Consolidating the Meta-Learning Zoo: A Unifying Perspective as Posterior Predictive Inference

Jonathan Gordon[†*] **John Bronskill**[†*] **Matthias Bauer**[†‡*]
Sebastian Nowozin[¶] **Richard E. Turner**[†¶]
[†]University of Cambridge [‡]Max Planck Institute for Intelligent Systems [¶]Microsoft Research
{jg801|jfb54|msb55|ret26}@cam.ac.uk Sebastian.Nowozin@microsoft.com

1 Introduction

A plethora of methods and approaches combining meta-learning [19, 21] with deep neural networks have recently been proposed, achieving great success in applications such as few-shot learning. Much of the existing work may be characterized as either gradient-based [4, 17, 9], metric-based [22, 20], or amortized MAP based meta-learning [8, 16]. Due to the ubiquity of recent work, a unifying view is useful for understanding and improving these methods. Existing frameworks [5, 9] are limited to specific families of approaches, namely gradient-based methods [17, 4]. In this paper we develop a framework for meta-learning approximate probabilistic inference for prediction, or ML-PIP for short. ML-PIP provides this unifying perspective in terms of amortizing posterior predictive distributions. We show that ML-PIP re-frames and extends existing probabilistic interpretations of meta-learning [5, 9] to cover both point-estimates and variational posteriors, as well as a broader class of methods, including gradient based meta-learning [4, 17], metric based meta-learning [20], amortized MAP inference [16], and conditional probability modelling [6, 7].

2 Meta-Learning Probabilistic Inference For Prediction

Our framework consists of (i) a standard and general multi-task probabilistic model, and (ii) a method for meta-learning probabilistic inference.

2.1 Probabilistic Model

Two principles guide the choice of model: First, the use of discriminative models to maximize predictive performance on supervised learning tasks [15]; second, the need to leverage shared statistical structure between tasks (i.e. multi-task learning). Both criteria are met by the standard multi-task directed graphical model [1, 9, 10] shown in Fig. 1 that employs shared (global) parameters θ, which are common to all tasks, and task specific parameters $\{\psi^{(t)}\}_{t=1}^{T}$. Inputs are denoted x and outputs y. Training data $D^{(t)} = \{(x_n^{(t)}, y_n^{(t)})\}_{n=1}^{N_t}$, and test data $\{(\tilde{x}_m^{(t)}, \tilde{y}_m^{(t)})\}_{m=1}^{M_t}$ are explicitly distinguished for each task t, as this is key for meta-learning.

In this work we focus on point-estimates for θ and develop an approximate inference scheme for the posterior distribution for ψ. We justify this choice as follows: θ is determined by all observations, and as such its posterior uncertainty is likely small, whereas the task-specific variables are determined by relatively few observations, and treating uncertainty properly is more important in this case.

2.2 Probabilistic Inference

This section provides a framework for meta-learning approximate inference that is a simple reframing and extension of existing approaches [5, 9]. Once the shared parameters θ are learned, the probabilistic

[*] Authors contributed equally.

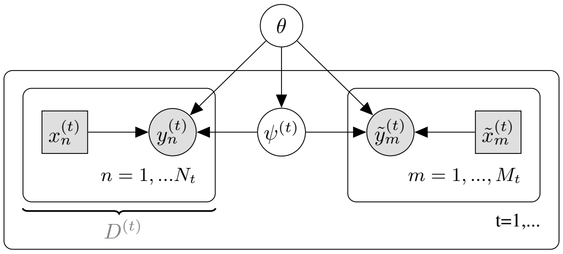

Figure 1: Directed graphical model for multi-task learning.

solution to few-shot learning in the model above comprises two steps: First, form the posterior distribution over the task-specific parameters $p(\psi^{(t)}|\tilde{x}^{(t)}, D^{(t)}, \theta)$; second, use that to compute the posterior predictive $p(\tilde{y}^{(t)}|\tilde{x}^{(t)}, \theta)$. These posteriors are in general intractable and in the following we explain how to approximate them. To reduce notational clutter, we suppress dependencies on θ and inputs $\tilde{x}$ for now, but reintroduce them in the final expression.

