PAC-BAYES FEW-SHOT META-LEARNING WITH IMPLICIT LEARNING OF MODEL PRIOR DISTRIBUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce a new and rigorously-formulated PAC-Bayes few-shot meta-learning algorithm that implicitly learns a model prior distribution of interest. Our proposed method extends the PAC-Bayes framework from a single task setting to the few-shot meta-learning setting to upper-bound generalisation errors on unseen tasks. We also propose a generative-based approach to model the shared prior and task-specific posterior more expressively compared to the usual diagonal Gaussian assumption. We show that the models trained with our proposed meta-learning algorithm are well calibrated and accurate, with state-of-the-art calibration and classification results on mini-ImageNet benchmark, and competitive results in a multi-modal task-distribution regression.

1 INTRODUCTION

One unique ability of humans is to be able to quickly learn new tasks with only a few *training* examples. This is due to the fact that humans tend to exploit prior experience to facilitate the learning of new tasks. Such exploitation is markedly different from conventional machine learning approaches, where no prior knowledge (e.g. training from scratch with random initialisation) (Glorot & Bengio, 2010), or weak prior knowledge (e.g., fine tuning from pre-trained models) (Rosenstein et al., 2005) are used when encountering an unseen task for training. This motivates the development of novel learning algorithms that can effectively encode the knowledge learnt from training tasks, and exploit that knowledge to quickly adapt to future tasks (Lake et al., 2015).

Prior knowledge can be helpful for future learning only if all tasks are assumed to be distributed according to a latent task distribution. Learning this latent distribution is, therefore, useful for solving an unseen task, even if the task contains a limited number of training samples. Many approaches have been proposed and developed to achieve this goal, namely: *multi-task learning* (Caruana, 1997), *domain adaptation* (Bridle & Cox, 1991; Ben-David et al., 2010) and *meta-learning* (Schmidhuber, 1987; Thrun & Pratt, 1998). Among these, meta-learning has flourished as one of the most effective methods due to its ability to leverage the knowledge learnt from many training tasks to quickly adapt to unseen tasks.

Recent advances in meta-learning have produced state-of-the-art results in many benchmarks of few-shot learning data sets (Santoro et al., 2016; Ravi & Larochelle, 2017; Munkhdalai & Yu, 2017; Snell et al., 2017; Finn et al., 2017; Zhang et al., 2018; Rusu et al., 2019). Learning from a few examples is often difficult and easily leads to over-fitting, especially when no model uncertainty is taken into account. This issue has been addressed by several recent Bayesian meta-learning approaches that incorporate model uncertainty into prediction, notably LLAMA that is based on Laplace method (Grant et al., 2018), or PLATIPUS (Finn et al., 2017), Amortised Meta-learner (Ravi & Beatson, 2019) and VERSA (Gordon et al., 2019) that use variational inference (VI). However, these works have not thoroughly investigated the generalisation errors for unseen samples, resulting in limited theoretical generalisation guarantees. Moreover, most of these papers are based on variational functions that may not represent well the richness of the underlying distributions. For instance, a common choice for the variational function relies on the diagonal Gaussian distribution, which can potentially worsen the prediction accuracy given its limited representability.

In this paper, we address the two problems listed above with the following technical novelties: *(i)* derivation of a rigorous upper-bound for the generalisation errors of few-shot meta-learning using PAC-Bayes framework, and *(ii)* proposal of a novel variational Bayesian learning based on implicit

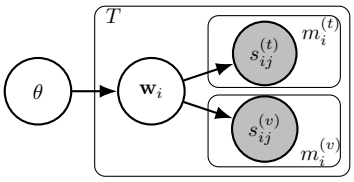


Figure 1: The few-shot meta-learning problem is modelled using a hierarchical model that learns a prior $p(\mathbf{w}_i; \theta)$ using a few data points $s_{ij}^{(t)} = \{(x_{ij}^{(t)}, y_{ij}^{(t)})\}$ to infer $s_{ij}^{(v)} = \{(x_{ij}^{(v)}, y_{ij}^{(v)})\}$. Shaded nodes denote observed variables, while white nodes denote hidden variables.

generative models to facilitate the learning of unseen tasks. Our evaluation shows that the models trained with our proposed meta-learning algorithm is at the same time well calibrated and accurate, with competitive results in terms of Expected Calibration Error (ECE) and Maximum Calibration Error (MCE), while outperforming state-of-the-art methods in a few-shot classification benchmark (mini-ImageNet).

2 Related Work

Our paper is related to Bayesian few-shot meta-learning techniques that have been developed to incorporate uncertainty into model estimation. LLAMA (Grant et al., 2018) employs the Laplace method to extend the deterministic estimation assumed in MAML to a Gaussian distribution. However, the need to estimate and invert the Hessian matrix makes this approach computationally challenging for large-scale models, such as deep neural networks. Variational inference (VI) addresses such scalability issue – remarkable examples of VI-based methods are PLATIPUS (Finn et al., 2018), BMAML (Yoon et al., 2018), Amortised meta-learner (Ravi & Beatson, 2019) and VERSA (Gordon et al., 2019). Although these VI-based approaches have demonstrated impressive results in regression, classification as well as reinforcement learning, they do not provide any theoretical guarantee on generalisation errors for unseen samples within a task. Moreover, the overly-simplified family of diagonal Gaussian distributions used in most of these works limits the expressiveness of the variational approximation, resulting in a less accurate prediction.

Our work is also related to the PAC-Bayes framework used in multi-task learning (Pentina & Lampert, 2014; Amit & Meir, 2018) that provides generalisation error bounds with certain confidence levels. These previously published papers jointly learn a single shared prior and many task-specific posteriors without relating the shared prior to any task-specific posterior. Hence, these approaches need to store all task-specific posteriors, resulting in un-scalable solutions, especially when the number of tasks is large. In contrast, our proposed method learns only the shared prior of model parameters and uses that prior to estimate the task-specific posterior through the likelihood function by performing a fixed number of gradient updates. This proposed variant of amortised inference allows a memory efficient solution, and therefore, more favourable for applications with large number of tasks, such as few-shot meta-learning.

3 Methodology

In this section, we first define and formulate the few-shot meta-learning problem. Subsequently, we derive the generalisation upper-bound based on PAC-Bayes framework. We then present our proposed approach that employs implicit variational distributions for few-shot meta-learning.

