BAYESIAN ONLINE META-LEARNING WITH LAPLACE APPROXIMATION

Pau Ching Yap*
University College London

Hippolyt Ritter
University College London

David Barber
University College London
Alan Turing Institute

1 INTRODUCTION

Few-shot learning aims to adapt to novel classes with very few labelled examples from each class. Recent works show that meta-learning (Finn et al., 2017; Li et al., 2017; Santoro et al., 2016) provides promising approaches to few-shot classification problems. Despite being a promising solution to few-shot classification problems, meta-learning methods suffer from a limitation where a meta-learned model loses its few-shot classification ability on previous datasets as new ones arrive subsequently for training. A meta-learned model is restricted to do few-shot classification on a specific dataset, in the sense that the base and novel classes have to originate from the same dataset distribution (Snell et al., 2017; Vinyals et al., 2016; Sung et al., 2018). This paper considers meta-learning a single model for few-shot classification on multiple datasets that arrive sequentially for training.

In this paper, we introduce the Bayesian Online Meta-Learning with Laplace Approximation (BOMLA) framework to train a model that is applicable to a broader scope of few-shot classification datasets by overcoming catastrophic forgetting. We extend the online Laplace approximation framework (Ritter et al., 2018) to a Bayesian online meta-learning framework using the model-agnostic meta-learning (MAML) algorithm (Finn et al., 2017). Our framework combines Bayesian online learning (BOL) and MAML to find the posterior of the meta-parameters that can adapt quickly to novel classes using very few labelled examples.

2 MODEL-AGNOSTIC META-LEARNING

In this paper, we are interested in the well-known meta-learning algorithm MAML (Finn et al., 2017). Each updating step of MAML aims to improve the ability of the meta-parameters to act as a good model initialisation for a quick adaptation on unseen tasks. Each iteration of the MAML algorithm samples M tasks from the base class set $\widetilde{\mathcal{D}}$ and runs a few steps of stochastic gradient descent (SGD) for an inner loop task-specific learning. The number of tasks sampled per iteration is known as the *meta-batch size*. For task m, the inner loop outputs the task-specific $\tilde{\theta}^m$ from a k-step SGD quick adaptation on the objective $\mathcal{L}(\theta, \widetilde{\mathcal{D}}^{m,S})$ with the support set $\widetilde{\mathcal{D}}^{m,S}$ and initialised at θ:

$$\tilde{\theta}^m = SGD_k(\mathcal{L}(\theta, \widetilde{\mathcal{D}}^{m,S})), \tag{1}$$

where $m = 1, \ldots, M$. The outer loop gathers all task-specific adaptations to update the meta-parameters θ using the loss $\mathcal{L}(\tilde{\theta}^m, \widetilde{\mathcal{D}}^{m,Q})$ on the query set $\widetilde{\mathcal{D}}^{m,Q}$.

The overall MAML optimisation objective is

$$\arg\min_{\theta} \frac{1}{M} \sum_{m=1}^{M} \mathcal{L}(SGD_k(\mathcal{L}(\theta, \widetilde{\mathcal{D}}^{m,S})), \widetilde{\mathcal{D}}^{m,Q}). \tag{2}$$

Like most meta-learning algorithms, MAML assumes a stationary task distribution during meta-training and meta-evaluation. Under this assumption, a meta-learned model is only applicable to a specific dataset distribution. When the model encounters a sequence of datasets, it loses the few-shot classification ability on previous datasets as new ones arrive for meta-training. Our work aims to meta-learn a single model for few-shot classification on multiple datasets that arrive sequentially for meta-training. We achieve this goal by incorporating MAML into the BOL framework to give a *Bayesian online meta-learning* framework that finds the posterior of the meta-parameters.