Specification of the approximate posterior predictive distribution. We approximate the posterior predictive distribution by a variational distribution $q_\phi(\tilde{y}|D)$, which we choose to *amortize*, that is, we learn a feed-forward inference network with parameters ϕ that takes any training dataset $D^{(t)}$ and test input $\tilde{x}$ as inputs and returns the predictive distribution over the test output $\tilde{y}^{(t)}$. We construct this by amortizing the approximate posterior $q_\phi(\psi|D)$ and then form the approximate posterior predictive distribution using:

$$q_\phi(\tilde{y}|D) = \int p(\tilde{y}|\psi) q_\phi(\psi|D) \mathrm{d}\psi. \tag{1}$$

Computing this integral may require additional approximation e.g. Monte Carlo sampling or point estimates. This is similar to a Variational Autoencoder [13, 18], which maps individual datapoints to a distribution in the latent space with an encoder, and employs a decoder to map points in latent space back to a distribution of interest (e.g., over the observed data). In contrast, our encoder maps a *set* of training observations to a single distribution in latent space, and we decode points in latent-space into a distribution over task-specific parameters. Crucially, the amortization enables fast predictions at test time, as we only need to perform a forward-pass to obtain the task specific parameters for a new task.

Meta-learning the approximate posterior predictive distribution. We measure the quality of the approximate posterior predictive for a single task by the KL-divergence between the true and approximate posterior predictive distribution

$$\phi^* = \underset{\phi}{\operatorname{argmin}} \underset{p(D)}{\mathbb{E}} \left[\mathrm{KL}\left[p(\tilde{y}|D) \| q_\phi(\tilde{y}|D)\right]\right] = \underset{\phi}{\operatorname{argmax}} \underset{p(\tilde{y},D)}{\mathbb{E}} \left[\log \int p(\tilde{y}|\psi) q_\phi(\psi|D) \mathrm{d}\psi\right]. \tag{2}$$

Training will therefore return parameters ϕ that best approximate the posterior predictive distribution in an average KL sense. So, if the approximate posterior $q_\phi(\psi|D)$ is rich enough, *global* optimization will recover the true posterior $p(\psi|D)$ (assuming $p(\psi|D)$ obeys identifiability conditions [2]).[2] Thus, the amortized procedure meta-learns approximate inference that supports accurate prediction.

The right hand side of Eq. (2) indicates how training could proceed: (i) select a task t at random, (ii) sample training data for that task $D^{(t)}$, (iii) form the posterior predictive over labels $q_\phi(\cdot|D^{(t)})$ and, (iv) compute the log-density $\log q_\phi(\tilde{y}^{(t)}|D^{(t)})$ at test data $\tilde{y}^{(t)}$ *not included in* $D^{(t)}$. Repeating this process many times and averaging the results will provide an unbiased estimate of the objective which can then be optimized. Thus, our probabilistic perspective that starts from minimizing $\mathrm{KL}\left[p(\tilde{y}|D) \| q_\phi(\tilde{y}|D)\right]$ naturally recovers the train / test splits at meta-train time that corresponds to "episodic training" typically used in the meta-learning literature [17, 20, 22]. This split also makes it clear that the procedure is scoring the approximate inference procedure by simulating approximate Bayesian held-out log-likelihood evaluation.

[2] Note that the true *predictive* posterior $p(y|D)$ is recovered regardless of the identifiability of $p(\psi|D)$.