3.1 Few-shot meta-learning

We use the notation of *task environment* (Baxter, 2000) to describe the unknown distribution $p(\mathcal{T})$ over a family of tasks, from where tasks are sampled. Each task $\mathcal{T}_i$ in this family is indexed by $i \in \{1, ..., T\}$ and associated with a dataset $\{\mathcal{X}_i, \mathcal{Y}_i\}$ consisting of a training/support set $\{\mathcal{X}_i^{(t)}, \mathcal{Y}_i^{(t)}\}$ and a validation/query set $\{\mathcal{X}_i^{(v)}, \mathcal{Y}_i^{(v)}\}$, with $\mathcal{X}_i^{(t)} = \{\mathbf{x}_{ij}^{(t)}\}_{j=1}^{m_i^{(t)}}$ and $\mathcal{Y}_i^{(t)} = \{y_{ij}^{(t)}\}_{j=1}^{m_i^{(t)}}$ ($\mathcal{X}_i^{(v)}$ and $\mathcal{Y}_i^{(v)}$ are similarly defined with $m_i^{(v)}$ samples). The aim of few-shot learning is to accurately predict

the output $y_{ij}^{(v)}$ of the query input $\mathbf{x}_{ij}^{(v)}$ given the small support set for task $\mathcal{T}_i$. We rely on a Bayesian hierarchical model (Grant et al., 2018) as shown in Figure 1, where $\mathbf{w}_i$ represents the model parameters for task $\mathcal{T}_i$, and θ denotes the meta-parameters shared across all tasks. For example, in MAML (Finn et al., 2017), $\mathbf{w}_i$ are the neural network weights for task $\mathcal{T}_i$ that is initialised from θ and obtained by performing truncated gradient descent using $\{\mathcal{X}_i^{(t)}, \mathcal{Y}_i^{(t)}\}$.

While the conventional graphical model methods in meta-learning learn the joint probability $p(\theta, \mathbf{w}_{1:T}|\mathcal{Y}_{1:T}, \mathcal{X}_{1:T})$ (Amit & Meir, 2018, Section A.3), our objective function for the few-shot meta-learning is to minimise the negative log predictive probability w.r.t. the meta-parameters θ as follows:

$$\theta^* = \arg\min_{\theta} \frac{1}{T} \sum_{i=1}^{T} \mathbb{E}_{y_{ij} \sim \mathcal{Y}_i^{(v)}} \left[-\log p(y_{ij}^{(v)}|\mathcal{Y}_i^{(t)}; \theta) \right], \tag{1}$$

where we simplify the notation by dropping the explicit dependence on $\mathcal{X}_i^{(t)}$ and $\mathcal{X}_i^{(v)}$ from the set of conditioning variables (this simplification is adopted throughout the paper). The predictive probability term inside the expectation in (1) can be expanded by applying the sum rule of probability and lower-bounded by Jensen's inequality:

$$\log p(y_{ij}^{(v)}|\mathcal{Y}_i^{(t)}; \theta) = \log \int p(y_{ij}^{(v)}|\mathbf{w}_i) p(\mathbf{w}_i|\mathcal{Y}_i^{(t)}; \theta) d\mathbf{w}_i \geq \mathbb{E}_{p(\mathbf{w}_i|\mathcal{Y}_i^{(t)}; \theta)} \left[\log p(y_{ij}^{(v)}|\mathbf{w}_i) \right]. \tag{2}$$

In practice, the task-specific posterior $p(\mathbf{w}_i|\mathcal{Y}_i^{(t)}; \theta)$ is often intractable, and therefore, approximated by a distribution $q(\mathbf{w}_i; \lambda_i)$ parameterised by $\lambda_i = \lambda_i(\mathcal{Y}_i^{(t)}, \theta)$. Given this assumption and the result in (2), the upper bound of the objective function in (1) can be presented as:

$$\mathcal{L}^{(v)}(\theta) = \frac{1}{T} \sum_{i=1}^{T} \mathcal{L}_i^{(v)}(\theta), \tag{3}$$

where:

$$\mathcal{L}_i^{(v)}(\theta) = \mathbb{E}_{y_{ij}^{(v)} \sim \mathcal{Y}_i^{(v)}} \mathbb{E}_{q(\mathbf{w}_i; \lambda_i)} \left[-\log p(y_{ij}^{(v)}|\mathbf{w}_i) \right]. \tag{4}$$

Hence, instead of minimising the objective function in (1), we minimise the upper bound in (3). There are two issues related to the optimisation of this upper bound: *(i)* the generalisation error for $(x_{ij}^{(v)}, y_{ij}^{(v)})$ sampled from the true query set $\{\mathcal{X}_i^{(v)}, \mathcal{Y}_i^{(v)}\}$, and *(ii)* how to estimate $q(\mathbf{w}_i; \lambda_i)$ that can approximate the true posterior $p(\mathbf{w}_i|\mathcal{Y}_i^{(t)}; \theta)$ accurately, so that we can evaluate and minimise the upper-bound in (3). We address the generalisation error in Section 3.2 and present a variational method to obtain an expressive variational posterior $q(\mathbf{w}_i; \lambda_i)$ in Section 3.3.

3.2 PAC-Bayes generalisation bound

We first introduce the PAC-Bayes bound for the single-task problem in Theorem 1.

Theorem 1 (PAC-Bayes bound for single-task setting (McAllester, 1999)). *Let $\mathcal{D}$ be an arbitrary distribution over an example domain $\mathcal{Z}$. Let $\mathcal{H}$ be a hypothesis class, $\ell : \mathcal{H} \times \mathcal{Z} \to [0, 1]$ be a loss function, π be a prior distribution over $\mathcal{H}$, and $\delta \in (0, 1)$. If $S = \{z_j\}_{j=1}^m$ is an i.i.d. training set sampled according to $\mathcal{D}$, then for any "posterior" Q over $\mathcal{H}$, the following holds:*

$$p\left(\mathbb{E}_{z_j \sim p(z)} \mathbb{E}_{q \sim Q} \ell(q, z_j) \leq \mathbb{E}_{z_j \sim \mathcal{S}} \mathbb{E}_{q \sim Q} \ell(q, z_j) + \sqrt{\frac{D_{\mathrm{KL}}\left[Q \| \pi\right] + \log \frac{m}{\delta}}{2(m-1)}} \right) \geq 1 - \delta,$$

where

$$D_{\mathrm{KL}}\left[Q \| \pi\right] = \mathbb{E}_{q \sim Q} \left[\log \frac{Q(q)}{\pi(q)} \right]$$

is the Kullback-Leibler divergence.