*Corresponding author: p.yap@cs.ucl.ac.uk

3 OVERVIEW OF OUR BAYESIAN ONLINE META-LEARNING APPROACH

Our central contribution is to extend the benefits of meta-learning to the Bayesian online scenario, thereby training models that can generalise across tasks whilst dealing with parameter uncertainty in the setting of sequentially arriving datasets. Online meta-training occurs sequentially on the datasets $\mathcal{D}_1, \ldots, \mathcal{D}_T$. A newly-arrived $\mathcal{D}_{t+1}$ is separated into the base class set $\widetilde{\mathcal{D}}_{t+1}$ and novel class set $\widehat{\mathcal{D}}_{t+1}$ for meta-training and meta-evaluation respectively. Notationally, let $\widetilde{\mathcal{D}}^S_{t+1}$ and $\widetilde{\mathcal{D}}^Q_{t+1}$ denote the collection of support sets and query sets respectively from all possible tasks in the base class set $\widetilde{\mathcal{D}}_{t+1}$, so that $\widetilde{\mathcal{D}}_{t+1} = \widetilde{\mathcal{D}}^S_{t+1} \cup \widetilde{\mathcal{D}}^Q_{t+1}$. We are interested in a MAP estimate $\theta^* = \arg\max_\theta p(\theta|\widetilde{\mathcal{D}}_{1:t+1})$. Using Bayes' rule on the posterior gives the recursive formula

$$p(\theta|\widetilde{\mathcal{D}}_{1:t+1}) \propto p(\widetilde{\mathcal{D}}^S_{t+1}, \widetilde{\mathcal{D}}^Q_{t+1}|\theta)\, p(\theta|\widetilde{\mathcal{D}}_{1:t}) \tag{3}$$

$$= p(\widetilde{\mathcal{D}}^Q_{t+1}|\theta, \widetilde{\mathcal{D}}^S_{t+1})\, p(\widetilde{\mathcal{D}}^S_{t+1}|\theta)\, p(\theta|\widetilde{\mathcal{D}}_{1:t}) \tag{4}$$

$$\approx \left\{ \int p(\widetilde{\mathcal{D}}^Q_{t+1}|\tilde{\theta})\, p(\tilde{\theta}|\theta, \widetilde{\mathcal{D}}^S_{t+1})\, d\tilde{\theta} \right\} p(\theta|\widetilde{\mathcal{D}}_{1:t}) \tag{5}$$

where Eq. (3) follows from the assumption that each dataset is independent given θ, and Eq. (5) follows by dropping the likelihood term for a closer comparison to MAML.

From the meta-learning perspective, the parameters $\tilde{\theta}$ introduced in Eq. (5) can be viewed as the task-specific parameters in MAML. There are various choices for the distribution $p(\tilde{\theta}|\theta, \widetilde{\mathcal{D}}^S_{t+1})$ in Eq. (5). In particular if we choose to set it as the deterministic function of taking several steps of SGD on loss $\mathcal{L}$ with the support set collection $\widetilde{\mathcal{D}}^S_{t+1}$ and initialised at θ, we have

$$p(\tilde{\theta}|\theta, \widetilde{\mathcal{D}}^S_{t+1}) = \mathbb{1}\{\tilde{\theta} = SGD_k(\mathcal{L}(\theta, \widetilde{\mathcal{D}}^S_{t+1}))\}. \tag{6}$$

and this recovers the MAML inner loop with SGD quick adaptation in Eq. (1). The recursion given by Eq. (5) forms the basis of our approach and the remainder of this paper explains how we implement this. In order to do so we give a mini tutorial on Bayesian online learning and Laplace approximation in Appendix A.

4 BOMLA IMPLEMENTATION

This section demonstrates how we arrive at the Bayesian Online Meta-Learning with Laplace Approximation (BOMLA) framework by implementing the Laplace approximation to the posterior of the Bayesian online meta-learning framework in Eq. (5). BOMLA provides a grounded framework for an online training on the sequential few-shot classification datasets.

Laplace approximation rationalises the use of a Gaussian approximate posterior by Taylor expanding the log-posterior around a mode up to the second order. The second order term corresponds to the log-probability of a Gaussian distribution.

