Amortized Bayesian Prototype Meta-learning: A New Probabilistic Meta-learning Approach to Few-shot Image Classification

Zhuo Sun[1] **Jijie Wu**[2] **Xiaoxu Li**[2] **Wenming Yang**[3] **Jing-Hao Xue**[1]
[1]University College London [2]Lanzhou University of Technology [3]Tsinghua University

Abstract

Probabilistic meta-learning methods recently have achieved impressive success in few-shot image classification. However, they introduce a huge number of random variables for neural network weights and thus severe computational and inferential challenges. In this paper, we propose a novel probabilistic meta-learning method called amortized Bayesian prototype meta-learning. In contrast to previous methods, we introduce only a small number of random variables for latent class prototypes rather than a huge number for network weights; we learn to learn the posterior distributions of these latent prototypes in an amortized inference way with no need for an extra amortization network, such that we can easily approximate their posteriors conditional on few labeled samples, whenever at meta-training or meta-testing stage. The proposed method can be trained end-to-end without any pre-training. Compared with other probabilistic meta-learning methods, our proposed approach is more interpretable with much less random variables, while still be able to achieve competitive performance for few-shot image classification problems on various benchmark datasets. Its excellent robustness and predictive uncertainty are also demonstrated through ablation studies.

1 Introduction

Humans are able to quickly grasp new concepts from a small number of samples in a new domain. Such a remarkable ability was built on i) good leverage of past relevant experience and ii) fast adaption to novel concepts. In contrast, deep learning (e.g., LeCun et al., 2015) often requires a large amount of data to grasp a novel concept. However, it is expensive or impossible to collect a large dataset of labeled samples in a novel domain. The challenging problem to learn a novel concept when few examples are available in the new domain is often referred to as few-shot learning.

Meta-learning (e.g., Bartunov and Vetrov, 2018; Jamal and Qi, 2019; Finn et al., 2017; Grant et al., 2018; Amit and Meir, 2018; Li et al., 2019; Xu et al., 2020; Ren et al., 2019; Rusu et al., 2019; Sun et al., 2019; Hospedales et al., 2020; Wang et al., 2019; Li et al., 2020; Iakovleva et al., 2020; Patacchiola et al., 2020), or learning to learn, aims to develop methods that can solve novel tasks based on experience established through the meta-training process of previous tasks. Meta-learning methods have achieved state-of-the-art performance in few-shot classification on many image datasets, e.g., *mini-ImageNet* (Vinyals et al., 2016) and *CUB-200-2011* (Wah et al., 2011). Built on probabilistic structures over data and parameters of neural networks and the power of Bayesian inference, probabilistic meta-learning methods (Grant et al., 2018; Gordon et al., 2019; Ravi and Beatson, 2019; Finn et al., 2018; Yoon et al., 2018; Nguyen et al., 2020; Patacchiola et al., 2020) are able to learn the posteriors of parameters and then use posterior predictive samples to solve novel tasks.

However, these probabilistic methods treat the network weights as random variables, introducing a huge number of random variables and consequently severe computational and inferential problems, e.g. identifiability. *Our aim is: can we instead introduce an embedding space with much less random variables while still well representing the generative process of meta-learning?* To this end, we introduce latent class prototypes to probabilistic meta-learning, which largely reduce the number of random variables while achieving competitive performance. A latent class prototype is a latent random variable that has a distribution defining the generative process of this class (Fig. 1).

Proceedings of the 24[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

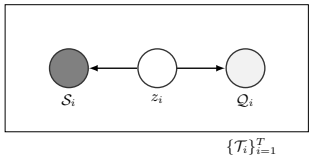

Figure 1: *Graphical Model of Our Method.* z_i is a latent random vector denoting the class prototypes *(white node: unobserved)* for task $\mathcal{T}_i$, which contains a support set $\mathcal{S}_i$ *(dark gray node: label observed during meta-training and meta-testing)* and a query set $\mathcal{Q}_i$ *(light gray node: label observed only during meta-training).*

To summarize, our main contributions are as follows.

First, we propose a novel and simple probabilistic meta-learning method, *amortized Bayesian prototype meta-learning*, for few-shot classification. It introduces a new latent random vector z serving as class prototypes and learns to learn the posterior distributions of these latent variables whenever at meta-training or meta-testing stage. We achieve this by carefully designing the prior and variational distributions for the latent class prototypes: Instead of using the trivial standard Gaussian distributions as priors, we propose to use task-dependent priors for latent class prototypes conditional on the support set of each task and replace the $\mathbb{KL}$ divergence term in the evidence lower bound of marginal log-likelihood of the support set with an unbiased estimator to its expectation.

Secondly, the proposed method can be trained end-to-end without any pre-training and achieves state-of-the-art or competitive performance on many real-world benchmark datasets, e.g., *mini-ImageNet* (Vinyals et al., 2016), *Stanford-dogs* (Khosla et al., 2011) and *CUB-200-2011* (Wah et al., 2011). Inference is amortized via learning a shared set of parameters such that a few steps of gradient descent can fast generate well-behaved approximate posteriors of latent prototypes.

Thirdly, through ablation studies, we demonstrate the robustness of our method that it can preserve high performance when altering the number of ways or shots at meta-testing stage without the need to re-train the whole neural network. We also demonstrate the excellent quality of predictive uncertainty of our model.

2 Preliminaries

Meta-learning Given a series of tasks $\{\mathcal{T}_t\}_{t=1}^T$ sampled from an environment ρ, meta-learning, also called learning to learn, aims to learn an algorithm $\mathcal{A}$ that has minimum transfer risk on a new task, $\mathcal{R} = \mathbb{E}_{\mathcal{T}\sim\rho}\mathbb{E}_{\{x_i,y_i\}_{i=1}^n\sim\mathcal{T}^n}\mathbb{E}_{\{\tilde{x},\tilde{y}\}\sim\mathcal{T}}[l(\mathcal{A}(\{x_i,y_i\}_{i=1}^n),\{\tilde{x},\tilde{y}\})]$, where $l(\cdot)$ is some loss function, and $\{x_i,y_i\}_{i=1}^n$ are n training samples sampled from a new random task $\mathcal{T}$ from the environment ρ, which is distinguished from $\{\mathcal{T}_t\}_{t=1}^T$, and $\{\tilde{x},\tilde{y}\}$ is a random test pair sampled from the same task. The environment ρ refers to a meta distribution of tasks $\mathcal{T}$ (Denevi et al., 2019, 2018; Baxter, 2000). To simplify notation and bridge the gap to few-shot learning, we increase the number of test samples to m, i.e. $\{\tilde{x}_j,\tilde{y}_j\}_{j=1}^m$, and denote $\{x_i,y_i\}_{i=1}^n$ by $\mathcal{S}$ and $\{\tilde{x}_j,\tilde{y}_j\}_{j=1}^m$ by $\mathcal{Q}$, and we have

$$\mathcal{R} = \mathbb{E}_{\mathcal{T}\sim\rho}\mathbb{E}_{\mathcal{S}\sim\mathcal{T}^n}\mathbb{E}_{\mathcal{Q}\sim\mathcal{T}^m}\left[l\Big(\mathcal{A}(\mathcal{S}),\mathcal{Q}\Big)\right]. \tag{1}$$

The expectations are often approximated by Monte Carlo integration. Therefore, given a new task $\mathcal{T} = \mathcal{S}\cup\mathcal{Q}$, the learned algorithm $\mathcal{A}$ is often further adapted to the sampled support set $\mathcal{S}$ and its performance for this task is measured by the empirical loss of a sampled query set $\mathcal{Q}$. The overall performance of $\mathcal{A}$ is the averaged performance on tasks sampled from ρ.