Theorem 1 indicates that with a high probability, the expected error of an arbitrary posterior Q on data distribution $p(z)$ is upper-bounded by the empirical error plus a complexity regularisation term. These two terms express the trade-off between fitting data (bias) and regularising model complexity (variance).

Remark 1. *Despite the assumption based on bounded loss function, the PAC-Bayes bound can also be extended to unbounded loss function (McAllester, 1999, Section 5).*

Before presenting the novel bound for few-shot meta-learning, we define some notations. Recall that $m_i^{(v)}$ is the number of samples in the query set $\{\mathcal{X}_i^{(v)}, \mathcal{Y}_i^{(v)}\}$. Let $\delta \in (0, 1)$ and

$$\hat{\mathcal{L}}_i^{(v)}(\theta) = \frac{\sum_{j=1}^{m_i^{(v)}} \mathbb{E}_{q(\mathbf{w}_i;\lambda_i)} \left[-\log p(y_{ij}|\mathbf{w}_i)\right]}{m_i} + \sqrt{\frac{D_{\mathrm{KL}}\left[q(\mathbf{w}_i;\lambda_i)\|p(\mathbf{w}_i;\theta)\right] + \log \frac{m_i T}{\delta}}{2(m_i - 1)}}. \tag{5}$$

The novel bound on the generalisation error for the few-shot meta-learning problem is shown in Theorem 2. Please refer to Appendix A for the proof.

Theorem 2 (PAC-Bayes bound for few-shot meta-learning in (3))**.** *For the general error of few-shot meta-learning in (3), the following holds:*

$$p\left(\mathcal{L}^{(v)}(\theta) \leq \frac{1}{T}\sum_{i=1}^{T} \hat{\mathcal{L}}_i^{(v)}(\theta)\right) \geq 1 - \delta.$$

Remark 2. *The result derived in Theorem 2 is different from the one in (Amit & Meir, 2018, Theorem 2). As mentioned in Section 2, the prior work (Amit & Meir, 2018) does not relate the posterior of model parameters $q(\mathbf{w}_i;\lambda_i)$ to the shared prior $p(\mathbf{w}_i;\theta)$. The "hypothesis" in that case is a tuple including the model parameters sampled from the prior and task-specific posterior. In contrast, our approach is a variant of amortised inference that relates the posterior from the prior and likelihood function by gradient updates (see Section 3.3). Hence, the "hypothesis" in our case includes the parameters sampled from the task-specific posterior only. The discrepancy of the "hypothesis" used between the two approaches results in different upper-bounds, particularly at the regularisation term.*

Given the result in Theorem 2, the objective function of interest is to minimise the generalisation upper-bound:

$$\hat{\mathcal{L}}^{(v)}(\theta) = \frac{1}{T}\sum_{i=1}^{T} \hat{\mathcal{L}}_i^{(v)}(\theta). \tag{6}$$

3.3 Gradient-based variational inference

As denoted in Section 3.1, $q(\mathbf{w}_i;\lambda_i)$ is a variational posterior that approximates the true posterior $p(\mathbf{w}_i|\mathcal{Y}_i^{(t)};\theta)$ for task $\mathcal{T}_i$, and therefore, can be obtained by minimising the following KL divergence:

$$\begin{aligned}
\lambda_i^* &= \arg\min_{\lambda_i} D_{\mathrm{KL}}\left[q(\mathbf{w}_i;\lambda_i)\|p(\mathbf{w}_i|\mathcal{Y}_i^{(t)};\theta)\right] \\
&= \arg\min_{\lambda_i} D_{\mathrm{KL}}\left[q(\mathbf{w}_i;\lambda_i)\|p(\mathbf{w}_i;\theta)\right] - \mathbb{E}_{q(\mathbf{w}_i;\lambda_i)}\left[\ln p(\mathcal{Y}_i^{(t)}|\mathbf{w}_i)\right] + \underbrace{\ln p(\mathcal{Y}^{(t)}|\theta)}_{\text{const. wrt } \lambda_i}.
\end{aligned} \tag{7}$$

The resulting cost function (excluding the constant term) in (7) is often known as the variational free energy (VFE). For simplicity, we denote the cost function as

$$\mathcal{L}_i^{(t)}\left(\mathcal{Y}_i^{(t)}, \lambda_i, \theta\right) = D_{\mathrm{KL}}\left[q(\mathbf{w}_i;\lambda_i)\|p(\mathbf{w}_i;\theta)\right] + \mathbb{E}_{q(\mathbf{w}_i;\lambda_i)}\left[-\ln p\left(\mathcal{Y}_i^{(t)}|\mathbf{w}_i\right)\right]. \tag{8}$$

The first term of VFE can be considered as a regularisation that penalises the difference between the shared prior $p(\mathbf{w}_i;\theta)$ and the variational task-specific posterior $q(\mathbf{w}_i;\lambda_i)$, while the second term is referred as data-dependent or likelihood cost. Exactly minimising the cost function in (8) is computationally challenging, so gradient descent is used with θ as the initialisation:

$$\lambda_i \leftarrow \theta - \alpha_t \nabla_{\lambda_i} \mathcal{L}_i^{(t)}\left(\mathcal{Y}_i^{(t)}, \lambda_i; \theta\right), \tag{9}$$

where α_t is the learning rate and the truncated gradient descent consists of a single step (the extension to a larger number of steps is trivial). Given the approximated posterior $q(\mathbf{w}_i;\lambda_i)$ with

parameter λ_i obtained from (9), we can calculate and optimise the generalisation upper bound in (6) w.r.t. θ.

In Bayesian statistics, the shared prior $p(\mathbf{w}_i;\theta)$ represents a modelling assumption, and the variational task-specific posterior $q(\mathbf{w}_i;\lambda_i)$ is a flexible function that can be adjusted to achieve a good trade-off between performance and complexity. In general, $p(\mathbf{w}_i;\theta)$ and $q(\mathbf{w}_i;\lambda_i)$ can be modelled using two general types of probabilistic models: prescribed and implicit (Diggle & Gratton, 1984). For example, Amortised Meta-learner (Ravi & Beatson, 2019) is a prescribed approach where both distributions are assumed to be diagonal Gaussians. In this paper, we present a more expressive way of implicitly modelling the shared prior and task-specific posterior.