The Bayesian online meta-learning framework in Section 3 with a Gaussian approximation posterior q of mean and precision $\phi_t = \{\mu_t, \Lambda_t\}$ from the Laplace approximation gives a MAP estimate

$$\theta^* = \arg\max_\theta \left\{ \log \int p(\widetilde{\mathcal{D}}^Q_{t+1}|\tilde{\theta}) p(\tilde{\theta}|\theta, \widetilde{\mathcal{D}}^S_{t+1})\, d\tilde{\theta} - \frac{1}{2}(\theta - \mu_t)^T \Lambda_t (\theta - \mu_t) \right\}. \tag{7}$$

Using the deterministic $\tilde{\theta}$ in Eq. (6) and sampling M tasks per iteration as in MAML for the optimisation in Eq. (7) leads to minimising the objective

$$f_{t+1}(\theta, \mu_t, \Lambda_t) = \frac{1}{M} \sum_{m=1}^{M} -\log p(\widetilde{\mathcal{D}}^{m,Q}_{t+1}|\tilde{\theta}^m) + \frac{1}{2}(\theta - \mu_t)^T \Lambda_t (\theta - \mu_t), \tag{8}$$

where $\tilde{\theta}^m = SGD_k(\mathcal{L}(\theta, \widetilde{\mathcal{D}}^{m,S}_{t+1}))$ for $m = 1, \ldots, M$. The first term of the objective in Eq. (8) corresponds to the MAML objective in Eq. (2) with a cross-entropy loss, and the second term can be seen as a regulariser.

5 HESSIAN APPROXIMATION

Appendix B explains the block-diagonal Hessian approximation required to update the Gaussian posterior precision. The Hessian matrix corresponding to the first term of the BOMLA objective in Eq. (8) is

$$H_{t+1}^{ij} = \frac{1}{M}\sum_{m=1}^{M} -\frac{\partial^2}{\partial\theta^{(i)}\partial\theta^{(j)}}\log p(\widetilde{\mathcal{D}}_{t+1}^{m,Q}|\tilde{\theta}^m))\Big|_{\theta=\mu_{t+1}}. \tag{9}$$

It is worth noting that the BOMLA Hessian deviates from the original BOL Hessian in Eq. (17). This requires deriving an adjusted approximation to the Hessian with some further assumptions.

In the BOMLA framework, each (x, y) pair for the Fisher is associated to a task m. The Fisher information matrix $\widetilde{F}$ corresponding to the BOMLA Hessian in Eq. (9) is

$$\widetilde{F} = \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}_{x,y}\left[\left(\frac{\partial\tilde{\theta}^m}{\partial\theta}\right)\frac{d}{d\tilde{\theta}^m}\log p(y|x,\tilde{\theta}^m)\frac{d}{d\tilde{\theta}^m}\log p(y|x,\tilde{\theta}^m)^T\left(\frac{\partial\tilde{\theta}^m}{\partial\theta}\right)^T\right]. \tag{10}$$

The additional Jacobian matrix $\frac{\partial\tilde{\theta}^m}{\partial\theta}$ breaks the Kronecker-factored structure described by Martens & Grosse (2015) for the original Fisher in Eq. (18). By assuming the Jacobian to be the identity matrix, we retain the Kronecker-factored structure of Fisher approximation. The Fisher $\widetilde{F}$ in Eq. (10) with identity Jacobian matrix is an approximation to the Hessian $\widetilde{H}_{t+1}$ adjusted from H_{t+1} in Eq. (9) for a single data point:

$$\widetilde{H}_{t+1}^{ij} = \frac{1}{M}\sum_{m=1}^{M} -\frac{\partial^2\log p(\widetilde{\mathcal{D}}_{t+1}^{m,Q}|\tilde{\theta}^m))}{\partial\tilde{\theta}^{m,(i)}\partial\tilde{\theta}^{m,(j)}}\Big|_{\theta=\mu_{t+1}}. \tag{11}$$

6 EXPERIMENTS

Our experiments compare BOMLA to applying MAML continuously on sequential datasets in terms of their ability to overcome catastrophic forgetting in various few-shot classification settings. In both experiments we consider the 1-shot 5-way few-shot classification setting. The model architecture and other few-shot details can be found in Appendix C.