Few-shot Image Classification When n is considerably small, this corresponds to few-shot learning. To train and test a meta algorithm $\mathcal{A}$, a dataset $\mathcal{D}$ is divided into three parts, namely $\mathcal{D}_{tr} = \{\{x_{tr,j},y_{tr,j}\}_{j=1}^{J_{tr}}, y_{tr,j}\in\mathcal{C}_{tr}\}$, $\mathcal{D}_{val} = \{\{x_{val,j},y_{val,j}\}_{j=1}^{J_{val}}, y_{val,j}\in\mathcal{C}_{val}\}$, $\mathcal{D}_{te} = \{\{x_{te,j},y_{te,j}\}_{j=1}^{J_{te}}, y_{te,j}\in\mathcal{C}_{te}\}$, where $\{x_j,y_j\}$ is the jth image in the associated dataset; J_{tr}, J_{val} and J_{te} are the total numbers of samples in $\mathcal{D}_{tr}$, $\mathcal{D}_{val}$ and $\mathcal{D}_{te}$, respectively; and $\mathcal{C}_{tr}$, $\mathcal{C}_{val}$ and $\mathcal{C}_{te}$ are the three associated label sets. In this work, as we focus on few-shot image classification, we further require that the label sets $\mathcal{C}_{tr}$, $\mathcal{C}_{val}$ and $\mathcal{C}_{te}$ are mutually disjoint.

A task $\mathcal{T}$ in few-shot image classification literature often refers to a C-way K-shot problem, which is sampled from an environment ρ. In few-shot learning literature, ρ actually refers to the whole label sets $\mathcal{C} = \mathcal{C}_{tr}\cup\mathcal{C}_{val}\cup\mathcal{C}_{te}$, of which realisations are the samples in the whole dataset $\mathcal{D} = \mathcal{D}_{tr}\cup\mathcal{D}_{val}\cup\mathcal{D}_{te}$. To generate realisations of a task $\mathcal{T}$, firstly, we randomly sample C classes from a set $\mathcal{C}^*$, where $\mathcal{C}^*$ can be either $\mathcal{C}_{tr}$, $\mathcal{C}_{val}$ or $\mathcal{C}_{te}$ (corresponding to meta-training, meta-validation and meta-testing stages, respectively). Then we further sample n and m instances from these C classes to construct a support set $\mathcal{S}$ and a query set $\mathcal{Q}$ to construct a C-class classification problem.

To simplify notation and terms, in the following, a task $\mathcal{T}$ is referred to as a dataset that contains a support set $\mathcal{S}$ and a query set $\mathcal{Q}$, of which instances are all from some C classes. A C-way K-shot few-shot learning problem means that we only have $n = C\times K$ samples in the support set for any tasks generated

from $\mathcal{D}_{tr}$, $\mathcal{D}_{val}$ or $\mathcal{D}_{te}$. A meta algorithm is usually trained/tested on $\mathcal{D}_{tr}/\mathcal{D}_{te}$ through a series of tasks $\{\mathcal{T}_{tr,t}\}_{t=1}^{T_{tr}}$ / $\{\mathcal{T}_{te,t}\}_{t=1}^{T_{te}}$, following the standard meta-training/meta-testing procedures proposed by Vinyals et al. (2016). $\mathcal{D}_{val}$ is only used for model selection, and the overall performance on $\mathcal{D}_{te}$ is reported as the generalization performance of the model, measured empirically by the mean accuracy of all query sets of the T_{te} tasks $\{\mathcal{T}_{te,t}\}_{t=1}^{T_{te}}$ sampled from $\mathcal{D}_{te}$.

3 Related Work

Probabilistic Meta-learning Methods Our method is based on *MAML* (Finn et al., 2017), a gradient-based meta-learning method aiming to learn a shared initialization of neural network's parameters that has excellent generalization ability to an unseen novel task with only few steps of stochastic gradient descent. Probabilistic variants of *MAML* include *MAML-HB* (Grant et al., 2018), *BMAML* (Yoon et al., 2018), *PLATIPUS* (Finn et al., 2018), *VAMPIRE* (Nguyen et al., 2020), *Meta-Mixture* (Jerfel et al., 2019) and *ABML* (Ravi and Beatson, 2019). *MAML-HB* interprets *MAML* as a hierarchical Bayes learning procedure. *PLATIPUS* proposes to learn the joint posterior of meta initialization θ and task-specific parameters conditional on the support set of each task $\mathcal{T}_i$, while *BMAML*, *VAMPIRE* and *ABML* learn the posterior distributions of the task-specific parameters conditional on the θ and the support set. More specific, *BMAML* learns the posteriors of task-specific parameters through Stein variational gradient descent, which is distinct from the others. *ABML* proposes to minimize the loss of the support and query sets of a task jointly (i.e. equivalent to maximizing $\mathbb{E}[\log p(\mathcal{S},\mathcal{Q})]$), which does not explicitly encourage neural networks to maximize $\mathbb{E}[\log p(\mathcal{Q}|\mathcal{S})]$. *VAMPIRE* is similar to *ABML*. The main difference is that *VAMPIRE* only uses the loss of the query set of a task to update the global shared parameter θ, while *ABML* uses both the support and query sets. *VERSA* (Gordon et al., 2019) proposes to directly maximize $\log p(\mathcal{Q}|\mathcal{S})$, which is achieved by only learning the posteriors of parameters of the linear classifier via an extra amortization network.

Metric-based Methods Metric-based methods for few-shot classification (Vinyals et al., 2016; Snell et al., 2017; Sung et al., 2018; Allen et al., 2019) employ some metric for all tasks and learn an appropriate feature mapping that best captures the discriminative information for tasks, and expect the learned feature mapping to generalize well to novel tasks from unseen classes. *Prototype Network* (Snell et al., 2017) uses Euclidean distance, which implicitly assumes features form Gaussian class-specific distributions with identity covariance matrices. *Infinite Mixture Prototype Network* (Allen et al., 2019) assumes multiple clusters within a class through DP-means or Chinese restaurant process, and still employs Euclidean distance as the evaluation metric. It also implicitly assumes that each cluster is Gaussian distributed with shared covariance matrices among clusters. There are also other non-standard similarity-based methods, e.g., *Relation Network* (Sung et al., 2018), which uses a so-called *deep relation* similarity measurement.

Key Difference from Previous Work Our proposed approach has some key difference from previous work in the following aspects. 1) Previous probabilistic meta-learning methods (e.g., Gordon et al., 2019; Ravi and Beatson, 2019; Finn et al., 2018; Yoon et al., 2018; Nguyen et al., 2020; Jerfel et al., 2019) treat weights of neural networks as random variables, while our method introduces latent prototypes as random variables. 2) Our approach learns to learn the posterior distributions of latent class prototypes in an amortized inference way with no need for an extra amortization network. 3) Assuming that support images and query images come from same data generating process, we replace the $\mathbb{KL}$ term in the evidence lower bound of $\log p_\theta(\mathcal{S})$ with an unbiased estimator to its expectation (Eq.7), which is purely dependent on the support set of a task. In addition, the proposed method provides us with a more interpretable and simpler way of modelling, and achieves state-of-the-art or competitive performance on various benchmark datasets.