End-to-end stochastic training. Armed by the insights above we now layout the full training procedure. We reintroduce inputs and shared parameters θ and the objective becomes

$$\mathcal{L}\left(\phi\right) = -\mathop{\mathbb{E}}_{p(D,\tilde{y},\tilde{x})}\left[\log q_{\phi}(\tilde{y}|\tilde{x},\theta)\right] = -\mathop{\mathbb{E}}_{p(D,\tilde{y},\tilde{x})}\left[\log \int p(\tilde{y}|\tilde{x},\psi,\theta)q_{\phi}(\psi|D,\theta)\mathrm{d}\psi\right]. \quad (3)$$

We optimize the objective over the shared parameters θ as this will maximize predictive performance (i.e., Bayesian held out likelihood). An end-to-end stochastic training objective for θ and ϕ is:

$$\hat{\mathcal{L}}\left(\theta,\phi\right) = \frac{1}{MT}\sum_{M,T}\log\frac{1}{L}\sum_{l=1}^{L}p\left(\tilde{y}_m^{(t)}|\tilde{x}_m^{(t)},\psi_l^{(t)},\theta\right), \quad \text{with } \psi_l^{(t)} \sim q_{\phi}(\psi|D^{(t)},\theta) \quad (4)$$

and $\{\tilde{y}_m^{(t)},\tilde{x}_m^{(t)},D^{(t)}\} \sim p(\tilde{y},\tilde{x},D)$, where p represents the data distribution (e.g., sampling tasks and splitting them into disjoint training data D and test data $\{(\tilde{x}_m^{(t)},\tilde{y}_m^{(t)})\}_{m=1}^{M_t}$). As stated above, this type of training uses episodic train / test splits at meta-training time. We have also approximated the integral over ψ using L Monte Carlo samples. The local reparametrization [14] trick enables optimization. Interestingly, the learning objective does not require an explicit specification of the prior distribution over parameters, $p(\psi^{(t)}|\theta)$, learning it implicitly through $q_{\phi}(\psi|D,\theta)$ instead.

Comparison to standard VI. We stress that our approach ML-PIP is different from standard Variational Inference (VI), which is the de-facto standard method for inference in probabilistic deep learning. ML-PIP directly minimizes the KL between the predictive distributions, whereas standard VI would proceed as follows: (i) Start with the marginal likelihood lower bound of *all* the meta-training data $p(\bar{y}|\bar{D},\theta)$, where $\bar{D} = D \cup \tilde{D}$ and $\bar{y} = y \cup \tilde{y}$; (ii) approximate the posterior over ψ, $p(\psi|\bar{D})$ with a variational distribution $q_{\phi}(\psi|\bar{D})$; (iii) use Jensen's inequality to obtain the following evidence lower bound (ELBO):

$$\hat{\mathcal{L}}(\theta,\phi) = \frac{1}{T}\sum_{t=1}^{T}\left(\sum_{(x,y)\in\bar{D}^{(t)}}\left(\frac{1}{L}\sum_{l=1}^{L}\log p(y|x,\psi_l^{(t)},\theta)\right) - \mathrm{KL}\left[q_{\phi}(\psi|\bar{D}^{(t)},\theta)\|p(\psi|\theta)\right]\right), \quad (5)$$

where $\psi_l^{(t)} \sim q_{\phi}(\psi|\bar{D}^{(t)})$. We note the following important differences between ML-PIP and VI: (i) Standard VI considers all the meta-training data in the same way, and so Eq. (5) does not lead to train / test splits at meta-training time. Importantly, such a split would violate Cox's axioms [3, 11] as it would entail performing inference *without using all available information*. (ii) The KL-term in the VI objective (Eq. (5)) regularizes the posterior $q_{\phi}(\psi|D)$ with a prior, whereas ML-PIP does not employ such a regularizer. (iii) Rather than minimizing $\mathrm{KL}(q_{\phi}(\psi|D)\|p(\psi|D))$, ML-PIP minimizes $\mathrm{KL}(p(\tilde{y}|D)\|q_{\phi}(\tilde{y}|D))$ directly. Thus, while standard VI focuses on $q_{\phi}(\psi|D)$, ML-PIP focuses on the approximate predictive posterior $q_{\phi}(\tilde{y}|D))$, and learns a posterior $q_{\phi}(\psi|D)$ that supports this predictive distribution.