Rainbow Omniglot The artificial dataset sequence *Rainbow Omniglot* is generated analogous to Rainbow MNIST (Finn et al., 2019). We run the experiment on a sequence of 10 different datasets randomly chosen from Rainbow Omniglot. The details to generate Rainbow Omniglot with the 10 chosen datasets and the implementation details of this experiment can be found in Appendix D.1.

Figure 1 shows that MAML has a high fluctuation in the performance. When MAML encounters a more different dataset, the meta-parameters divert from previous experiences for an optimised performance on the current dataset. BOMLA on the other hand, has a relatively stable performance across dissimilar datasets, as it takes previous experiences into account when obtaining a posterior of the meta-parameters with the new dataset included.

A More Challenging Few-Shot Classification Dataset Sequence We implement BOMLA to a more challenging few-shot classification sequence:

$$\text{Omniglot} \rightarrow \textit{mini}\text{QuickDraw} \rightarrow \text{CIFAR-FS}$$

The details of this experiment and the datasets can be found in Appendix D.2. The result in Figure 2 shows that BOMLA is able to prevent few-shot catastrophic forgetting. BOMLA is able to proceed with learning *mini*QuickDraw with almost no forgetting on Omniglot. There is a small trade-off in the few-shot performance for CIFAR-FS as BOMLA avoids catastrophically forgetting Omniglot and *mini*QuickDraw.

Tuning the hyperparameter λ in Eq. (19) corresponds to balancing between a smaller performance trade-off on a new dataset and less forgetting on previous datasets. The λ value 1 results in a more concentrated Gaussian posterior and is therefore unable to learn new datasets well, but can better retain the performances on previously learned datasets. The λ value 0.01 on the other hand gives a

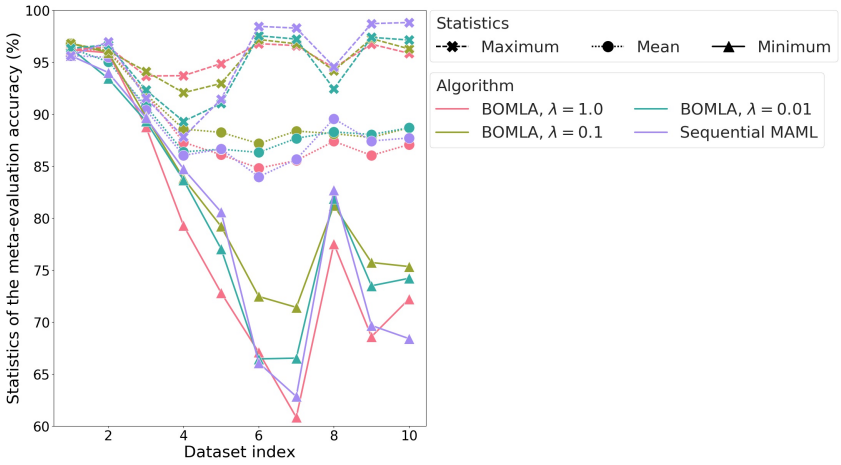

Figure 1: The statistics of the meta-evaluation accuracy over all datasets up to (and including) the current meta-training dataset index. Higher accuracy values with more stable lines within each statistic category indicate better results. BOMLA with $\lambda = 0.1$ outperforms MAML and other λ values in this context.