4 The Proposed Method

4.1 Meta-learning via Maximizing Expectation of Posterior Predictive Likelihood

Suppose $\mathcal{S}$ and $\mathcal{Q}$ are two random variables representing a support set and a query set, respectively. Consider a probabilistic generative model, parametrized by θ, which defines a prior $p_\theta(z)$ on latent variables z and a conditional likelihood $p_\theta(\mathcal{S}|z)$ on the support set $\mathcal{S}$. To approximate the posterior $p_\theta(z|\mathcal{S})$, we can use the evidence lower bound of $\log p_\theta(\mathcal{S})$:

$$\log p_\theta(\mathcal{S}) \geq \mathbb{E}_{z\sim q_\phi(z)}[\log p_\theta(\mathcal{S}|z)] - \mathbb{KL}[q_\phi(z)||p_\theta(z)]. \tag{2}$$

The lower bound is tight when $q_\phi(z) = p_\theta(z|\mathcal{S})$. We can optimize the evidence lower bound with respect to the variational parameters ϕ to obtain the approximate posterior $q_\phi(z)$. Given $q_\phi(z) \approx p_\theta(z|\mathcal{S})$, we can approximate the log posterior predictive likelihood of

$\mathcal{Q}$ conditional on $\mathcal{S}$. That is,

$$\begin{aligned}\log p_\theta(\mathcal{Q}|\mathcal{S}) &= \log \int p_\theta(\mathcal{Q}|z)p_\theta(z|\mathcal{S})\, dz \\ &\approx \log \mathbb{E}_{z\sim q_\phi(z)}[p_\theta(\mathcal{Q}|z)] \\ &\geq \mathbb{E}_{z\sim q_\phi(z)}[\log p_\theta(\mathcal{Q}|z)] \quad . \end{aligned} \tag{3}$$

This indicates that, given θ, we need to first optimize the lower bound of the log-likelihood of $\mathcal{S}$, $\log p_\theta(\mathcal{S})$, to approximate the posterior $p_\theta(z|\mathcal{S})$, and then we can optimize the lower bound of conditional likelihood $\log p_\theta(\mathcal{Q}|\mathcal{S})$ (R.H.S. of Eq.3) w.r.t. θ. Maximizing the expectation of $\log p_\theta(\mathcal{Q}|\mathcal{S})$ is equivalent to minimizing the transfer risk defined in Eq.1 when $l(\mathcal{A}(\mathcal{S}), \mathcal{Q}) \triangleq -\log p_\theta(\mathcal{Q}|\mathcal{S})$. The expectation of $\log p_\theta(\mathcal{Q}|\mathcal{S})$ can be well approximated by the average of $\{\log p_\theta(\mathcal{Q}_i|\mathcal{S}_i)\}_{i=1}^T$ of tasks $\{\mathcal{T}_i\}_{i=1}^T$ when the total number T of tasks is large, which is often the case in few-shot learning (e.g., Vinyals et al., 2016). However, in few-shot classification, $\{\mathcal{T}_i\}_{i=1}^T$ arrive in a sequence, and we cannot use Monte Carlo integration, i.e., the average, to approximate the expectation as this requires to know all tasks and compute all relevant terms simultaneously. Instead, we can iteratively update θ through $\log p_\theta(\mathcal{Q}_i|\mathcal{S}_i)$, for $i = 1, \ldots, T$. As the uncertainty of the posterior of θ is low due to a large number of tasks generated during meta-training, a point estimate of θ is reasonable, as discussed by Ravi and Beatson (2019).

Hence, learning is achieved by introducing auxiliary latent random variables z so that we can deal with intractable likelihood and posteriors. In this work, we propose to assume that there exists a generative process such that images can be generated from latent random variables $z = [z_1, \ldots, z_C]^\top$ which represent C class prototypes for a C-way K-shot task. The inferred posterior $q_\phi(z)$ can then be readily used as a learnable discriminative classifier for classification tasks.

4.2 Amortized Bayesian Prototype Meta-learning

In this section, we first give an overview of our proposed amortized Bayesian prototype meta-learning approach and algorithm, summarize its novelty and strength, and then present technical rationale and details of variational distributions, prior distributions and classification loss we design for this new approach.

Overview Instead of inferring variational parameters ϕ_i for each task $\mathcal{T}_i = \mathcal{S}_i \cup \mathcal{Q}_i$, we can make a global model V learn to estimate those variational parameters dependent on the dataset $\mathcal{S}_i$ and a shared set of parameters θ jointly (Marino et al., 2018), i.e., $\phi_i = V(\mathcal{S}_i, \theta)$ for all i. This is referred to as amortized variational inference (Kingma and Welling, 2013; Marino et al., 2018; Ravi and Beatson, 2019). In this work, we follow Grant et al. (2018) and Ravi and Beatson (2019) and set $q_{\phi_i}(z) = \mathcal{N}(z; \mu_{\phi_i}, \Sigma_{\phi_i})$, where the variational parameters $\phi_i = [\mu_{\phi_i}, \Sigma_{\phi_i}]$ can be obtained by $\phi_i = V(\mathcal{S}_i, \theta) \triangleq \theta + \alpha \nabla_\theta \log p_\theta(\mathcal{S}_i)$ for all i. Inference for all tasks is amortized by the shared set of parameters θ. To simplify notation, we drop the subscript i in the following.

In this work, for a C-way K-shot task, we assume that $z = [z_1, \cdots, z_C]^\top$ is a latent random vector. Each of its elements follows a class-specific Gaussian distribution parametrized by ϕ, and we set $\phi = V(\mathcal{S}, \theta)$. Since the first term in the evidence lower bound of $\log p_\theta(\mathcal{S})$ (R.H.S. of Eq.2) and the lower bound of $\log p_\theta(\mathcal{Q}|\mathcal{S})$ (Eq.3) represent negative reconstruction loss, we can reformulate them by a negative classification loss $-\mathcal{L}_{PR}$ for classification tasks (given in Eq.9). In this way, following Eq.2 and Eq.3 and given a task $\mathcal{T} = \mathcal{S} \cup \mathcal{Q}$, our loss on $\mathcal{S}$ and our meta-loss for meta-update of θ can be respectively expressed as

$$\mathcal{L}(\mathcal{S}) \triangleq \mathcal{L}_{PR}(\mathcal{S}|z) + \mathbb{KL}[q_\phi(z|\mathcal{S}) \,||\, p_\theta(z)] \quad , \tag{4}$$

$$\mathcal{L}_{meta} \triangleq \mathcal{L}_{PR}(\mathcal{Q}|z) \quad . \tag{5}$$

Given that current value of θ is $\theta^{(i)}$, $\mathcal{L}(\mathcal{S})$ in Eq.4 is used to optimize variational parameters ϕ which are initialized at $\theta^{(i)}$, e.g., performing certain steps of stochastic gradient descent; and $\mathcal{L}_{meta}$ is used to update θ from the current value $\theta^{(i)}$ to $\theta^{(i+1)}$ conditional on the current query set $\mathcal{Q}$, e.g., $\theta^{(i+1)} = \theta^{(i)} - \beta \nabla_{\theta^{(i)}} \mathcal{L}_{meta}$. We summarize the meta-training process of the proposed method in Alg. 1. See the supplementary material for the pseudo-code of the meta-validation/meta-testing processes.

Novelty and Strength Our method aims to learn to learn posterior distributions $p_\theta(z|\mathcal{S})$ of the latent class prototypes z, in an amortized variational inference way. A significant difference from previous probabilistic meta-learning methods (e.g., Gordon et al., 2019; Ravi and Beatson, 2019; Finn et al., 2018; Yoon et al., 2018) is that we directly approximate the posterior distributions of the latent class prototypes, rather than generating stochastic classifiers via neural networks parametrized by random weights sampled from their approximate posteriors. With such an approximate posterior $q_\phi(z|\mathcal{S})$, our method is able to explicitly encourage better classification through the classification loss $\mathcal{L}_{PR}$. Moreover, in contrast to previous probabilistic meta-learning work, our method requires much less random variables and does not require Monte Carlo integration to approximate expectations of conditional likelihood $\mathbb{E}[\log p_\theta(\mathcal{S}|z)]$.