In summary, we have developed an approach for Meta-Learning Probabilistic Inference for Prediction (ML-PIP). Next, we show that this formulation unifies a number of existing approaches.

3 Unifying Disparate Related Work with ML-PIP

In this section, we continue in the spirit of Grant et al. [9], recasting a broader class of meta-learning approaches as approximate inference in hierarchical models. We show that ML-PIP unifies a number of important approaches to meta-learning, including *both* gradient and metric based variants, as well as amortized posterior inference [16, 8], and conditional modelling approaches [7, 6]. We lay out these connections, most of which rely on point estimates for the task-specific parameters corresponding to $q(\psi^{(t)}|D^{(t)},\theta) = \delta\left(\psi^{(t)} - \psi^{*}(D^{(t)},\theta)\right)$.

Gradient-Based Meta-Learning. Let the task-specific parameters $\psi^{(t)}$ be all the parameters in a neural network. Consider a point estimate formed by taking a step of gradient ascent of the training objective, initialized at ψ_0 and with learning rate η.

$$\psi^{*}(D^{(t)},\theta) = \psi_0 + \eta\frac{\partial}{\partial\psi}\log\sum_{n=1}^{N_t}p(y_n^{(t)}|x_n^{(t)},\psi,\theta)\bigg|_{\psi_0}. \quad (6)$$

This is an example of semi-amortized inference [12], as the only shared inference parameters are the initialization and learning rate, and optimization is required for each task (albeit only for one step). Importantly, Eq. (6) recovers *model-agnostic meta-learning* [4], providing a perspective as semi-amortized ML-PIP. This perspective is complementary to that of Grant et al. [9] who justify the one-step gradient parameter update employed by MAML through MAP inference and the form of the prior $p(\psi|\theta)$. Note that the episodic train / test splits do not follow from the perspective provided by Grant et al. [9]. Instead we view the update choice as one of amortization which is trained using the predictive KL and naturally recovers the test-train splits. More generally, multiple gradient steps could be fed into an RNN to compute ψ^* which recovers Ravi and Larochelle [17].

Metric-Based Few-Shot Learning. Let the task-specific parameters be the top layer softmax weights and biases of a neural network $\psi^{(t)} = \{w_c^{(t)}, b_c^{(t)}\}_{c=1}^{C}$. The shared parameters are the lower layer weights. Consider amortized point estimates for these parameters constructed by averaging the top-layer activations for each class,

$$\psi^*(D^{(t)}, \theta) = \{w_c^*, b_c^*\}_{c=1}^{C} = \left\{\mu_c^{(t)}, -\|\mu_c^{(t)}\|^2/2\right\}_{c=1}^{C} \quad \text{where} \;\; \mu_c^{(t)} = \frac{1}{k_c}\sum_{n=1}^{k_c} h_\theta(x_n^{(c)}) \tag{7}$$

These choices lead to the following predictive distribution:

$$p(\tilde{y}^{(t)} = c|\tilde{x}^{(t)}, \theta) \propto \exp\left(-d(h_\theta(\tilde{x}^{(t)}), \mu_c^{(t)})\right) = \exp\left(h_\theta(\tilde{x}^{(t)})^T \mu_c^{(t)} - \frac{1}{2}\|\mu_c^{(t)}\|^2\right), \tag{8}$$

which recovers *prototypical networks* [20] using a Euclidean distance function d with the final hidden layer being the embedding space.

Amortized distributional meta-learning. As in the metric-based case, let the task specific parameters be the top layer linear classifier of a network and the shared parameters be the lower layers of the network. Now, let:

$$q_\phi(\psi|D, \theta) = \prod_{c=1}^{C} q_\phi\left(\psi_c|\{x_n^c\}_{n=1}^{k_c}, \theta\right), \tag{9}$$

where $\{x_n^c\}_{n=1}^{k_c}$ are observations in class c from task t, and ϕ are the parameters of a shared network that outputs a distribution over the weights and biases $w_c^{(t)}, b_c^{(t)}$ given these observations. This recovers VERSA [8], a method that processes sets [23] and employs distributional estimates for $\psi^{(t)}$.