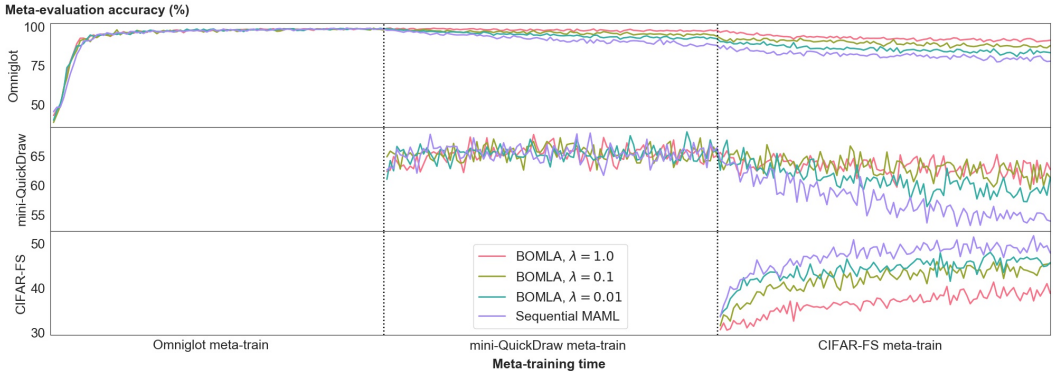

Figure 2: Meta-evaluation accuracy on Omniglot, *mini*QuickDraw and CIFAR-FS along meta-training. Higher accuracy values indicate better results with less forgetting as we proceed to new datasets. MAML catastrophically forgets the previously meta-learned Omniglot and *mini*QuickDraw when meta-training on CIFAR-FS, whereas BOMLA gives a good performance balance between the previous and current datasets.

widespread Gaussian posterior and learns better on new datasets by sacrificing the performance on the previous datasets. The result shows that the λ value 0.1 gives the best balance between old and new datasets.

7 Conclusion

We introduced the Bayesian Online Meta-learning with Laplace Approximation (BOMLA) framework to overcome catastrophic forgetting in few-shot classification problems by merging the BOL framework and the MAML algorithm. We proposed the necessary adjustment in the Hessian and Fisher approximation for BOMLA. The experiments show that BOMLA is able to retain the few-shot classification ability when trained on sequential datasets, resulting in the ability to perform few-shot classification on multiple datasets with a single meta-learned model.

REFERENCES

L. Bertinetto, J. F. Henriques, P. Torr, and A. Vedaldi. Meta-Learning with Differentiable Closed-Form Solvers. In *International Conference on Learning Representations*, 2019.

A. Botev, H. Ritter, and D. Barber. Practical Gauss-Newton Optimisation for Deep Learning. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

J. S. Denker and Y. LeCun. Transforming Neural-Net Output Levels to Probability Distributions. In *Advances in Neural Information Processing Systems 3*, 1991.

C. Finn, P. Abbeel, and S. Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

C. Finn, A. Rajeswaran, S. Kakade, and S. Levine. Online Meta-Learning. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

R. Grosse and J. Martens. A Kronecker-Factored Approximate Fisher Matrix for Convolution Layers. In *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

D. Ha and D. Eck. A Neural Representation of Sketch Drawings. *arXiv preprint*, arXiv:1704.03477, 2017.

D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015.

J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming Catastrophic Forgetting in Neural Networks. *Proceedings of the National Academy of Sciences*, 2017.

A. Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto, 2009.

B. Lake, R. Salakhutdinov, J. Gross, and J.B. Tenenbaum. One Shot Learning of Simple Visual Concepts. In *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, 2011.

Z. Li, F. Zhou, F. Chen, and H. Li. Meta-SGD: Learning to Learn Quickly for Few-Shot Learning. *arXiv preprint*, arXiv:1707.09835, 2017.

J. Martens and R. Grosse. Optimizing Neural Networks with Kronecker-Factored Approximate Curvature. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

M. Opper. *A Bayesian Approach to Online Learning*. Cambridge University Press, 1998.

H. Ritter, A. Botev, and D. Barber. Online Structured Laplace Approximations for Overcoming Catastrophic Forgetting. In *Advances in Neural Information Processing Systems 31*, 2018.

H. Robbins and S. Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 1951.

A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. Meta-Learning with Memory-Augmented Neural Networks. In *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

J. Snell, K. Swersky, and R. Zemel. Prototypical Networks for Few-Shot Learning. In *Advances in Neural Information Processing Systems 30*, 2017.

F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales. Learning to Compare: Relation Network for Few-Shot Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching Networks for One Shot Learning. In *Advances in Neural Information Processing Systems 29*, 2016.