Variational Distributions We measure the uncertainty of each image's deep representation through an embedding network f_θ, which consists of a convolution neural network and a fully connected layer. More specific, we generate a feature vector $f_\theta(x) \in \mathbb{R}^{2p}$ for each image x after it passes through the embedding network f_θ. Then, we split this vector into two parts, $\mu(x) \in \mathbb{R}^p$ and $\sigma(x) \in \mathbb{R}^p$. To ensure a finite $\sigma(x)^2$, we further introduce two learnable scalars $w \in \mathbb{R}^1$ and $b \in \mathbb{R}^1$, and set $\sigma(x)^2 = |w| \odot S(\sigma(x)^2) \oplus |b|$, where $S(\cdot)$ is a element-wise sigmoid function, and $\{w, b\}$ are broadcast to match the dimension of $S(\sigma(x)^2) \in \mathbb{R}^p$, and $\odot$ and $\oplus$ are element-wise multiplication and element-wise addition operators, respectively. Afterwards, an image x can be represented by a Gaussian distribution centred around its mean vector, i.e. $\mathcal{N}(\mu(x), \Sigma(x))$, where $\Sigma(x) \in \mathbb{R}^{p\times p}$ is a diagonal matrix with diagonal elements $\sigma(x)^2 \in \mathbb{R}^p$. To aggregate distributions of images from the same class c into one single Gaussian $\mathcal{N}(\mu_c, \Sigma_c)$, we consider a matrix-version of harmonic average[1]. Letting $\mathcal{S}_c$ denote a subset of $\mathcal{S}$ containing all support images from the class $c \in [C]$, we have

$$\begin{aligned} \mu_c &= \left\{ \sum_{x_i \in \mathcal{S}_c} \Sigma(x_i)^{-1} \right\}^{-1} \left\{ \sum_{x_i \in \mathcal{S}_c} \Sigma(x_i)^{-1} \mu(x_i) \right\}, \\ \Sigma_c &= \left\{ \frac{1}{|\mathcal{S}_c|} \sum_{x_i \in \mathcal{S}_c} [\Sigma(x_i)^{-1}] \right\}^{-1}. \end{aligned} \tag{6}$$

We can set $q(z_c|\mathcal{S}_c) = \mathcal{N}(z_c; \mu_c, \Sigma_c)$ for all $c \in [C]$.

Prior Distributions of z Since z is no longer the weights of neural networks, it is not desirable to specify some trivial prior distributions (e.g., standard Gaussian distributions) as those in Grant et al. (2018); Gordon et al. (2019); Ravi and Beatson (2019); Finn et al. (2018); Yoon et al. (2018); Nguyen et al. (2020); Jerfel et al. (2019). In this work, we propose to use task-dependent priors for latent random variables, which is $\mathcal{N}(\mu(x), \Sigma(x))$ conditional on an image x sampled from the support set $\mathcal{S}$ of a task $\mathcal{T}$, because it has already been extracted via our neural network and no extra efforts are required. Due to the randomness induced by x, we propose to replace the $\mathbb{KL}$ divergence term in Eq.4 by its expectation,

$$\mathbb{E}_{\{x,y\}\sim\mathcal{T}} \Big[\mathbb{KL}[q_\phi(z|\mathcal{S}) || p_\theta(z; \mu(x), \Sigma(x))] \Big] . \tag{7}$$

Its unbiased estimator is given by $\frac{1}{CK} \sum_{c,i} \mathbb{KL}[q_\phi(z_c|\mathcal{S}_c) || \mathcal{N}(z_c; \mu(x_i^{(\mathcal{S}_c)}), \Sigma(x_i^{(\mathcal{S}_c)}))]$, where $x_i^{(S_c)}$ is the ith image in the subset $\mathcal{S}_c$. This is valid when $K > 1$. For $K = 1$, we may define a prior by using the query set and the support set jointly. However, it is arguable to use query images to construct prior distributions because label/information leaks during this process. Therefore, for the 1-shot classification, we propose to use the same aggregation rule in Eq.6 to further merge all C Gaussian distributions of the C support images into a single Gaussian distribution $\mathcal{N}(\mu_{union}, \Sigma_{union})$ to represent the prior distribution $p_\theta(z)$. This is a quite strong prior on latent random class prototypes, which may lead to shrinkage. But it is still reasonable if compared with trivial standard Gaussian densities. The estimator of the $\mathbb{KL}$ term in Eq.4 then becomes $\frac{1}{C} \sum_{c=1}^{C} \mathbb{KL}[q_\phi(z_c|\mathcal{S}_c) \,||\, \mathcal{N}(\mu_{union}, \Sigma_{union})]$.

Classification Loss For an image x, its class membership to each of the C classes is measured by

$$\Pr[\mu(x)|z_c] = \mathcal{N}(\mu(x); \mu_c, \Sigma_c) \ , \ c = 1, \ldots, C. \tag{8}$$

As the sum of the above equation over C classes is not restricted to be 1, it is vital to introduce normalization to attain a valid loss function, which is achieved by using log ratio of probabilities. Therefore, $\mathcal{L}_{PR}$ (in Eq.4 or Eq.5, with τ denoting either $\mathcal{S}$ or $\mathcal{Q}$) becomes

$$\mathcal{L}_{PR}(\tau|z) = -\frac{1}{|\tau|} \sum_{n=1}^{|\tau|} \left[\log \left(\frac{\Pr[\mu(x_n)|z_{y_n}]}{\sum_{c=1}^{C} \Pr[\mu(x_n)|\mathcal{S}_c]} \right) \right] . \tag{9}$$

To see the connection between the classification loss and $\mathbb{E}_z[\log p_\theta(\tau|z)]$, firstly note that $\log p_\theta(\tau|z) = \log(\prod_{n=1}^{|\tau|} p_\theta(y_n|x_n, z)) = \sum_{n=1}^{|\tau|} \log p_\theta(y_n|x_n, z)$. We know that the response y_n is discrete (taking values in $[C]$ in a C-way classification task) and should have a probability mass function, which can be achieved by normalizing over the sum of class membership: $p_\theta(y_n|x_n, z) = \Pr[\mu(x_n)|z_{y_n}]/(\sum_{c=1}^{C} \Pr[\mu(x_n)|z_c])$. It follows that $\log p_\theta(\tau|z) = \sum_{n=1}^{|\tau|} \log p_\theta(y_n|x_n, z) = \sum_{n=1}^{|\tau|} \log(\Pr[\mu(x_n)|z_{y_n}]/(\sum_{c=1}^{C} \Pr[\mu(x_n)|z_c]))$. An unbiased estimator of $\mathbb{E}_z[\log p_\theta(\tau|z)]$ is $\log p_\theta(\tau|z)$ when the sample size of z is one, which is a well-behaved case in our experiments as z are latent prototypes. Therefore, an unbiased estimator of the reconstruction loss $-\mathbb{E}_z[\log p_\theta(\tau|z)]$ is $-\sum_{n=1}^{|\tau|} \log(\Pr[\mu(x_n)|z_{y_n}]/(\sum_{c=1}^{C} \Pr[\mu(x_n)|z_c]))$, which is our classification loss (scaled by a factor of $1/|\tau|$). Although we replace the $\mathbb{KL}$ term in the evidence lower bound of $\log p_\theta(\mathcal{S})$ with Eq.7 as our proposed prior distributions of z are now dependent on the support set $\mathcal{S}$, our estimator to Eq.4 is still an unbiased estimator to evidence lower bound of $\log p_\theta(\mathcal{S})$ (scaled by a constant). Therefore the proposed method still learns to learn the approximate posteriors of latent z conditional on $\mathcal{S}$ properly. Details are presented in the supplementary material.

[1] Other forms of average can also be taken into account, e.g., simple average.

Algorithm 1: Meta-training for C-way K-shot classification

Input: Model $\mathcal{M}$, Optimizer, Number of mini-batches B, $\mathcal{D}_{tr}$, C, K.

for *b from 1 to B* **do**
  Generate a mini-batch of tasks $\mathcal{T}_i = \mathcal{S}_i \cup \mathcal{Q}_i$ from $\mathcal{D}_{tr}$, for $i = 1, ..., T$.
  for *each task* $\mathcal{T}_i$ *from 1 to T* **do**
    Initialize $\phi_i \leftarrow \theta$.
    /* Approximate inference for posteriors of z_i conditional on $\mathcal{S}_i$ */
    for *d from 1 to D* **do**
      Update: $\phi_i \leftarrow \phi_i - \alpha \nabla_{\phi_i} \{\mathcal{L}_{PR}(\mathcal{S}_i|z) + \mathbb{KL}[q_{\phi_i}(z|\mathcal{S}_i) \,||\, p_\theta(z)]\}$.
    Compute prediction loss: $\mathcal{L}(\mathcal{Q}_i|z) = \mathcal{L}_{PR}(\mathcal{Q}_i|z)$.
  /* Update globally shared parameters θ */
  Update: $\theta \leftarrow \theta - \beta \nabla_\theta \frac{1}{T} \sum_{i=1}^{T} \mathcal{L}(\mathcal{Q}_i|z)$.