Both distributions $p(\mathbf{w}_i;\theta)$ and $q(\mathbf{w}_i;\lambda_i)$ are now defined at a more fundamental level whereby data is generated through a stochastic mechanism without specifying parametric distributions. We use a parameterised model (i.e., a generator G represented by a deep neural network) to model the sample generation from the prior and posterior:

$$\begin{cases} \mathbf{w}_i \sim p(\mathbf{w}_i;\theta) & \Leftrightarrow \quad \mathbf{w}_i = G(\mathbf{z};\theta), \mathbf{z} \sim p(\mathbf{z}) \\ \mathbf{w}_i \sim q(\mathbf{w}_i;\lambda_i) & \Leftrightarrow \quad \mathbf{w}_i = G(\mathbf{z};\lambda_i), \mathbf{z} \sim p(\mathbf{z}), \end{cases} \tag{10}$$

where $p(\mathbf{z})$ is usually denoted by a Gaussian model $\mathcal{N}(0,\mathbf{I})$ or a uniform model $\mathcal{U}(0,1)$.

Due to the nature of implicit models, the KL divergence term in (8), in particular the density ratio $q(\mathbf{w}_i;\lambda_i)/p(\mathbf{w}_i;\theta)$, cannot be evaluated either analytically or symbolically. We, therefore, propose to employ the *probabilistic classification* approach (Sugiyama et al., 2012, Chapter 4) to estimate the KL divergence term. We use a parameterised model – a discriminator D represented by a deep neural network – as a classifier to distinguish different $\mathbf{w}_i$ sampled from the prior $p(\mathbf{w}_i;\theta)$ (label 1) or the posterior $q(\mathbf{w}_i;\lambda_i)$ (label 0). The objective function to train the discriminator D can be written as:

$$\max_{\omega_i} \mathcal{L}_D(\omega_i) = \max_{\omega_i} \mathbb{E}_{p(\mathbf{z})}\left[\ln D\left(G(\mathbf{z};\theta);\omega_i\right)\right] + \mathbb{E}_{p(\mathbf{z})}\left[\ln\left(1 - D\left(G(\mathbf{z};\lambda_i);\omega_i\right)\right)\right], \tag{11}$$

where ω_i is the parameters of D for task $\mathcal{T}_i$.

Given the discriminator D, the KL divergence term in (8) can be estimated as:

$$D_{\mathrm{KL}}\left[q(\mathbf{w}_i;\lambda_i)\|p(\mathbf{w}_i;\theta)\right] = \mathbb{E}_{q(\mathbf{w}_i;\lambda_i)}\left[\ln \frac{q(\mathbf{w}_i;\lambda_i)}{p(\mathbf{w}_i;\theta)}\right] \approx -\frac{1}{L_t}\sum_{l=1}^{L_t} V\left(G(\mathbf{z}^{(l)};\lambda_i);\omega_i\right), \tag{12}$$

where $\mathbf{z}^{(l)} \sim p(\mathbf{z})$, L_t is the number of Monte Carlo samples, and $V(.,\omega_i)$ is the output of the discriminator D without sigmoid activation.

The variational-free energy in (8) can, therefore, be rewritten as:

$$\mathcal{L}_i^{(t)}(\lambda_i,\omega_i,\theta) \approx -\frac{1}{L_t}\sum_{l=1}^{L_t}\left[V\left(G(\mathbf{z}^{(l)};\lambda_i);\omega_i\right) + \ln p\left(\mathcal{Y}_i^{(t)}\middle|G(\mathbf{z}^{(l)};\lambda_i)\right)\right]. \tag{13}$$

One problem that arises when estimating the loss in (13) is how to obtain the local optimal parameters ω_i^* for the discriminator D. One simple approach is to generate several model parameters $\mathbf{w}_i$ from the prior $p(\mathbf{w}_i;\theta)$ and posterior $q(\mathbf{w}_i;\lambda_i)$ following (10) to train $D(.;\omega_i)$ by optimising the cost in (11). The downside is the significant increase in training time and memory usage to store the computational graph to later be used for minimising the upper-bound in (6) w.r.t. θ. To overcome this limitation, we propose to meta-learn ω_i using MAML (Finn et al., 2017). In this scenario, we define ω_0 as the meta-parameters (or initialisation) of ω_i. Within each task, we initialise ω_i at ω_0 and use the generated $\mathbf{w}_i$ from (10) as training data. This approach leads to our proposed algorithm, named Statistical Implicit Bayesian Meta-Learning (SImBa), shown in Algorithm 1.

Our assumption here is that the discriminator can provide an optimal estimate of the KL divergence term as shown in (12). This strong theoretical property only holds when the discriminator model is correctly-specified (Sugiyama et al., 2012, Remark 4.7). To this end, we employ the universal approximation theorem (Cybenko, 1989; Hornik, 1991) to model the discriminator as a feed-forward fully connected neural network. We expect that under this modelling approach, the discriminator model is approximately correctly-specified.

Algorithm 1 SImBa

Input: task distribution $p(\mathcal{T})$ and hyper-parameters: $T, L_t, L_v, L_D, \alpha_t, \alpha_v, \gamma_t, \gamma_v, \delta$
Output: meta-parameters θ of the shared prior $p(\mathbf{w}_i; \theta)$, and discriminator meta-parameters ω_0

initialise θ and ω_0
while θ not converged **do**
  sample a mini-batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$, where $i = 1 : T$
  for each task $\mathcal{T}_i$ **do**
    $\lambda_i \leftarrow \theta$
    $\omega_i \leftarrow \omega_0$
    generate $\mathbf{w}_{i,j}^{(p)} = G(\mathbf{z}; \theta)$, and $\mathbf{w}_{i,j}^{(q)} = G(\mathbf{z}; \lambda_i)$, where $j = 1 : L_D$
    update $\omega_i \leftarrow \omega_i + \gamma_t \nabla_{\omega_i} \mathcal{L}_D(\omega_i)$ {Eq. (11)}
    generate $\hat{\mathbf{w}}_i^{(l_t)} = G(\mathbf{z}^{(l_t)}; \lambda_i), l_t = 1 : L_t$ to calculate VFE
    update: $\lambda_i \leftarrow \lambda_i - \alpha_t \nabla_{\lambda_i} \mathcal{L}_i^{(t)}(\lambda_i, \omega_i, \theta)$ {Eq. (13)}
    generate $\hat{\mathbf{w}}_i^{(l_v)} = G(\mathbf{z}^{(l_v)}; \lambda_i), l_v = 1 : L_v$
    compute $\hat{\mathcal{L}}_i^{(v)}(\theta)$ {Eq. (5)}
    repeat step 7 to calculate discriminator loss $\mathcal{L}_D(\omega_i)$
  end for
  update $\theta \leftarrow \theta - \frac{\alpha_v}{T} \nabla_\theta \sum_{i=1}^{T} \hat{\mathcal{L}}_i^{(v)}(\theta)$ {Eq. (6)}
  update $\omega_0 \leftarrow \omega_0 + \frac{\gamma_v}{T} \nabla_{\omega_0} \sum_{i=1}^{T} \mathcal{L}_D(\omega_i)$
end while

Another approach to estimate the KL divergence term in (8) is to use a lower bound of f-divergence (Nguyen et al., 2010; Nowozin et al., 2016). There is a difference between the *lower bound approach* and the *probabilistic classification* presented in this subsection. In the former approach, the lower bound of the KL divergence is maximised to tighten the bound. In the latter approach, a discriminator is trained to minimise the logistic regression loss to estimate the ratio $q(\mathbf{w}_i; \lambda_i)/p(\mathbf{w}_i; \theta)$, and use Monte Carlo sampling to approximate the KL divergence of interest.

