way by approximating the non-conjugate term by a Gaussian distribution; on the other hand, our approach can maximize the ELBO more efficiently by exploiting the non-Euclidean geometry as the mirror descent is the steepest descent direction along the corresponding Riemannian manifold and invariant to the parameterization of the variational distribution (Raskutti & Mukherjee, 2015).

B. Derivation of Theorem 3.1

Consider a exponential-family distribution $p(\mathbf{f}|\boldsymbol{\theta}) = h(\mathbf{f}) \exp(\boldsymbol{\theta}^\top \phi(\mathbf{f}) - A(\boldsymbol{\theta}))$ where $\boldsymbol{\theta}$ is the natural parameter, $\phi(\mathbf{f})$ is the sufficient statistics, $A(\boldsymbol{\theta})$ is the log-partition function $A(\boldsymbol{\theta}) = \log \int h(\mathbf{f}) \exp(\boldsymbol{\theta}^\top \phi(\mathbf{f})) d\mathbf{f}$ that is convex. A minimal

exponential family distribution can also be parameterized by the mean parameter $\boldsymbol{\mu}$ which is the mean of the sufficient statistics $\boldsymbol{\mu} = \mathbb{E}_p[\phi(\mathbf{f})]$. Considering the negative entropy of the distribution as a function of the mean parameter $H(\boldsymbol{\mu}) = \mathbb{E}_p[\log p(\mathbf{f}|\boldsymbol{\theta})] = \boldsymbol{\theta}^\top \boldsymbol{\mu} - A(\boldsymbol{\theta})$ where we omit the constant term, it is a well known result that the log partition function and negative entropy function are convex conjugate functions: $H(\boldsymbol{\mu}) = \sup_{\boldsymbol{\theta}'}[\boldsymbol{\theta}'^\top \boldsymbol{\mu} - A(\boldsymbol{\theta}')]$ and $A(\boldsymbol{\theta}) = \sup_{\boldsymbol{\mu}'}[\boldsymbol{\theta}^\top \boldsymbol{\mu}' - H(\boldsymbol{\mu}')]$. Further, if $A(\cdot)$ is strictly convex and twice differentiable, then so is $H(\cdot)$. These two convex conjugate functions provide an important relation between natural parameter and mean parameter: $\boldsymbol{\mu} = \nabla A(\boldsymbol{\theta})$ and $\boldsymbol{\theta} = \nabla H(\boldsymbol{\mu})$ (Rockafellar, 2015).

The two convex conjugate functions will induce a pair of dual Bregman divergences, and then the dual Bregman divergences will further induce a pair of dual Riemannian manifolds. Given the log-partition function $A(\boldsymbol{\theta})$, it induces a Bregman divergence $B_A(\boldsymbol{\theta}, \boldsymbol{\theta}') : \boldsymbol{\Theta} \times \boldsymbol{\Theta} \to \mathbb{R}^+$ that defines a positive definite Riemannian metric $\nabla^2 A(\boldsymbol{\theta})$ because $A(\boldsymbol{\theta})$ is a strictly convex and twice differentiable function. Therefore, $B_A(\boldsymbol{\theta}, \boldsymbol{\theta}')$ induces the Riemannian manifold $(\boldsymbol{\Theta}, \nabla^2 A(\boldsymbol{\theta}))$. Similarly, if all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ are transformed by $\nabla A(\boldsymbol{\theta})$, we can get the mean parameter $\boldsymbol{\mu} \in \mathcal{M}$. The Bregman divergence $B_H(\boldsymbol{\mu}, \boldsymbol{\mu}') : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+$ induces a Riemannian manifold $(\mathcal{M}, \nabla^2 H(\boldsymbol{\mu}))$. $(\boldsymbol{\Theta}, \nabla^2 A(\boldsymbol{\theta}))$ is called the primal Riemannian manifold and $(\mathcal{M}, \nabla^2 H(\boldsymbol{\mu}))$ is called the dual Riemannian manifold. Raskutti & Mukherjee (2015) proved the equivalence between mirror descent and natural gradient descent by the application of chain rule and dual Riemannian manifolds. We restate the proof here.