Output: Return $\mathcal{M}$.

5 Few-shot Image Classification

5.1 Implementation Details

Network Architecture Our network only needs a feature embedding network. The embedding network can be deep networks, such as VGG styled convolution networks (Simonyan and Zisserman, 2015) or ResNet (He et al., 2016). In this work, to make fair comparisons with relevant work, we use a shallow network which consists of four convolution blocks followed by a fully-connected linear layer as our feature extractor f_θ. Each convolution block consists of 64 3-by-3 filters, followed by batch-normalization, ReLU activation, and 2-by-2 max-pooling. The fully-connected layer maps the flattened features from the four convolution blocks into vectors $\in \mathbb{R}^{128}$. Then we follow the settings in Sec.4.2 to formulate a Gaussian distribution for deep representation $f_\theta(x)$ of each image x.

Image Datasets *Omniglot* (Lake et al., 2011) is widely used as a toy image dataset. It contains 1623 classes from 50 languages. Each class contains 20 samples. For *Omniglot*, we follow the settings of Vinyals et al. (2016), Snell et al. (2017) and Chen et al. (2019) to augment the classes by rotations in 90, 180 and 270 degrees, resulting in 6492 classes in total. We also follow the settings of Snell et al. (2017) and Chen et al. (2019) to split these classes into 4112 classes for meta-training, 688 classes for meta-validation, and 1692 classes for meta-testing. Besides, all images are down-sampled to $28 \times 28 \times 1$ as a pre-processing step.

Another generic dataset used for object recognition is *mini-ImageNet* (Vinyals et al., 2016), a subset of *ImageNet* (Deng et al., 2009), and was firstly proposed by Vinyals et al. (2016) to investigate few-shot meta-learning problems. It contains 100 classes, and each class has 600 images. In this work, we follow the settings of recent work (e.g., Ravi and Larochelle, 2016; Chen et al., 2019). We randomly split the whole dataset into 64 classes for meta-training, 16 classes for meta-validation, 20 classes for meta-testing. We also test our method on two fine-grained image datasets, i.e., *CUB-200-2011* (Wah et al., 2011) and *Stanford-dogs* (Khosla et al., 2011). *CUB-200-2011* contains 11788 bird images from 200 classes. The whole dataset is randomly split into 100 classes for meta-training, 50 classes for meta-validation and 50 classes for meta-testing, following the settings of (Chen et al., 2019). *Stanford-dogs* contains 20580 dog images from 120 classes. We randomly split the dataset into three mutually disjoint subsets, 60 classes for meta-training, 30 classes for meta-validation and 30 classes for meta-testing. All images from *mini-ImageNet*, *CUB-200-2011* and *Stanford-dogs* are down-sampled to $84 \times 84 \times 3$ before being fed into neural networks, and standard data augmentation techniques are applied, i.e., random sized crop, random horizontal flip, and image jitter.

Setup All experiments are implemented through the episodic meta-training/meta-evaluation processes proposed by Vinyals et al. (2016). An episode actually refers to a task, i.e., a C-way K-shot classification problem. We split each datasets into three disjoint parts and set up three procedures, meta-training, meta-validation and meta-testing. We select the optimal number of meta-training epochs according to the classification accuracy on the meta-validation set. At the meta-testing stage, we randomly sample 600 novel tasks from the meta-testing set, and report the mean accuracy with its 95% confidence interval. We use *PyTorch* (Paszke et al., 2019) for all experiments[2].

5.2 Experimental Results

Comparisons to Probabilistic Meta-learning Methods In terms of the 5-shot experiments on *mini-ImageNet*, we achieve state-of-the-art accuracy for 5-way 5-shot tasks on *mini-ImageNet* with a lower variance, as shown in Table 1. For 1-shot classification on *mini-ImageNet*, the mean accuracy of our method is slightly lower than that of *BMAML* (Yoon et al., 2018), but it still falls in the 95% confidence interval of the performance of *BMAML*. Although experimental results show that the proposed method slightly de-

[2]See the supplementary material for detailed setting of hyper-parameters, e.g., T, D, α and β in Alg. 1.

Table 1: *Meta-testing Accuracy for* 5*-way Classification on Mini-ImageNet and Omniglot.* These methods all use a comparable feature embedding, i.e. shallow convolution networks (see the supplementary material for details). The result with $*$ is reported by Nguyen et al. (2020) as the original paper does not give the corresponding performance. Bold text indicates the highest mean accuracy and results that overlap with the confidence intervals of those highest mean accuracy.

	Omniglot (%)		*mini-ImageNet* (%)	
	5-way 1-shot	*5-way 5-shot*	*5-way 1-shot*	*5-way 5-shot*
BMAML (Yoon et al., 2018)	-	-	$\mathbf{53.80 \pm 1.46}$	$64.23 \pm 0.69^{*}$
PLATIPUS (Finn et al., 2018)	-	-	50.13 ± 1.86	-
ABML (Ravi and Beatson, 2019)	-	-	45.00 ± 0.60	-
Amortized VI (Gordon et al., 2019)	97.77 ± 0.55	98.71 ± 0.22	44.13 ± 1.78	55.68 ± 0.91
VERSA (Gordon et al., 2019)	$\mathbf{99.70 \pm 0.20}$	$\mathbf{99.75 \pm 0.13}$	$\mathbf{53.40 \pm 1.82}$	67.37 ± 0.86
Meta-Mixture (Jerfel et al., 2019)	-	-	51.20 ± 1.52	65.00 ± 0.96
VAMPIRE (Nguyen et al., 2020)	98.41 ± 0.19	$\mathbf{99.56 \pm 0.08}$	51.54 ± 0.74	64.31 ± 0.74
DKT (Patacchiola et al., 2020)	-	-	49.73 ± 0.07	64.00 ± 0.09
Ours	98.83 ± 0.17	$\mathbf{99.54 \pm 0.08}$	$\mathbf{53.28 \pm 0.91}$	$\mathbf{70.44 \pm 0.72}$

Table 2: *Meta-testing Accuracy for* 5*-way Classification on CUB-200-2011, Stanford-dogs and Mini-ImageNet.* These methods all use the same feature embedding architecture in (Chen et al., 2019), i.e. four convolution blocks. Results with superscript $*$ means training and testing locally.

	CUB-200-2011 (%)		*Stanford-dogs* (%)		*mini-ImageNet* (%)	
	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
MatchingNet (Vinyals et al., 2016)	60.52 ± 0.88	75.29 ± 0.75	$45.65 \pm 0.90^{*}$	$60.87 \pm 0.71^{*}$	48.14 ± 0.78	63.48 ± 0.66
ProtoNet (Snell et al., 2017)	50.46 ± 0.88	76.39 ± 0.64	$41.07 \pm 0.84^{*}$	$62.47 \pm 0.69^{*}$	44.42 ± 0.84	64.24 ± 0.72
RelationNet (Sung et al., 2018)	62.34 ± 0.94	77.84 ± 0.68	$47.20 \pm 0.89^{*}$	$66.12 \pm 0.71^{*}$	49.31 ± 0.85	66.60 ± 0.69
Baseline++ (Chen et al., 2019)	60.53 ± 0.83	79.34 ± 0.61	$44.15 \pm 0.71^{*}$	$64.42 \pm 0.66^{*}$	48.24 ± 0.75	66.43 ± 0.63
IMP (Allen et al., 2019)	$59.50 \pm 0.93^{*}$	$79.50 \pm 0.65^{*}$	$48.29 \pm 0.84^{*}$	$68.00 \pm 0.67^{*}$	49.60 ± 0.80	68.10 ± 0.80
MAML (Finn et al., 2017)	56.10 ± 1.01	75.41 ± 0.74	43.35 ± 0.85	60.55 ± 0.77	48.70 ± 1.84	63.11 ± 0.92
Ours	$\mathbf{63.46 \pm 0.98}$	$\mathbf{80.94 \pm 0.62}$	$\mathbf{54.45 \pm 0.94}$	$\mathbf{72.61 \pm 0.64}$	$\mathbf{53.28 \pm 0.91}$	$\mathbf{70.44 \pm 0.72}$

grades on *Omniglot*, it still achieves comparable results to those of recent probabilistic meta-learning methods.