A BACKGROUND

A.1 BAYESIAN ONLINE LEARNING

Upon the arrival of the $(t+1)$-th dataset $\mathcal{D}_{t+1}$ for large-scale supervised classification, we are interested in a MAP estimate $\theta^* = \arg\max_\theta p(\theta|\mathcal{D}_{1:t+1})$ for the parameters θ of a neural network. Using Bayes' rule on the posterior gives the recursive formula

$$p(\theta|\mathcal{D}_{1:t+1}) \propto p(\mathcal{D}_{t+1}|\theta)p(\theta|\mathcal{D}_{1:t}) \tag{12}$$

where Eq. (12) follows from the assumption that each dataset is independent given θ. As the normalised posterior $p(\theta|\mathcal{D}_{1:t})$ is usually intractable, it may be approximated by a parametric distribution q with parameter ϕ_t. The BOL framework consists of the *update step* and the *projection step* (Opper, 1998). The update step uses the approximate posterior $q(\theta|\phi_t)$ obtained from the previous step for an update in the form of Eq. (12):

$$p(\theta|\mathcal{D}_{1:t+1}, \phi_t) \propto p(\mathcal{D}_{t+1}|\theta)q(\theta|\phi_t). \tag{13}$$

The new posterior $p(\theta|\mathcal{D}_{1:t+1}, \phi_t)$ might not belong to the same parametric family as $q(\theta|\phi_t)$. In this case, the new posterior has to be projected into the same parametric family to obtain $q(\theta|\phi_{t+1})$. Opper (1998) performs this projection by minimising the KL-divergence between the new posterior and the parametric q, while Ritter et al. (2018) use the Laplace approximation instead.

A.2 LAPLACE APPROXIMATION

We discover that the Laplace approximation method provides a well-fitted meta-training framework for Bayesian online meta-learning in Eq. (5). Each updating step in the approximation procedure can be modified to correspond to the meta-parameters for few-shot classification, instead of the model parameters for large-scale supervised classification. Laplace approximation rationalises the use of a Gaussian approximate posterior by Taylor expanding the log-posterior around a mode up to the second order. The second order term corresponds to the log-probability of a Gaussian distribution.

For large-scale supervised classification, we consider finding a MAP estimate following from Eq. (12):

$$\theta^*_{t+1} = \arg\max_\theta p(\theta|\mathcal{D}_{1:t+1}) = \arg\max_\theta \{\log p(\mathcal{D}_{t+1}|\theta) + \log p(\theta|\mathcal{D}_{1:t})\}. \tag{14}$$

Since the posterior $p(\theta|\mathcal{D}_{1:t})$ of a neural network is intractable except for small architectures, the unnormalised posterior $\tilde{p}(\theta|\mathcal{D}_{1:t})$ is considered instead. Performing Taylor expansion on the logarithm of the unnormalised posterior around a mode θ^*_t gives

$$\log \tilde{p}(\theta|\mathcal{D}_{1:t}) \simeq \log \tilde{p}(\theta|\mathcal{D}_{1:t})\big|_{\theta=\theta^*_t} - \frac{1}{2}(\theta - \theta^*_t)^T A_t (\theta - \theta^*_t), \tag{15}$$

where A_t denotes the Hessian matrix of the negative log-posterior evaluated at θ^*_t. The expansion in Eq. (15) suggests using a Gaussian approximate posterior. Given the parameter $\phi_t = \{\mu_t, \Lambda_t\}$, a mean μ_{t+1} for step $t+1$ can be obtained by finding a mode of the approximate posterior as follows via standard gradient-based optimisation:

$$\mu_{t+1} = \arg\max_\theta \log p(\mathcal{D}_{t+1}|\theta) - \frac{1}{2}(\theta - \mu_t)^T \Lambda_t (\theta - \mu_t). \tag{16}$$

The precision matrix is updated as $\Lambda_{t+1} = H_{t+1} + \Lambda_t$, where H_{t+1} is the Hessian matrix of the negative log likelihood for $\mathcal{D}_{t+1}$ evaluated at μ_{t+1} with entries

$$H^{ij}_{t+1} = -\frac{\partial^2}{\partial\theta^{(i)}\partial\theta^{(j)}} \log p(\mathcal{D}_{t+1}|\theta)\bigg|_{\theta=\mu_{t+1}}. \tag{17}$$

For a neural network model, gradient-based optimisation methods such as SGD (Robbins & Monro, 1951) and Adam (Kingma & Ba, 2015) are the standard gradient-based methods in finding a mode for the Laplace approximation in Eq. (16). We show in Section 4 that this provides a well-suited skeleton to implement Bayesian online meta-learning in Eq. (5) with the mode-seeking optimisation procedure.