Conditional models trained via maximum likelihood. In cases where a point estimate of the task-specific parameters are used the predictive becomes

$$q_\phi(\tilde{y}|D, \theta) = \int p(\tilde{y}|\psi, \theta) q_\phi(\psi|D, \theta) \mathrm{d}\psi = p(\tilde{y}|\psi^*(D, \theta), \theta). \tag{10}$$

In such cases the amortization network that computes $\psi^*(D, \theta)$ can be equivalently viewed as part of the model specification rather than the inference scheme. From this perspective, the ML-PIP training procedure for ϕ and θ is equivalent to training a conditional model $p(\tilde{y}|\psi_\phi^*(D, \theta), \theta)$ via maximum likelihood estimation, establishing the connection to neural processes [6, 7].

4 Conclusions

In this paper we have proposed ML-PIP, a general and powerful probabilistic framework for meta-learning. We have shown that many of the recent works and dominant approaches to meta-learning can be viewed as instances of this framework derived by a few key design choices. We believe this view is useful in understanding the different properties of existing approaches, and suggests additional design choices for future work. In particular, this view led to the development of VERSA (presented in an extended version of this workshop paper: [8]), which achieves state-of-the-art results in a number of few-shot learning tasks.

Acknowledgements

We thank Ravi and Larochelle for providing the miniImageNet dataset, and Yingzhen Li, Niki Kilbertus, Will Tebbutt, Maria Lomelli, and Robert Pinsler for their useful feedback. J.G. acknowledges funding from a Samsung Doctoral Scholarship. M.B. acknowledges funding by the EPSRC and a Qualcomm European Scholarship in Technology. R.E.T. acknowledges support from EPSRC grants EP/M0269571 and EP/L000776/1.

References

[1] B. Bakker and T. Heskes. Task clustering and gating for Bayesian multitask learning. *Journal of Machine Learning Research*, 4(May):83–99, 2003.

[2] G. Casella and R. L. Berger. *Statistical inference*, volume 2. Duxbury Pacific Grove, CA, 2002.

[3] R. T. Cox. Probability, frequency and reasonable expectation. *American journal of physics*, 14 (1):1–13, 1946.

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

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

[6] M. Garnelo, D. Rosenbaum, C. J. Maddison, T. Ramalho, D. Saxton, M. Shanahan, Y. W. Teh, D. J. Rezende, and S. Eslami. Conditional neural processes. *arXiv preprint arXiv:1807.01613*, 2018.

[7] M. Garnelo, J. Schwarz, D. Rosenbaum, F. Viola, D. J. Rezende, S. Eslami, and Y. W. Teh. Neural processes. *arXiv preprint arXiv:1807.01622*, 2018.

[8] J. Gordon, J. Bronskill, M. Bauer, S. Nowozin, and R. E. Turner. Decision-theoretic meta-learning: Versatile and efficient amortization of few-shot learning. *arXiv preprint arXiv:1805.09921*, 2018.

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

[10] T. Heskes. Empirical bayes for learning to learn. 2000.

[11] E. T. Jaynes. *Probability theory: the logic of science*. Cambridge university press, 2003.

[12] Y. Kim, S. Wiseman, A. C. Miller, D. Sontag, and A. M. Rush. Semi-amortized variational autoencoders. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[13] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

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

[15] A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems*, pages 841–848, 2002.

[16] S. Qiao, C. Liu, W. Shen, and A. Yuille. Few-shot image recognition by predicting parameters from activations. *arXiv preprint arXiv:1706.03466*, 2017.

[17] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[18] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286, 2014.

[19] J. Schmidhuber. *Evolutionary principles in self-referential learning*. PhD thesis, Technische Universität München, 1987.

[20] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4080–4090, 2017.

[21] S. Thrun and L. Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

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

[23] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola. Deep sets. In *Advances in Neural Information Processing Systems*, pages 3394–3404, 2017.