Comparisons to Metric-based Methods Chen et al. (2019) provided fair comparisons of recent metric-based methods. Our neural network uses the same four convolution blocks as those in Chen et al. (2019), and our experiments are under the same settings as those in Chen et al. (2019). To make fair comparisons here, we use the results of *MatchingNet*, *ProtoNet* and *RelationNet* reported in Chen et al. (2019). As shown in Table 2, our method achieves state-of-the-art performance on all three datasets.

5.3 Ablation Studies

On Robustness Our proposed method enables us to have a luxury of varying the number of ways C or the number of shots K without re-training the network. To study the robustness of our method in these scenarios, we experiment on *Omniglot* and *mini-ImageNet*.

Omniglot. The proposed method is first trained for 5-way 5-shot or 10-way 5-shot on the meta-training set of *Omniglot*. Then, we vary the number of ways C (Fig. 2-a) or the number of shots K (Fig. 2-b) at meta-testing stage. As shown in Fig. 2-a, the proposed method still has a high mean accuracy above 96% even when $C = 50$ at meta-testing stage.

mini-ImageNet. We train/tune the proposed method for 5-way 5-shot on the meta-training/meta-validation set of *mini-ImageNet*, and test its performance for a higher C-way classification problem at meta-testing stage. In Table 3, the results show that our method compares favorably against other metric-based methods. For instance, without re-training, it preserves a high accuracy above 56% for 10-way 5-shot tasks.

On Effectiveness of Inference We investigate the effectiveness of our probabilistic inference, i.e., learning the posterior distributions of latent class prototypes for each task, by comparing with the removal of the $\mathbb{KL}$ term in Eq.4. In addition, since we do not use *dropout* (Kingma et al., 2015; Gal and Ghahramani, 2016) as an extra regularization (Gordon et al., 2019), we also investigate the effectiveness of *dropout*. Results of 5-way 5-shot classification on *Omniglot* and *mini-ImageNet* are presented in Fig. 2-c. In brief, our probabilistic inference plays a vital role in meta-learning as it boosts performance, e.g., $43.08 \pm 0.62\%$ versus $70.44 \pm 0.72\%$ on *mini-ImageNet*. Experiments also show that extra *dropout* is not effective to our method.

On Quality of Predictive Uncertainty The quality of predictive uncertainty of our model is measured by expected calibration error (ECE) and maxi-

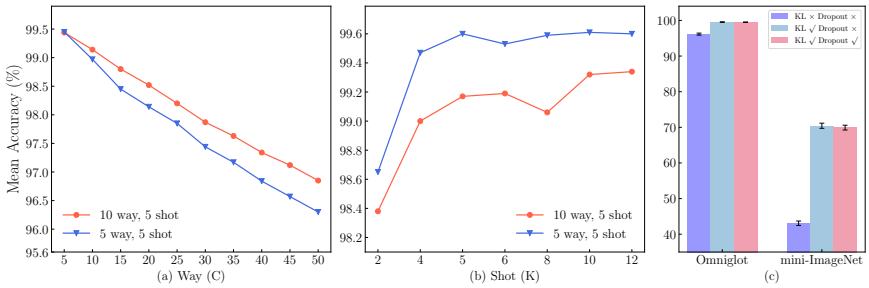

Figure 2: *Ablation Studies.* (a)&(b): Robustness of the proposed method on *Omniglot*. (c): Effectiveness of our probabilistic inference and extra dropout regularization. Details are presented in the supplementary material.

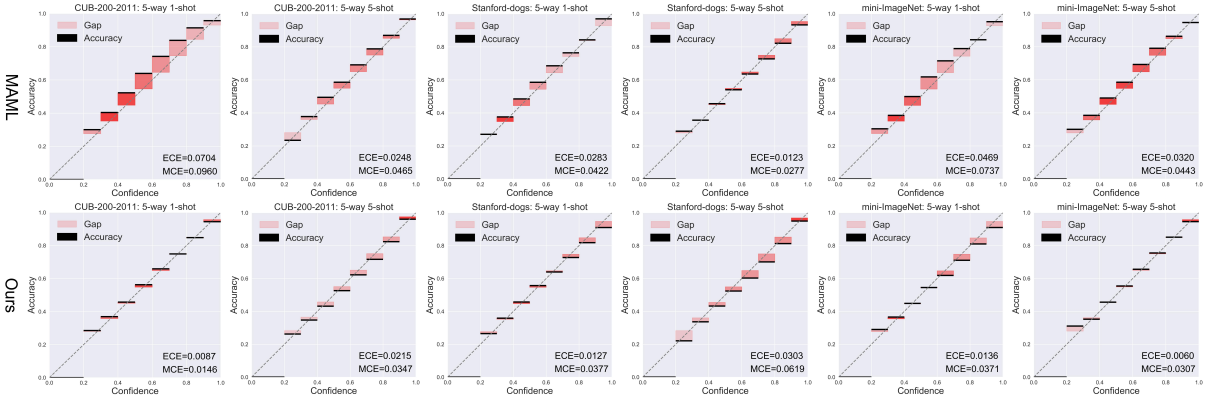

Figure 3: *Reliability Diagrams on Various Image Datasets.*

Table 3: *Ablation Study: Robustness on Mini-ImageNet.* All methods are trained for 5-way 5-shot at meta-training stage and are tested for C-way 5-shot at meta-testing stage.

C-way	$C = 5$ (%)	$C = 10$ (%)	$C = 20$ (%)
MatchingNet	63.48 ± 0.66	47.61 ± 0.44	33.97 ± 0.24
ProtoNet	64.24 ± 0.68	48.77 ± 0.45	34.58 ± 0.23
RelationNet	66.60 ± 0.69	47.77 ± 0.43	33.72 ± 0.22
Baseline++	66.43 ± 0.63	52.26 ± 0.40	38.03 ± 0.24
Ours	$\mathbf{70.44 \pm 0.72}$	$\mathbf{56.21 \pm 0.47}$	$\mathbf{43.43 \pm 0.25}$

mum calibration error (MCE) (Guo et al., 2017; Naeini et al., 2015), which are presented together with reliability diagrams in Fig. 3. A perfect calibration should have its predicting probabilities identical to the true correctness likelihood, i.e., $\Pr[\hat{y} = y|\hat{p} = p] = p$, where $\hat{y}$ and $\hat{p}$ are the model's prediction and its corresponding prediction confidence/probability, respectively, and $p \in [0, 1]$. This implies that a well calibrated model should have its bars close to the diagonal of reliability diagrams and have small values of MCE and ECE. In Fig. 3, it is shown that our method is well calibrated among various datasets and tasks.

6 Conclusions

In this paper, we propose a new, simple yet effective probabilistic meta-learning approach, *amortized Bayesian prototype meta-learning*. The proposed model can be trained end-to-end without any pre-training and learn to learn posterior distributions of latent prototypes whenever doing meta-training or meta-testing. Inference is amortized via learning a shared set of parameters such that a few steps of gradient descent can fast produce well-behaved approximate posteriors of latent prototypes. Randomness is considered through the learned posteriors of latent class prototypes, which results in excellent classification performance. With no need of extra amortization network, our method achieves state-of-the-art or competitive performance on various image datasets, e.g., *mini-ImageNet*, *CUB-200-2011* and *Stanford-dogs*. Ablation studies also demonstrate its the robustness, effectiveness and generalization ability.

Acknowledgements

This work was partly supported by the Engineering and Physical Sciences Research Council (EPSRC), U.K., under Grant EP/R513143/1, and the National Natural Science Foundation of China (NSFC) under Grant 61906080.

References

Allen, K., Shelhamer, E., Shin, H., and Tenenbaum, J. (2019). Infinite mixture prototypes for few-shot learning. In *International Conference on Machine Learning*, pages 232–241.