Proof. Assume that we use a mirror descent to maximize the ELBO $\tilde{\mathcal{L}}(\boldsymbol{\mu}) = \mathcal{L}(\boldsymbol{\theta})$ over the mean parameter $\boldsymbol{\mu}$ using the Bregman divergence induced by the negative entropy function $H(\boldsymbol{\mu})$:

$$\boldsymbol{\mu}_{t+1} = \arg\max_{\boldsymbol{\mu} \in \mathcal{M}} \nabla \tilde{\mathcal{L}}(\boldsymbol{\mu}_t)^\top \boldsymbol{\mu} - \frac{1}{\rho_t} B_H(\boldsymbol{\mu}, \boldsymbol{\mu}_t).$$

Substituting $B_H(\boldsymbol{\mu}, \boldsymbol{\mu}_t) = H(\boldsymbol{\mu}) - H(\boldsymbol{\mu}_t) - \nabla H(\boldsymbol{\mu}_t)^\top (\boldsymbol{\mu} - \boldsymbol{\mu}_t)$, we can obtain the solution:

$$\nabla H(\boldsymbol{\mu}_{t+1}) = \nabla H(\boldsymbol{\mu}_t) + \rho_t \nabla \tilde{\mathcal{L}}(\boldsymbol{\mu}_t).$$

Using $\boldsymbol{\mu} = \nabla A(\boldsymbol{\theta})$ and $\boldsymbol{\theta} = \nabla H(\boldsymbol{\mu})$, we obtain the following equivalent expression:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \rho_t \nabla_{\boldsymbol{\mu}} \tilde{\mathcal{L}}(\nabla A(\boldsymbol{\theta}_t)).$$

Substituting the chain rule $\nabla_{\boldsymbol{\theta}} \tilde{\mathcal{L}}(\nabla A(\boldsymbol{\theta})) = \nabla^2_{\boldsymbol{\theta}} A(\boldsymbol{\theta}) \nabla_{\boldsymbol{\mu}} \tilde{\mathcal{L}}(\nabla A(\boldsymbol{\theta}))$, we obtain:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \rho_t [\nabla^2_{\boldsymbol{\theta}} A(\boldsymbol{\theta}_t)]^{-1} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_t),$$

which is just the natural gradient descent in Equation (2). □

C. Gradients w.r.t. Mean Parameters

In this section, we provide a proof of Section 3.3 in the main paper.

Proof. Given the softmax likelihood: $p(\mathbf{y}_n|\mathbf{f}_n) = \prod_c \exp(f_n^c \cdot y_n^c) / \sum_c \exp(f_n^c)$, we can obtain the log-likelihood and the first and the diagonal of second-order gradients of log-likelihood w.r.t. $\mathbf{f}_n$:

$$\log p(\mathbf{y}_n|\mathbf{f}_n) = \mathbf{f}_n \cdot \mathbf{y}_n - \log(\sum_c \exp(\mathbf{f}_n)),$$

$$\nabla_{\mathbf{f}_n} \log p(\mathbf{y}_n|\mathbf{f}_n) = \mathbf{y}_n - \frac{\exp(\mathbf{f}_n)}{\sum_c \exp(\mathbf{f}_n)},$$

$$\mathrm{diag}(\nabla^2_{\mathbf{f}_n} \log p(\mathbf{y}_n|\mathbf{f}_n)) = \frac{\exp(2\mathbf{f}_n)}{(\sum_c \exp(\mathbf{f}_n))^2} - \frac{\exp(\mathbf{f}_n)}{\sum_c \exp(\mathbf{f}_n)}.$$

Given a factorized variaitonal distribution $q(\mathbf{f}|\boldsymbol{\theta}) = \prod_c \mathcal{N}(\mathbf{f}^c|\boldsymbol{\theta}^c)$, the ELBO can be written as:

$$\mathcal{L} = \sum_n \mathbb{E}_q[\log p(\mathbf{y}_n|\mathbf{f}_n)] - \sum_c \mathrm{KL}[\mathcal{N}(\mathbf{f}^c|\boldsymbol{\theta}^c) \| \mathcal{N}(\mathbf{f}^c|\mathbf{0}, \mathbf{K}^c)].$$

If we parameterize $q(\mathbf{f}_n)$ as $\mathcal{N}(\mathbf{f}_n|\mathbf{m}_n, \text{diag}(\mathbf{v}_n))$, where $\text{diag}(\mathbf{v}_n)$ is a diagonal matrix corresponding to vector $\mathbf{v}_n$ due to the assumption of independence between latent functions of different classes, the gradient of $\mathbb{E}_{q(\mathbf{f}_n)}[\log p(\mathbf{y}_n|\mathbf{f}_n)]$ w.r.t. $\mathbf{m}_n$ and $\mathbf{v}_n$ can be expressed as:

$$\mathbf{g}_{\mathbf{m}_n} \coloneqq \nabla_{\mathbf{m}_n}\mathbb{E}_{q(\mathbf{f}_n)}[\log p(\mathbf{y}_n|\mathbf{f}_n)] = \mathbb{E}_q \nabla_{\mathbf{f}_n} \log p(\mathbf{y}_n|\mathbf{f}_n) = \mathbb{E}_q\left[\mathbf{y}_n - \frac{\exp(\mathbf{f}_n)}{\sum_c \exp(\mathbf{f}_n)}\right],$$