B BLOCK-DIAGONAL HESSIAN

Since the full Hessian matrix in Eq. (17) is intractable for large neural networks, we seek for an efficient and relatively close approximation to the Hessian matrix. Diagonal approximations (Denker & LeCun, 1991; Kirkpatrick et al., 2017) are memory and computationally efficient, but sacrifice approximation accuracy as they ignore the interaction between parameters. Consider instead separating the Hessian matrix into blocks where different blocks are associated to different layers of a neural network. A particular diagonal block corresponds to the Hessian for a particular layer of the neural network. The block-diagonal Kronecker-factored approximation (Martens & Grosse, 2015; Grosse & Martens, 2016; Botev et al., 2017) utilises the fact that each diagonal block of the Hessian is Kronecker-factored for a single data point. This provides a better Hessian approximation as it takes the parameter interactions within a layer into consideration.

The BOL Hessian in Eq. (17) for a single data point can be approximated using the Fisher information matrix F to ensure its positive semi-definiteness (Martens & Grosse, 2015):

$$F = \mathbb{E}_{x,y}\left[\frac{d}{d\theta}\log p(y|x,\theta)\frac{d}{d\theta}\log p(y|x,\theta)^T\right]. \tag{18}$$

Ritter et al. (2018) use a hyperparameter λ as a multiplier to the Hessian when updating the precision:

$$\Lambda_{t+1} = \lambda H_{t+1} + \Lambda_t. \tag{19}$$

In the large-scale supervised classification setting, this hyperparameter has a regularising effect on the Gaussian posterior approximation for a balance between having a good performance on a new dataset and maintaining the performance on previous datasets (Ritter et al., 2018). A large λ results in a sharply peaked Gaussian posterior and is therefore unable to learn new datasets well, but can prevent forgetting previously learned datasets. A small λ on the other hand gives a dispersed Gaussian posterior and allows better performance on new datasets by sacrificing the performance on the previous datasets.

C FEW-SHOT DETAILS

We average over 100 tasks sampled from the novel classes when reporting the meta-evaluation accuracy in all of the experiments. Each task in both of the 1-shot 5-way experiments consists of a support set with 1 sample per class and a query set with 15 samples per class. We use the model architecture proposed by Vinyals et al. (2016) that takes 4 modules with 64 filters of size 3×3, followed by a batch normalisation, a ReLU activation and a 2×2 max-pooling. A fully-connected layer is appended to the final module before getting the class probabilities with softmax.

D IMPLEMENTATION DETAILS

D.1 RAINBOW OMNIGLOT

Analogous to the Rainbow MNIST (Finn et al., 2019), our Rainbow Omniglot sequence is generated by transforming the character images from Omniglot in the following ways: scaling, rotating and changing the background colour of the images. To generate one of the sequential datasets, we apply a combined transformation formed by randomly selecting a scaling (to size 16×16 or original size), a rotation degree ($0°$, $90°$, $180°$or $270°$) and a background colour out of seven different colours. Rainbow MNIST scales the images to either half size or original, but we find the factor of half to be too small to generate reasonably interpretable images in Omniglot.

The sequence of the 10 Rainbow Omniglot datasets in this experiment is as follows:

1. Original scale, $0°$rotation and red background
2. Original scale, $270°$rotation and blue background
3. 16×16 scale, $0°$rotation and red background
4. 16×16 scale, $270°$rotation and green background
5. 16×16 scale, $180°$rotation and red background
6. Original scale, $180°$rotation and white background
7. Original scale, $0°$rotation and indigo background
8. 16×16 scale, $270°$rotation and cyan background
9. Original scale, $0°$rotation and yellow background
10. Original scale, $270°$rotation and green background

Each dataset is meta-trained for 3000 iterations using Adam with learning rate 0.01 and meta-batch size 32 for the outer loop optimisation. We perform a one-step SGD adaptation with learning rate 0.4 for the inner loop update on each task. We sample 1000 tasks to approximate the Hessian when updating the Gaussian precision matrix.