One potential drawback of the implicit modelling used in this paper is the curse of dimensionality, resulting in an expensive computation during training. This is an active research question when dealing with generative models in general. This issue can be addressed by encoding the high-dimensional data, such as images, to a feature embedding space by supervised-learning on the same training data set (Rusu et al., 2019). This strategy reduces the dimension of the input space, leading to smaller generator and discriminator models. The trade-off lies in the possibility of losing relevant information that can affect the performance on held-out tasks.

It is also worthy noting that our proposed method is easier to train than prior Bayesian few-shot meta-learning (Finn et al., 2018; Ravi & Beatson, 2019) because we no longer need to estimate the weighting factor of the KL divergence term in (8). The trade-off of our approach lies in the need to set the significance level δ, but tuning δ is arguably more intuitive than estimating the correct weighting factor for the KL divergence term.

4 Experimental Evaluation

We evaluate SImBa in both few-shot regression and classification problems. We also compare to prior state-of-art meta-learning methods to show the strengths and weaknesses of SImBa.

4.1 Regression

The experiment in this subsection is a multi-modal task distribution where half of the data is generated from sinusoidal functions, while the other half is from linear functions (Finn et al., 2018). The details of the experimental setup and additional visualisation results are presented in Appendix B. The results in Figure 2 (*leftmost and middle graphs*) show that SImBa is able to vary the prediction variance, especially when there is more uncertainty in the training data, while MAML can only output a single value at each data point.

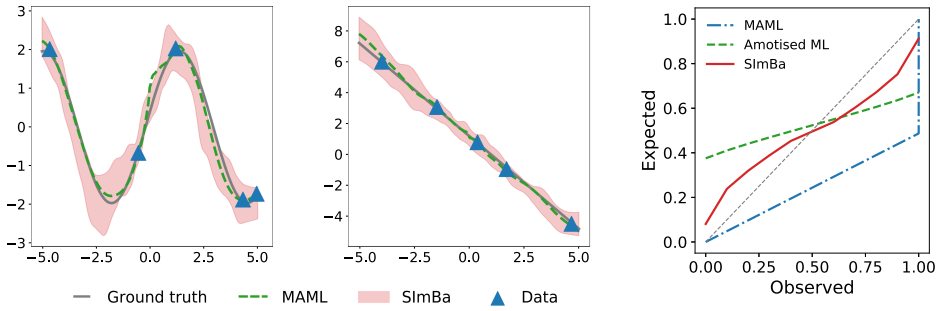


Figure 2: (*Leftmost and middle graphs*) SImBa and MAML are compared in a regression problem when training is based on multi-modal data – half of the tasks are generated from sinusoidal functions, and the other half are from linear functions. The shaded area is the prediction made by SImBa $\pm$ 2$\times$ standard deviation. (*Right*) Reliability chart of various meta-learning methods averaged over 1000 tasks.

To further evaluate the predictive uncertainty, we employ the reliability diagram based on the quantile calibration for regression (Song et al., 2019). The reliability diagram shows a correlation between predicted and actual probability. A perfect calibrated model will have its predicted probability equal to the actual probability, and hence, align well with the diagonal $y = x$. Figure 2 (*Right*) shows the results for SImBa and some published meta-learning methods. As expected, Bayesian meta-learning approaches, and in particular, Amortised Meta-learner, which relies on diagonal Gaussian distributions, are better calibrated than MAML – a deterministic approach. However, the averaged slope of the Amortised Meta-learner correlation curve is quite small, implying that its predicted probability is peaked at the mean of the ground-truth distribution with small covariances. In contrast, SImBa employs a much richer variational distribution, and therefore, resulting in a model with better calibration.

4.2 Few-shot Classification

We evaluate SImBa on the N-way k-shot setting, where a meta learner is trained on many related tasks containing N classes with k examples per class. We use the train-test split that consists of 64 classes for training, 16 for validation, and 20 for testing (Ravi & Larochelle, 2017). Please refer to Appendix C for the details of the model used.

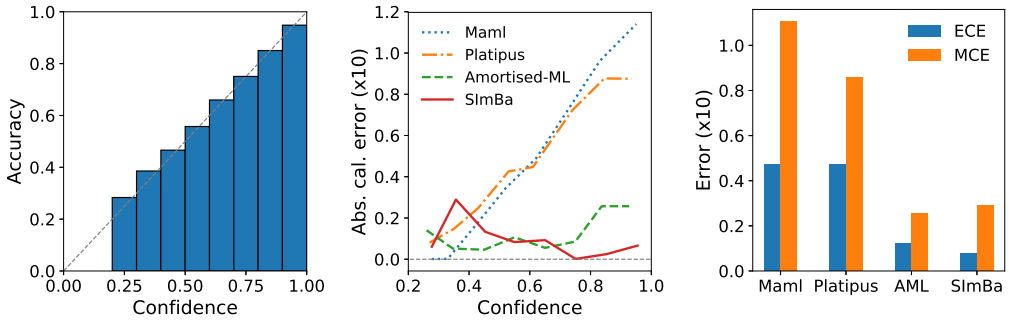
Although we target the estimation of model uncertainty, we also present the accuracy of SImBa against the state of the art on mini-ImageNet (Vinyals et al., 2016; Ravi & Larochelle, 2017). The results in Table 1 shows that SImBa achieves state-of-the-art in 1-shot setting when the base model is the 4-layer convolutional neural network (CNN) (Vinyals et al., 2016), and in 5-shot setting when different network architecture is used. We also show in Appendix D that generators with larger networks tend to classify better.