$$\mathbf{g}_{\mathbf{v}_n} \coloneqq \nabla_{\mathbf{v}_n}\mathbb{E}_{q(\mathbf{f}_n)}[\log p(\mathbf{y}_n|\mathbf{f}_n)] = \frac{1}{2}\mathbb{E}_q \text{diag}(\nabla^2_{\mathbf{f}_n} \log p(\mathbf{y}_n|\mathbf{f}_n)) = \frac{1}{2}\mathbb{E}_q\left[\frac{\exp(2\mathbf{f}_n)}{(\sum_c \exp(\mathbf{f}_n))^2} - \frac{\exp(\mathbf{f}_n)}{\sum_c \exp(\mathbf{f}_n)}\right].$$

according to Opper & Archambeau (2009).

Given the relation between the traditional parameters and the mean parameters of an independent multivariate Gaussian distribution: $\mathbf{m}_n = \boldsymbol{\mu}_n^{(1)}$ and $\mathbf{v}_n = \boldsymbol{\mu}_n^{(2)} - \boldsymbol{\mu}_n^{(1)2}$ where $\boldsymbol{\mu}_n^{(1)}$ and $\boldsymbol{\mu}_n^{(2)}$ are two mean parameters of $q(\mathbf{f}_n)$, we can use the chain rule to express the gradient w.r.t. the mean parameters:

$$\nabla_{\boldsymbol{\mu}_n^{(1)}}\mathbb{E}_{q(\mathbf{f}_n)}[\log p(\mathbf{y}_n|\mathbf{f}_n)] = \mathbf{g}_{\mathbf{m}_n} - 2\mathbf{g}_{\mathbf{v}_n} \circ \mathbf{m}_n,$$
$$\nabla_{\boldsymbol{\mu}_n^{(2)}}\mathbb{E}_{q(\mathbf{f}_n)}[\log p(\mathbf{y}_n|\mathbf{f}_n)] = \mathbf{g}_{\mathbf{v}_n}.$$

□

D. Computational Complexity for Equation (6a)

Note that the computational complexity of Monte Carlo in Equation (6a) comes from covariance decomposition and sample drawing. The first part costs $O(MN^2)$ and the second part costs $O(N^3)$, where N is the support size and M is the samples drawn. In a few-shot setting, M is much larger than N, so the total computational complexity is $O(MN^2)$, which does not pose a challenge to computational resources and is easily manageable.

E. Experimental Details

E.1. Datasets

We use four datasets as described below.

1. **Caltech-UCSD Birds (CUB).** CUB consists of a total of 11788 bird images from 200 distinct classes. The standard split of 100 training, 50 validation, and 50 test classes is employed (Snell & Zemel, 2021).

2. **mini-ImageNet.** mini-ImageNet is a small part of the full ImageNet dataset, where 100 classes with 600 images each are selected to form the dataset. We employed the common split of 64 training, 16 validation, and 20 test classes as well (Snell & Zemel, 2021).

3. **Omniglot.** There are 1623 grayscale characters selected from 50 languages contained in this dataset, which is further expanded to 6492 images by data augmentation (Lake et al., 2011). We use 4114 for training.

4. **EMNIST.** EMNIST (Cohen et al., 2017) consists of 62 classes of single digits and English characters. In the domain transfer task, we utilize 31 for validation and the other for test.

E.2. Comparison of Baselines

As for the description of baseline methods, we refer to Snell & Zemel (2021) for a detailed overview. Here we only compare the methods that are most similar to ours, which include DKT, Logistic-softmax, and OVE. All of these methods utilize a similar framework but with different likelihood and inference methods from ours.

1. **Deep Kernel Transfer (DKT)** (Patacchiola et al., 2020) employed the Gaussian likelihood to circumvent the conjugacy issue where the labels $\{+1, -1\}$ are treated as continuous values, leading to suboptimal accuracy.

2. **Logistic-softmax** (Galy-Fajou et al., 2020) applied the logistic-softmax for a conditional conjugate model after data augmentation. The Gibbs sampling version implemented by Snell & Zemel (2021) and the mean-field approximation version implemented by Ke et al. (2023) are considered for comparison.

3. **One-vs-Each Approximation (OVE)** (Snell & Zemel, 2021) proposed to approximate the softmax function for conditional conjugacy after data augmentation and applied Gibbs sampling for posterior inference.

E.3. Training Protocols

The Adam optimizer with a standard learning rate of 10^{-3} for the neural network and a learning rate of 10^{-4} for other kernel parameters is employed across all our experiments in the outer loop. For a single epoch, 100 random episodes are sampled from the complete dataset for all methods. As for the steps used for variational inference, we run 3 steps with $\rho = 1$ during training time and 50 steps during testing time with $\rho = 0.5$. We employ the COS kernel for CUB and mini-ImageNet and the RBF kernel for Omniglot and EMNIST.