Amit, R. and Meir, R. (2018). Meta-learning by adjusting priors based on extended PAC-Bayes theory. In *International Conference on Machine Learning*, pages 205–214.

Bartunov, S. and Vetrov, D. (2018). Few-shot generative modelling with generative matching networks. In *International Conference on Artificial Intelligence and Statistics*, pages 670–678. PMLR.

Baxter, J. (2000). A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198.

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

Denevi, G., Ciliberto, C., Grazzi, R., and Pontil, M. (2019). Learning-to-learn stochastic gradient descent with biased regularization. In *International Conference on Machine Learning*, pages 1566–1575.

Denevi, G., Ciliberto, C., Stamos, D., and Pontil, M. (2018). Learning to learn around a common mean. In *Advances in Neural Information Processing Systems*, pages 10169–10179.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

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

Finn, C., Xu, K., and Levine, S. (2018). Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pages 9516–9527.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059.

Gordon, J., Bronskill, J., Bauer, M., Nowozin, S., and Turner, R. (2019). Meta-learning probabilistic inference for prediction. In *International Conference on Learning Representations*.

Grant, E., Finn, C., Levine, S., Darrell, T., and Griffiths, T. (2018). Recasting gradient-based meta-learning as hierarchical Bayes. In *International Conference on Learning Representations*.

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

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

Hospedales, T., Antoniou, A., Micaelli, P., and Storkey, A. (2020). Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*.

Iakovleva, E., Verbeek, J., and Alahari, K. (2020). Meta-learning with shared amortized variational inference. In *International Conference on Machine Learning*, pages 4572–4582. PMLR.

Jamal, M. A. and Qi, G.-J. (2019). Task agnostic meta-learning for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11719–11727.

Jerfel, G., Grant, E., Griffiths, T., and Heller, K. A. (2019). Reconciling meta-learning and continual learning with online mixtures of tasks. In *Advances in Neural Information Processing Systems*, pages 9119–9130.

Khosla, A., Jayadevaprakash, N., Yao, B., and Li, F.-F. (2011). Novel dataset for fine-grained image categorization: Stanford dogs. In *CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*.

Kingma, D. P., Salimans, T., and Welling, M. (2015). Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pages 2575–2583.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

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

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Li, H., Dong, W., Mei, X., Ma, C., Huang, F., and Hu, B.-G. (2019). LGM-Net: Learning to generate matching networks for few-shot learning. In *In-*

ternational Conference on Machine Learning, pages 3825–3834.

Li, X., Sun, Z., Xue, J.-H., and Ma, Z. (2020). A concise review of recent few-shot meta-learning methods. *arXiv preprint arXiv:2005.10953*.

Marino, J., Yue, Y., and Mandt, S. (2018). Iterative amortized inference. In *International Conference on Machine Learning*, pages 3403–3412.

Naeini, M. P., Cooper, G. F., and Hauskrecht, M. (2015). Obtaining well calibrated probabilities using Bayesian binning. In *AAAI Conference on Artificial Intelligence*, volume 2015, page 2901.

Nguyen, C., Do, T.-T., and Carneiro, G. (2020). Uncertainty in model-agnostic meta-learning using variational inference. In *IEEE Winter Conference on Applications of Computer Vision*, pages 3090–3100.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035.

Patacchiola, M., Turner, J., Crowley, E. J., O'Boyle, M., and Storkey, A. (2020). Bayesian meta-learning for the few-shot setting via deep kernels. *Advances in Neural Information Processing Systems*, 33.

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

Ravi, S. and Larochelle, H. (2016). Optimization as a model for few-shot learning. In *International Conference on Learning Representations*.

Ren, M., Liao, R., Fetaya, E., and Zemel, R. (2019). Incremental few-shot learning with attention attractor networks. In *Advances in Neural Information Processing Systems*, pages 5276–5286.

Rusu, A. A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., and Hadsell, R. (2019). Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.

Snell, J., Swersky, K., and Zemel, R. (2017). Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087.

Sun, Q., Liu, Y., Chua, T.-S., and Schiele, B. (2019). Meta-transfer learning for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 403–412.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., and Hospedales, T. M. (2018). Learning to compare: Relation network for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. (2016). Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. (2011). The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.

Wang, Y., Yao, Q., Kwok, J., and Ni, L. M. (2019). Generalizing from a few examples: A survey on few-shot learning. *arXiv: 1904.05046*.

Xu, J., Ton, J.-F., Kim, H., Kosiorek, A. R., and Teh, Y. W. (2020). Metafun: Meta-learning with iterative functional updates. In *International Conference on Machine Learning*, pages 10617–10627.

Yoon, J., Kim, T., Dia, O., Kim, S., Bengio, Y., and Ahn, S. (2018). Bayesian model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pages 7332–7342.

Supplementary Material for the Paper: Amortized Bayesian Prototype Meta-learning: A New Probabilistic Meta-learning Approach to Few-shot Image Classification

1 Overview

In this document, we present details of experimental settings, including hyper-parameters (batch size, learning rate, etc.). We also provide pseudo-code for meta-validation/meta-testing and detailed statistics in plots and figures. All experiments are implemented with PyTorch.

2 Pseudo-code for Meta-validation/Meta-testing

Algorithm 2 Meta-validation/Meta-testing of the proposed method

Require: Input Meta-trained model $\hat{\mathcal{M}}$. Set $\tilde{\mathcal{D}} = \mathcal{D}_{val}$ or $\mathcal{D}_{te}$.

for i from 1 to E **do**:
  Generate a task $\mathcal{T}_i = \mathcal{S}_i \cup \mathcal{Q}_i$ from $\tilde{\mathcal{D}}$.
  Initialize $\phi_i \leftarrow \theta$.
  for d from 1 to D **do**:
    Compute $q_{\phi_i}(z|\mathcal{S}_i)$.
    Approximate KL.
    Update variational parameters $\phi_i \leftarrow \phi_i - \alpha \nabla_{\phi_i}\{\mathcal{L}_{PR}(\mathcal{S}_i|z) + KL[q_{\phi_i}(z|\mathcal{S}_i) \,||\, p(z|\theta)]\}$.
  Predict for an image x: $\hat{y} = \arg\max_c \Pr(\mu(x)|q_{\phi_i}(z_c|\mathcal{S}_{i,c})),\ c \in [C]$.
  Compute prediction accuracy a_i for $\mathcal{Q}_i$.
Output mean accuracy $\frac{1}{E}\sum_{i=1}^{E} a_i$ as $\hat{\mathcal{M}}$'s performance.

3 Proofs

3.1 Proof for Eq.2

In this section, we provide a detailed derivation of the evidence lower bound of $\log p_\theta(\mathcal{S})$.

$$\log p_\theta(\mathcal{S}) \geq \mathbb{E}_{z\sim q_\phi(z)}[\log p_\theta(\mathcal{S}|z)] - \mathbb{KL}[q_\phi(z) \,||\, p_\theta(z)]$$

Proof:

$$\begin{aligned}
\log p_\theta(\mathcal{S}) &= \log \int p_\theta(\mathcal{S}, z) dz \\
&= \log \int p_\theta(\mathcal{S}, z) \frac{q_\phi(z)}{q_\phi(z)} dz \\
&= \log \mathbb{E}_q \left[\frac{p_\theta(\mathcal{S}, z)}{q_\phi(z)} \right] \\
&\geq \mathbb{E}_q \left[\log p_\theta(\mathcal{S}|z) + \log \frac{p_\theta(z)}{q_\phi(z)} \right] \quad , \text{ by Jensen's inequality} \\
&= \mathbb{E}_q \Big[\log p_\theta(\mathcal{S}|z) \Big] - \mathbb{KL} \Big[q_\phi(z) || p_\theta(z) \Big]
\end{aligned}$$