Similar to the experiment for regression, we use reliability diagrams (Guo et al., 2017) to evaluate the predictive uncertainty. The reliability diagrams show how well calibrated a model is when testing across many unseen tasks. A perfectly calibrated model will have its values overlapped with the identity function $y = x$, indicating that the probability associated with the label prediction is the same as the true probability. Figures 3a and 3b show the results of SImBa and other Bayesian meta-learning methods. Visually, the model trained with SImBa shows better calibration than the ones trained with MAML and PLATIPUS, while being competitive to Amortised Meta-learner. To further evaluate, we compute the expected calibration error (ECE) and maximum calibration error (MCE) (Guo et al., 2017) of the models trained with these methods. The results plotted in Figure 3c show that the model trained with SImBa has smaller ECE and MCE compared to MAML and PLATIPUS. SImBa also has lower ECE and competitive MCE compared to Amortised Meta-learner, but notice that Amortised Meta-learner has a worse classification result than SImBa, as shown in Table 1.

[1]Trained on 30-way 1-shot setting

Table 1: The few-shot 5-way classification accuracy results (in percentage, with 95% confidence interval) of SImBa averaged over 600 mini-ImageNet tasks are competitive to the state-of-the-art methods. SImBa outperforms other prior methods in 1-shot setting when using the standard 4-layer CNN, and 5-shot setting when using non-standard network architectures.

Method	**1-shot**	**5-shot**
Mini-ImageNet (Ravi & Larochelle, 2017) - standard 4-block CNN		
Matching nets (Vinyals et al., 2016)	43.56 ± 0.84	55.31 ± 0.73
Meta-learner LSTM (Ravi & Larochelle, 2017)	43.44 ± 0.77	60.60 ± 0.71
MAML (Finn et al., 2017)	48.70 ± 1.84	63.15 ± 0.91
Prototypical nets (Snell et al., 2017)[1]	49.42 ± 0.78	**68.20 ± 0.66**
LLAMA (Grant et al., 2018)	49.40 ± 1.83	-
PLATIPUS (Finn et al., 2018)	50.13 ± 1.86	-
Amortised ML (Ravi & Beatson, 2019)	45.00 ± 0.60	-
SImBa	**51.01 ± 0.31**	63.94 ± 0.43
Mini-ImageNet (Ravi & Larochelle, 2017) - non-standard network		
Relation nets (Sung et al., 2018)	50.44 ± 0.82	65.32 ± 0.70
VERSA (Gordon et al., 2019)	53.40 ± 1.82	67.37 ± 0.86
SNAIL (Mishra et al., 2018)	55.71 ± 0.99	68.88 ± 0.92
adaResNet (Munkhdalai et al., 2018)	56.88 ± 0.62	71.94 ± 0.57
TADAM (Oreshkin et al., 2018)	58.50 ± 0.30	76.70 ± 0.30
LEO (Rusu et al., 2019)	61.76 ± 0.08	77.59 ± 0.12
LGM-Net (Li et al., 2019)	**69.13 ± 0.35**	71.18 ± 0.68
SImBa	63.45 ± 0.54	**77.98 ± 0.47**



(a) Reliability diagram for SImBa (b) Normalised reliability diagram (c) ECE and MCE comparison

Figure 3: Uncertainty evaluation for 5-way 1-shot classification on mini-ImageNet.

5 Conclusion

We introduce and formulate a new Bayesian algorithm for few-shot meta-learning. The proposed algorithm, SImBa, is based on PAC-Bayes framework which theoretically guarantees prediction generalisation on unseen tasks. In addition, the proposed method employs a generative approach that implicitly models the shared prior $p(\mathbf{w}_i; \theta)$ and task-specific posterior $q(\mathbf{w}_i; \lambda_i)$, resulting in more expressive variational approximation compared to the usual diagonal Gaussian methods, such as PLATIPUS (Finn et al., 2018) or Amortised Meta-learner (Ravi & Beatson, 2019). The uncertainty, in the form of the learnt implicit distributions, can introduce more variability into the decision made by the model, resulting in well-calibrated and highly-accurate prediction. The algorithm can be combined with different base models that are trainable with gradient-based optimisation, and is applicable in regression and classification. We demonstrate that the algorithm can make reasonable predictions about unseen data in a multi-modal 5-shot learning regression problem, and achieve state-of-the-art calibration and classification results with on few-shot 5-way tasks on mini-ImageNet data set.

REFERENCES

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

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

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

John S Bridle and Stephen J Cox. Recnorm: Simultaneous normalisation and classification applied to speech recognition. In *Advances in Neural Information Processing Systems*, pp. 234–240, 1991.

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

George Cybenko. Approximations by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:183–192, 1989.

Peter J Diggle and Richard J Gratton. Monte carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 46(2):193–212, 1984.

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

Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *Conference on Neural Information Processing Systems*, pp. 9537–9548, 2018.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 2010.

Jonathan Gordon, John Bronskill, Matthias Bauer, Sebastian Nowozin, and Richard Turner. Meta-learning probabilistic inference for prediction. In *International Conference on Learning Representations*, 2019.

Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. In *International Conference on Learning Representations*, 2018.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4 (2):251–257, 1991.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Huaiyu Li, Weiming Dong, Xing Mei, Chongyang Ma, Feiyue Huang, and Bao-Gang Hu. Lgm-net: Learning to generate matching networks for few-shot learning. In *International Conference on Machine Learning*, pp. 3825–3834, 2019.

David A McAllester. Pac-bayesian model averaging. In *Annual Conference on Computational Learning Theory*, volume 99, pp. 164–170, 1999.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018.

Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *International Conference on Machine Learning*, 2017.

Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, and Adam Trischler. Rapid adaptation with conditionally shifted neurons. In *International Conference on Machine Learning*, pp. 3661–3670, 2018.

XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, pp. 271–279, 2016.

Boris N Oreshkin, Alexandre Lacoste, and Pau Rodriguez. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Conference on Neural Information Processing Systems*, pp. 719–729, 2018.

Anastasia Pentina and Christoph Lampert. A pac-bayesian bound for lifelong learning. In *International Conference on Machine Learning*, pp. 991–999, 2014.

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

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2017.

Michael T Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G Dietterich. To transfer or not to transfer. In *NIPS 2005 workshop on transfer learning*, volume 898, pp. 1–4, 2005.

Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*, 2019.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International Conference on Machine Learning*, pp. 1842–1850, 2016.

Jürgen Schmidhuber. *Evolutionary principles in self-referential learning (On learning how to learn: the meta-meta-... hook)*. Diploma thesis, Technische Universität München, 1987.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 4077–4087, 2017.