D.2 MORE CHALLENGING EXPERIMENT

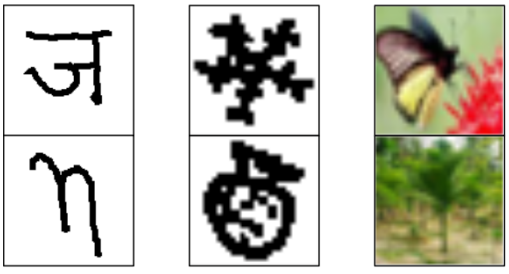

Figure 3: Examples of image instances from Omniglot (**left**), *mini*QuickDraw (**centre**) and CIFAR-FS (**right**).

Omniglot: The Omniglot dataset (Lake et al., 2011) comprises 1623 characters from 50 alphabets and each character has 20 instances. New classes with rotations in the multiples of 90° are formed after splitting the classes for meta-training and meta-evaluation.

***mini*QuickDraw:** The *mini*QuickDraw dataset is formed by randomly sampling 100 classes and 600 instances in each class from the QuickDraw dataset (Ha & Eck, 2017). The QuickDraw dataset comprises 345 categories of drawings collected from the players in the game "Quick, Draw!"

CIFAR-FS: The CIFAR-FS dataset (Bertinetto et al., 2019) is a variation on CIFAR100 (Krizhevsky, 2009) for the few-shot classification purpose, with 100 classes of objects and each class comprises 600 images of size 32×32. We rescale the images to 28×28 in this experiment.

For each of the Omniglot, *mini*QuickDraw and CIFAR-FS datasets, we update the meta-parameters in the outer loops using Adam with learning rate 0.001 and meta-batch size 32. Similar to the Rainbow Omniglot, the inner loop for Omniglot in this sequence does a one-step SGD with learning rate 0.4. The *mini*QuickDraw dataset uses a three-step SGD with learning rate 0.2 as an inner task-specific update. For CIFAR-FS, we perform a five-step SGD with learning rate 0.1 for an inner loop update. For each dataset, we sample 2000 tasks to approximate the Hessian when updating the Gaussian precision matrix.

Below are the 100 classes in the *mini*QuickDraw dataset:

```
'mailbox', 'whale', 'peanut', 'vase', 'octagon', 'dumbbell',
'hockey puck', 'chandelier', 'ocean', 'tennis racquet', 'bush',
'potato', 'tent', 'lobster', 'pool', 'squirrel', 'megaphone',
'bucket', 'golf club', 'jacket', 'computer', 'keyboard', 'basket',
'underwear', 'asparagus', 'cactus', 'arm', 'oven', 'elephant',
'moon', 'giraffe', 'couch', 'clock', 'suitcase', 'snowflake',
'scorpion', 'skyscraper', 'paint can', 'dragon', 'windmill',
'skateboard', 'fish', 'wristwatch', 'calculator', 'cat', 'hammer',
'sheep', 'necklace', 'bear', 'anvil', 'bulldozer', 'scissors',
'skull', 'syringe', 'zebra', 'helmet', 'bench', 'harp', 'river',
'monkey', 'bread', 'donut', 'train', 'flamingo', 'drill', 'peas',
'shorts', 'book', 'mushroom', 'brain', 'fireplace', 't-shirt',
'horse', 'cell phone', 'hexagon', 'zigzag', 'strawberry',
'sock', 'rainbow', 'crocodile', 'tree', 'bird', 'spreadsheet',
'teddy-bear', 'The Mona Lisa', 'bracelet', 'flying saucer',
'tractor', 'bathtub', 'cruise ship', 'car', 'parachute', 'grass',
'guitar', 'The Eiffel Tower', 'ear', 'drums', 'circle', 'compass',
'bandage'
```