E.4. Kernels

To begin with, we briefly review the definition of deep kernel proposed in Wilson et al. (2016). Deep kernel combines the traditional kernel with modern neural networks in a straightforward formulation, defined as $k(\mathbf{x}, \mathbf{x}' \mid \boldsymbol{\theta}, \mathbf{w}) = k'(g_{\mathbf{w}}(\mathbf{x}), g_{\mathbf{w}}(\mathbf{x}') \mid \boldsymbol{\theta})$, where k' represents a predefined base kernel with parameters $\boldsymbol{\theta}$. The inputs are projected by a deep neural network to increase flexibility. Note that the deep kernel parameters consist of $\boldsymbol{\theta}$ and $\mathbf{w}$. One upside of deep kernel is its potential to learn beyond Euclidean distance-based metrics via data-driven optimization. In the following, we provide the expressions for the relevant base kernels discussed in the main paper.

1. **Polynomial Kernel (POL).** The polynomial kernel is defined as

$$k'(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^\top \mathbf{x}' + c)^s,$$

   where s is the order of the kernel and c is a trainable parameter.

2. **Radial Basis Function Kernel (RBF).** The RBF kernel is defined as

$$k'(\mathbf{x}, \mathbf{x}') = \exp\Big(- \frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2l^2} \Big),$$

   where l is referred to as the length-scale parameter. The RBF kernel is smooth because of the exponential eigenvalue decay rate.

3. **Cosine Similarity Kernel (COS).** The cosine similarity kernel is defined as

$$k'(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^\top \mathbf{x}'}{\|\mathbf{x}\| \, \|\mathbf{x}'\|}.$$

   We use a variant of the kernel with batch normalization that centers the input vectors (Patacchiola et al., 2020).

E.5. Comparision of Actual Running Time

To further investigate the convergence rate in practice, we give a brief comparison of the actual running time between our method and other first-order methods in Table 4. We use the CUB dataset and run each method with 3 steps for the inner loop except CDKT (PL). The time reported for the inner loop and outer loop are from the last sampled task for the first epoch. We use one Quadro RTX 6000 to run each method. Note that CDKT (ML) is one of the sota baselines that use a very similar training procedure, where its inner loop is the mean-field approximation steps, and CDKT (PL) uses predictive likelihood with no inner loop so we just compare the total run time of an epoch. As can be seen from the table, our method has a similar run time compared to other first-order method. The result proves that our method indeed enjoys first-order computation and is comparable to other first-order methods in actual running time.

Table 4. Comparison of actual running time between MD-BSFC and other first-order methods with similar implementations on the CUB dataset. The inner loop is run with 3 steps.

	MD	**GD**	**CDKT (ML)**	**CDKT(PL)**
Inner Loop	0.0181	0.0172	0.0141	NA
Outer Loop	0.1035	0.0989	0.0814	NA
One Epoch	12.1253	12.0518	10.8549	12.5445

F. Related Work

To address the non-conjugate issue, Patacchiola et al. (2020) assumes a Gaussian likelihood for class labels to circumvent the non-conjugate computation, which exhibits inferior uncertainty quantification since class labels are discrete rather than continuous. Differently, Snell & Zemel (2021) proposes a novel approach that combines the Pólya-Gamma augmentation technique (Polson et al., 2013) and the one-vs-each softmax approximation (Titsias, 2016) to transform the non-conjugate GP classification model into a conditionally conjugate model. Ke et al. (2023) also utilizes the Pólya-Gamma augmentation with the logistic-softmax likelihood to achieve conditional conjugacy. These methods offer better uncertainty quantification but require additional auxiliary variables. Differing from the approaches mentioned above, we incorporate mirror descent-based variational inference into GP classification to obtain a conjugate algorithm without introducing any auxiliary variables.

The exploration of Bayesian meta-learning has yielded diverse approaches aimed at leveraging prior knowledge and adapting to new tasks. The framework proposed by Finn et al. (2018) adopts a Bayesian hierarchical modeling perspective, enabling the capture of uncertainty at various levels. In a different vein, Grant et al. (2018) reformulated meta-learning as inference in a GP, while Yoon et al. (2018) introduced Bayesian Model-Agnostic Meta-Learning (MAML) building upon the work of Finn et al. (2017). Notably, Patacchiola et al. (2020); Snell & Zemel (2021) employed GPs with deep kernels to facilitate task-specific inference, thereby contributing to Bayesian meta-learning in terms of parameter updates, uncertainty modeling, and prior distributions. In this context, our work makes a valuable contribution by presenting an effective alternative for task-level updates and offering insights into the coordination problem associated with bi-level optimization.