Hao Song, Tom Diethe, Meelis Kull, and Peter Flach. Distribution calibration for regression. In *International Conference on Machine Learning*, pp. 5897–5906, 09–15 Jun 2019.

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density ratio estimation in machine learning*. Cambridge University Press, 2012.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *Conference on Computer Vision and Pattern Recognition*, 2018.

Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 1998.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pp. 3630–3638, 2016.

Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *Advances in Neural Information Processing Systems 31*, pp. 7343–7353, 2018.

Ruixiang Zhang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. Metagan: An adversarial approach to few-shot learning. In *Advances in Neural Information Processing Systems 31*, pp. 2371–2380, 2018.

A PROOF OF PAC-BAYES FEW-SHOT META-LEARNING BOUND

First, we present the two auxiliary lemmas that helps to prove Theorem 2.

Lemma 1. *For $i = 1 : n$, if X_i and Y_i are random variables, then:*

$$p\left(\sum_{i=1}^{n} X_i \leq \sum_{i=1}^{n} Y_i\right) \geq p\left(\bigcap_{i=1}^{n} (X_i \leq Y_i)\right).$$

Proof. The proof is quite direct:

$$X_i \leq Y_i \implies \sum_{i=1}^{n} X_i \leq \sum_{i=1}^{n} Y_i. \tag{14}$$

Hence, the proof. □

Lemma 2. *For n events A_i with $i = 1 : n$, the following holds:*

$$p\left(\bigcap_{i=1}^{n} A_i\right) \geq \left(\sum_{i=1}^{n} p(A_i)\right) - (n-1),\ \forall n \geq 2.$$

Proof. Proof can be done by induction.

For $n = 2$:

$$p(A_1 \cap A_2) = p(A_1) + p(A_2) - p(A_1 \cup A_2) \geq p(A_1) + p(A_2) - 1.$$

Suppose that it is true for case n:

$$p\left(\bigcap_{i=1}^{n} A_i\right) \geq \left(\sum_{i=1}^{n} p(A_i)\right) - (n-1).$$

We prove that this is also true for case $(n+1)$:

$$\begin{aligned}
p\left(\bigcap_{i=1}^{n+1} A_i\right) &= p\left(\bigcap_{i=1}^{n} A_i\right) + p(A_{n+1}) - p\left(\left(\bigcap_{i=1}^{n} A_i\right) \bigcup A_{n+1}\right) \\
&\geq p\left(\bigcap_{i=1}^{n} A_i\right) + p(A_{n+1}) - 1 \\
&\geq \left(\sum_{i=1}^{n} p(A_i)\right) - (n-1) + p(A_{n+1}) - 1 \\
&\quad \text{(assumption of induction for case } n\text{)} \\
&\geq \left(\sum_{i=1}^{n+1} p(A_i)\right) - ((n+1)-1).
\end{aligned}$$

It is, therefore, true for $(n+1)$, and hence, the proof. □

Secondly, we apply the PAC-Bayes bound in Theorem 1 on the task i to obtain an upper-bound for a single task i shown in Corollary 1.

Corollary 1. *For a single task $\mathcal{T}_i$ in Eq. (3) and $\delta_i \in (0, 1)$, the following holds:*

$$p\left(\mathcal{L}_i^{(v)}(\theta) \leq \hat{\mathcal{L}}_i^{(v)}(\theta)\right) \geq 1 - \delta_i,$$

where: $\mathcal{L}_i^{(v)}(\theta)$ and $\hat{\mathcal{L}}_i^{(v)}(\theta)$ are defined in Eqs. (4) and (5).

Finally, we can employ Lemmas 1 and 2 combined with Corollary 1 to derive the novel upper-bound for few-shot meta-learning setting.

Theorem 2 (PAC-Bayes bound for few-shot meta-learning in (3))**.** *For the general error of few-shot meta-learning in (3), the following holds:*

$$p\left(\mathcal{L}^{(v)}(\theta) \leq \frac{1}{T}\sum_{i=1}^{T}\hat{\mathcal{L}}_i^{(v)}(\theta)\right) \geq 1-\delta.$$

Proof. Applying the inequality in Lemma 1 by replacing $X_i = \mathcal{L}_i^{(v)}(\theta)$ and $Y_i = \hat{\mathcal{L}}_i^{(v)}(\theta)$, and note that $\sum_{i=1}^{T}\mathcal{L}_i^{(v)}(\theta) = T\mathcal{L}^{(v)}$, gives:

$$p\left(T\mathcal{L}^{(v)}(\theta) \geq \sum_{i=1}^{T}\hat{\mathcal{L}}_i^{(v)}(\theta)\right) \geq p\left(\bigcap_{i=1}^{T}\left(\mathcal{L}_i^{(v)}(\theta) \leq \hat{\mathcal{L}}_i^{(v)}(\theta)\right)\right), \tag{15}$$

where $\mathcal{L}_i^{(v)}$, $\mathcal{L}^{(v)}$ and $\hat{\mathcal{L}}_i^{(v)}$ are defined at Eqs. (4), (3) and (5), respectively.

Applying Lemma 2 the right hand side term of Ineq. (15) gives:

$$p\left(\bigcap_{i=1}^{T}\left(\mathcal{L}_i^{(v)}(\theta) \leq \hat{\mathcal{L}}_i^{(v)}(\theta)\right)\right) \geq \sum_{i=1}^{T}p\left(\mathcal{L}_i^{(v)}(\theta) \leq \hat{\mathcal{L}}_i^{(v)}(\theta)\right) - (T-1). \tag{16}$$

Applying the transitive property for Ineqs. (15), (16) and Corollary 1, and setting $\delta_i = \delta/T$ prove the theorem. □

B SETUP OF REGRESSION EXPERIMENT

The experiment is carried out with half of the data being generated from sinusoidal functions, while the other half from linear functions. The amplitude and phase of the sinusoidal functions are uniformly sampled from [0.1, 5] and [0, π], respectively, while the slope and intercept of the lines are sampled from [-3, 3]. Data is uniformly generated from [-5, 5], and the corresponding label is added a zero-mean Gaussian noise with a standard deviation of 0.3. Each task consists of 5 data points used for training ($|\mathcal{Y}_i^{(t)}| = 5$), and 15 points used for validation ($|\mathcal{Y}_i^{(v)}| = 15$).

The base model used in the regression experiment is a three-hidden fully connected layer neural network. Each hidden layer has 100 hidden units ($1 \rightarrow 40 \rightarrow 40 \rightarrow 40 \rightarrow 1$), followed by *tanh* activation. No batch normalisation is used.