3.2 Unbiased Estimator Scaled by A Constant

Although we replace the $\mathbb{KL}$ term in the evidence lower bound of $\log p_\theta(\mathcal{S})$ with Eq.7 as our proposed prior distribution of z is now dependent on the support set $\mathcal{S}$, our estimator to Eq.4 is still an unbiased estimator to evidence lower bound of $\log p_\theta(\mathcal{S})$ in Eq.2 (scaled by a constant). Therefore the proposed method still learns to learn the approximate posteriors of latent z conditional on $\mathcal{S}$ properly. To appreciate this, note that during the inference stage τ is the support set $\mathcal{S}$ (and we have $|\tau| = CK$). Then, after putting the unbiased estimator of Eq.7 and Eq.9 into Eq.4, we can rewrite the loss in Eq.4 as

$$\mathcal{L}(\mathcal{S}) = \frac{1}{CK}\sum_{c=1}^{C}\sum_{i=1}^{K}\Big(-\log\Big(\frac{\Pr\left[\mu(x_i^{(\mathcal{S}_c)})|z_c\right]}{\sum_{c=1}^{C}\Pr\left[\mu(x_i^{(\mathcal{S}_c)})|z_c\right]}\Big) + \mathbb{KL}[q_\phi(z_c|\mathcal{S}_c)||p_\theta(z_c;\mu(x_i^{(\mathcal{S}_c)}),\Sigma(x_i^{(\mathcal{S}_c)}))]\Big)$$

, where $\mathcal{S}_c$ is the subset of $\mathcal{S}$ and only contains all support images from the class $c \in \{1,\ldots,C\}$, and $(x_i^{(\mathcal{S}_c)}, y_i^{(\mathcal{S}_c)} = c)$ is the i^{th} image in $\mathcal{S}_c$. This immediately tells that $-\mathcal{L}(\mathcal{S}) = \frac{1}{CK}\sum_{c=1}^{C}\sum_{i=1}^{K}(\log p_\theta(y_i^{(\mathcal{S}_c)}|x_i^{(\mathcal{S}_c)}, z_c) - \mathbb{KL}[q_\phi(z_c|\mathcal{S}_c)||p_\theta(z_c;\mu(x_i^{(\mathcal{S}_c)}),\Sigma(x_i^{(\mathcal{S}_c)}))])$, where the terms inside the double summation is an unbiased estimator of the evidence lower bound of $\log p_\theta(y_i^{(\mathcal{S}_c)}|x_i^{(\mathcal{S}_c)})$. Since $\frac{1}{CK}\sum_{c,i}\log p_\theta(y_i^{(\mathcal{S}_c)}|x_i^{(\mathcal{S}_c)}) = \frac{1}{CK}\log p_\theta(\mathcal{S})$, it tells that $-\mathcal{L}(\mathcal{S})$ is an unbiased estimator of the evidence lower bound of $\log p_\theta(\mathcal{S})$ scaled by a factor of $1/CK$.

4 Experimental Details

At the meta-training stage, except that the maximum training epoch is 12000 for 1-shot classification on *mini-ImageNet*, the maximum training epoch is set to be 3500 epochs for all the other experiments. We use a mini-batch of tasks consisting T tasks to update the shared θ during meta-training.

We select the optimal meta-training epoch on the meta-validation set according to classification accuracy. At the meta-testing stage, we randomly sample 600 novel tasks from the meta-testing set, and report the mean accuracy with its 95% confidence interval, i.e., mean acc. $\pm\, 1.96\frac{\text{std}}{\sqrt{600}}$. For C-way K-shot, a task is constructed by sampling C classes and then subsequently sampling $K+M$ instances for each class, with K being the number of support images in each class. In our experiments,

- *Omniglot*: $M = 15$ for meta-training/meta-validation/meta-testing;
- *mini-ImageNet*: $M = 16$ for meta-training and meta-validation, $M = 15$ for meta-testing;
- *CUB-200-2011*: $M = 16$ for meta-training and meta-validation, $M = 15$ for meta-testing;
- *Stanford-dogs*: $M = 16$ for meta-training and meta-validation, $M = 15$ for meta-testing.

The values of T, D, α and β in **Alg. 1** and **Alg. 2** are set to be

- *Omniglot*: $T = 32$, $D = 1$, $\alpha = 0.1$, $\beta = 0.001$;
- *mini-ImageNet*: $T = 4$, $D = 5$, $\alpha = 0.01$, $\beta = 0.001$;
- *CUB-200-2011*: $T = 4$, $D = 5$, $\alpha = 0.01$, $\beta = 0.001$;
- *Stanford-dogs*: $T = 4$, $D = 5$, $\alpha = 0.01$, $\beta = 0.001$.

In addition, we use standard stochastic gradient descent to generate variational parameters ϕ_i, during meta-training/meta-validation/meta-testing, for a task $\mathcal{T}_i$ and for all i. We use the *Adam* optimizer to update the shared parameter θ at meta-training stage.

5 Details of Figures

In this section, we present detailed statistics in **Fig. 2**.

Ablation study in **Fig.2-a**.

	Meta-training conditions	
C-way at meta-testing	5-way 5-shot (%)	10-way 5-shot (%)
$C = 5$	99.45 ± 0.09	99.44 ± 0.08
$C = 10$	98.97 ± 0.08	99.14 ± 0.08
$C = 15$	98.45 ± 0.09	98.80 ± 0.09
$C = 20$	98.14 ± 0.09	98.52 ± 0.08
$C = 25$	97.85 ± 0.09	98.20 ± 0.08
$C = 30$	97.44 ± 0.09	97.87 ± 0.08
$C = 35$	97.17 ± 0.09	97.63 ± 0.08
$C = 40$	96.84 ± 0.08	97.34 ± 0.08
$C = 45$	96.57 ± 0.08	97.12 ± 0.08
$C = 50$	96.30 ± 0.08	96.85 ± 0.08

Ablation study in **Fig.2-b**.

	Meta-training conditions	
K-shot at meta-testing	5-way 5-shot (%)	10-way 5-shot (%)
$K = 2$	98.65 ± 0.27	98.38 ± 0.15
$K = 4$	99.47 ± 0.11	99.00 ± 0.11
$K = 5$	99.60 ± 0.10	99.17 ± 0.09
$K = 6$	99.53 ± 0.11	99.19 ± 0.10
$K = 8$	99.59 ± 0.10	99.06 ± 0.13
$K = 10$	99.61 ± 0.09	99.32 ± 0.10
$K = 12$	99.60 ± 0.09	99.34 ± 0.09

- *Omniglot*: *Dropout* with a keep probability of 0.9.
- *mini-ImageNet*: *Dropout* with a keep probability of 0.5.

Ablation study in **Fig.2-c**.

KL	*Dropout*	*Omniglot* (%)	*mini-ImageNet* (%)
-	-	96.16 ± 0.28	43.08 ± 0.62
✓	-	99.54 ± 0.08	70.44 ± 0.72
✓	✓	99.50 ± 0.08	69.92 ± 0.67

6 Comparisons of Convolution Networks

Here, we present details of shallow convolution networks used in the probabilistic meta-learning methods listed in **Table 1**. CONV-X means a convolution network with X convolution blocks.

Convolution networks of methods in **Table 1**.

	Omniglot	*mini-ImageNet*
BMAML	CONV-5	CONV-5
PLATIPUS	CONV-4	CONV-4
VAMPIRE	CONV-4	CONV-4
ABML	CONV-4	CONV-4
Amortized VI	CONV-4	CONV-5
VERSA	CONV-4	CONV-5
Meta-Mixture	CONV-4	CONV-4
DKT	CONV-4	CONV-4
Ours	CONV-4	CONV-4

7 Effect of D

We also take the effect of D into account. Recall that D is the number of updates of the inner loop for the approximate inference. We consider the cases when $D = 1$, $D = 3$ and $D = 5$. Performance for each choice of D is measured on the meta-testing set.

Effect of D.

mini-ImageNet	$D = 1(\%)$	$D = 3(\%)$	$D = 5(\%)$
5-way 1-shot	52.79 ± 0.94	53.29 ± 0.89	53.28 ± 0.91
5-way 5-shot	69.63 ± 0.70	70.56 ± 0.70	70.44 ± 0.72