The generator is a fully connected network with two hidden layers consisting of 256 and 1024 units, respectively ($\dim(\mathbf{z}) \rightarrow 256 \rightarrow 1024 \rightarrow \dim(\mathbf{w}_i)$). The discriminator is also fully connected ($\dim(\mathbf{w}_i) \rightarrow 512 \rightarrow \dim(\mathbf{z}) \rightarrow 1$). These networks are activated by ReLU, except the last layer of the discriminator is activated by sigmoid function. No batch normalisation is used across these two networks.

The variational parameters λ_i and ω_i are estimated by performing five gradient updates with learning rate $\alpha_t = 0.001$ and $\gamma_t = 0.001$. The meta-parameters θ and the meta-parameter of the discriminator ω_0 are obtained with Adam (Kingma & Ba, 2015) with fixed step size $\alpha_v = 10^{-4}$ and $\gamma_v = 10^{-5}$. At the beginning of training, we clip the gradient when updating λ_i with a value of 10, and then gradually increase the clipping value. After 50,000 tasks, we remove the gradient clipping and continue to train until convergence.

The hyper-parameters related to Monte Carlo sampling are $L_t = L_v = 64$ and $L_D = 512$.

C SETUP OF CLASSIFICATION EXPERIMENT

The setup of the N-way k-shot is done with the validation consisting of 15 examples in each class ($|\mathcal{Y}_i^{(t)}| = kN$, and $|\mathcal{Y}_i^{(v)}| = 15N$). The latent noise $\mathbf{z}$ is a 100-dimensional vector sampled from a uniform distribution $\mathcal{U}(0, 1)$. Adam optimiser is employed to optimise both θ and ω_0. Please refer to Table 2 for other hyper-parameters used.

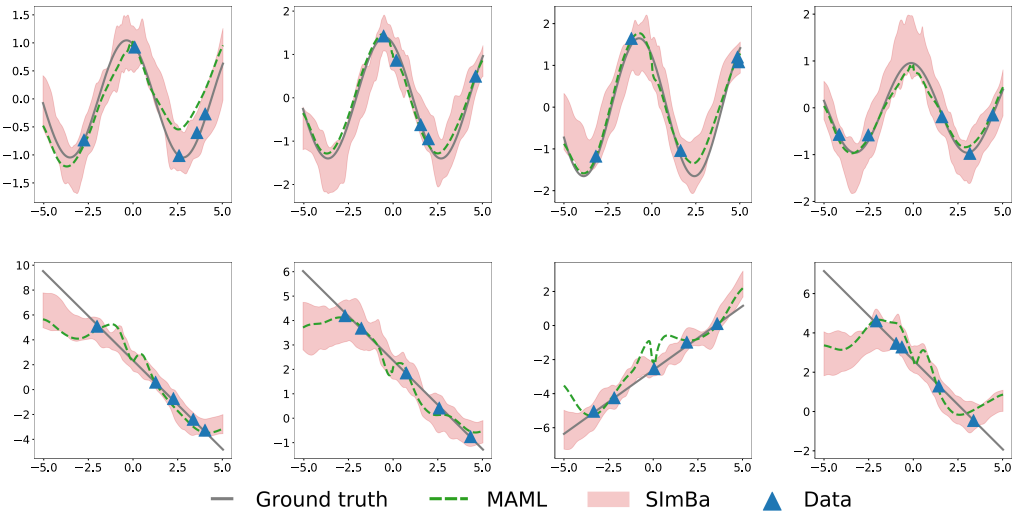


Figure 4: Additional regression results comparing SImBa and MAML.

C.1 STANDARD 4-BLOCK CNN

This model corresponds to the top part of Table 1. All input images are down-sampled to 84-by-84 pixels before performing experiments to be consistent with prior few-shot meta-learning works.

The base model is a 4-block CNN, where each block consists of 32 filters with a size of 3-by-3, followed by a batch normalisation and a ReLU activation function. The generator is a 2-hidden-layer fully connected network ($\dim(\mathbf{z}) \rightarrow 256 \rightarrow 1024 \rightarrow \dim(\mathbf{w}_i)$), where each layer is activated by ReLU without batch normalisation. The discriminator is also a fully connected network ($\dim(\mathbf{w}_i) \rightarrow 1024 \rightarrow 256 \rightarrow \dim(\mathbf{z}) \rightarrow 1$) with ReLU activation and without batch normalisation (the last activation function is a sigmoid).

C.2 NON-STANDARD NETWORKS

This corresponds to the bottom part of Table 1. Here, we employ the features extracted from (Rusu et al., 2019, Section 4.2.2) as the encoding of the input images. The training for the feature embedding consists of 3 steps. First, raw input images are down-sampled to 80-by-80 pixels. Second, a wide residual neural network WRN-28-10 is trained with data and labels from the 64 classes of the training set. Finally, the intermediate features of 640 dimensions at layer 21 are chosen as the embedding features used for our classification experiments.

The base model used in this experiment is a fully connected network with 1 hidden layer that consists of 128 hidden units ($640 \rightarrow 128 \rightarrow N$) followed by ReLU activation and batch normalisation. The

Table 2: Hyper-parameters used in the classification experiments.

Hyper-parameter	**Value**	
	Standard CNN	Non-standard network
T	2	10
L_t	8	32
L_v	8	32
α_t	10^{-2}	0.1
γ_t	3×10^{-6}	10^{-3}
α_v	10^{-5}	10^{-3}
γ_v	10^{-6}	5×10^{-5}

Table 3: Accuracy results in 5-way 1-shot classification carried out on mini-ImageNet using various base network and generator.

Number of hidden units		**Accuracy**
Generator	Base network	
128	16	52.84 ± 0.32
128	64	55.64 ± 0.47
256	16	54.19 ± 0.49
256	64	56.97 ± 0.56
512	64	59.67 ± 0.62
512	128	63.45 ± 0.54

generator model is constructed as a 1-hidden layer fully connected network with 512 hidden units, followed by ReLU without batch normalisation. The discriminator is also a fully connected network ($\dim(\mathbf{w}_i) \rightarrow 512 \rightarrow \dim(\mathbf{z}) \rightarrow 1$) with ReLU without batch normalisation.

D EFFECT OF NETWORK ARCHITECTURE

To study the effect of network architecture on the classification performance presented in Table 1, we repeat the classification experiment with the same setup, but different base networks. We vary the number of hidden units in the base network from 16 to 128, and also increase the size the the hidden layer of the generator from 256 to 512. The results in Table 3 show that the larger the base network and the generator are, the better the classification accuracy.