5th Symposium on Advances in Approximate Bayesian Inference, 2023 1–8

Amortised Inference in Neural Networks for Small-Scale Probabilistic Meta-Learning

Matthew Ashman* mca39@cam.ac.uk
Tommy Rochussen* tnr22@cam.ac.uk
Adrian Weller aw665@cam.ac.uk
University of Cambridge
** Equal contributions*

1. Introduction

In many machine learning applications, well-calibrated posterior predictive distributions are required for a number of closely-related datasets. Given similarity between datasets, it is natural to wish to develop meta-learning algorithms that utilise other datasets to reduce the computational complexity and / or improve predictive performance when deploying models on newly-seen datasets at test time. There have been a number of significant recent developments in meta-learning for predictive distributions, most notably that of the neural process (NP) family (Garnelo et al., 2018a,b; Foong et al., 2020; Gordon et al., 2018, 2019). Despite the utility of these methods on large-scale meta-datasets, they perform poorly in settings where the number of datasets and the total number of datapoints is small. We argue that this is a result of the large number of shared model parameters overfitting to the meta-dataset. A natural solution is to remove these shared model parameters, and instead train a meta-model to learn to approximate fully Bayesian inference over task-specific model parameters.

Recently, Ober and Aitchison (2021) developed a variational approximation for Bayesian neural networks (BNNs) based on using a set of inducing inputs to construct a series of conditional distributions that accurately approximate the conditionals of the true posterior distribution. Notably, the variational distribution consists of the prior multiplied by a set of approximate likelihoods for each inducing input. Our key insight is that these inducing inputs can be replaced by the actual data, such that the variational distribution consists of a set of approximate likelihoods for each datapoint. This structure lends itself to amortised inference, in which the parameters of each approximate likelihood are obtained by passing each datapoint through a meta-model known as the inference network. By training this inference network across related datasets, we can meta-learn Bayesian inference over task-specific BNNs, addressing the challenge above.

2. Related Work

Neural processes Our work is most similar to the NP family (Garnelo et al., 2018a,b), which seeks to meta-learn predictive distributions either through maximisation of the posterior predictive likelihood or variational inference (VI) (Foong et al., 2020). Similar to our method, NPs utilise an encoder to create embeddings for each datapoint. These embeddings

© M. Ashman, T. Rochussen & A. Weller.

are then aggregated to form a distribution over a latent variable which is then sampled and passed, together with a test datapoint, through a decoder. Volpp et al. (2020) propose the use of Bayesian aggregation, in which embeddings of individual datapoints take the form of approximate likelihoods which are multiplied together with the prior to form an approximate posterior distribution over the latent variable. Whilst these methods differ to ours in their use of shared model parameters, our method can be reinterpreted as a member of the NP family in which the latent variables are the parameters of the decoder. Through this perspective we can train our model in an identical way to NPs.

Meta-learning neural networks Meta-learning for neural networks had received a significant amount of attention from the research community. Notable examples include MAML Finn et al. (2017) and its extensions (Yoon et al., 2018; Antoniou et al., 2018), which seek good parameter initialisation, and those which explicitly condition on the dataset to obtain task specific parameters (Requeima et al., 2019; Gordon et al., 2018). Whilst conceptually similar to our approach, these methods differ in their use of shared model parameters—the task specific parameters amount to a small subset of the overall model parameters. In addition to requiring a large meta-dataset, this limits these methods to meta-datasets in which the individual datasets are very similar. By contrast, our approach does not use any shared model parameters but rather meta-learns inference in a BNN. We discuss the relationship between our method and NPs in Appendix A, demonstrating that they conceptually differ only in a change of objective functions.

3. Background

In this section we review the GI-BNN of Ober and Aitchison (2021) and NP family (Garnelo et al., 2018b). Throughout, let $\boldsymbol{\Xi} = \{\mathcal{D}\}$ denote a meta-dataset of $|\Xi|$ datasets, and $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$ denote a dataset consisting of inputs $\mathbf{X} \in \mathbb{R}^{N \times D_0}$ and outputs $\mathbf{y} \in \mathbb{R}^{N \times P}$.

3.1. Global Inducing Point Variational Posteriors for BNNs

Let $\mathbf{W} = \{\mathbf{W}^\ell\}_{l=1}^L$ denote the weights of an L-layer neural network, such that $\mathbf{W}^\ell \in \mathbb{R}^{D^{\ell-1} \times D^\ell}$ where D^ℓ denotes the dimensions of the ℓ-th hidden layer, and $\psi(\cdot)$ denote the element-wise activation function acting between layers. Ober and Aitchison (2021) introduce the global inducing point variational approximation for BNNs, in which the variational approximation to the posterior $p(\mathbf{W}|\mathcal{D})$ is defined recursively as $q_\phi(\mathbf{W}) = \prod_{\ell=1}^L q_\phi(\mathbf{W}^\ell | \{\mathbf{W}^{\ell'}\}_{\ell'=1}^{\ell-1}, \mathbf{U}^0)$, where

$$q_\phi\left(\mathbf{W}^\ell | \{\mathbf{W}^{\ell'}\}_{\ell'=1}^{\ell-1}, \mathbf{U}_0\right) \propto \prod_{d=1}^{D^\ell} p\left(\mathbf{w}_d^\ell\right) \underbrace{\mathcal{N}\left(\mathbf{v}_d^\ell; \psi(\mathbf{U}^\ell)\mathbf{w}_d^\ell, \left[\boldsymbol{\Lambda}_d^\ell\right]^{-1}\right)}_{t_d^\ell(\mathbf{w}_d^\ell)}. \tag{1}$$

This mirrors the structure of the true posterior in the sense that it is equivalent to the product of the prior and an *approximate likelihood*, $t_d^\ell(\mathbf{w}_d^\ell)$. The variational parameters ϕ of this approximation are the parameters of each approximate likelihood, $\mathbf{v}_d^\ell \in \mathbb{R}^M$ and $\boldsymbol{\Lambda}_d^\ell \in \mathbb{R}^{M \times M}$—which themselves can be interpreted as *pseudo observations*—and the *global*

inducing locations, $\mathbf{U}_0 \in \mathbb{R}^{M \times D_0}$, which are used to define $\{\mathbf{U}^\ell\}_{\ell=1}^L$ according to

$$\mathbf{U}^1 = \mathbf{U}^0 \mathbf{W}_1, \quad \mathbf{U}^\ell = \psi(\mathbf{U}^{\ell-1})\mathbf{W}^\ell \quad \ell = 2, \ldots, L. \tag{2}$$

Optimisation of ϕ is achieved through maximisation of the evidence lower bound (ELBO):

$$\mathcal{L}_{\text{ELBO}}(\phi; \mathcal{D}) = \mathbb{E}_{q_\phi(\mathbf{W})}\left[\log p(\mathbf{y}|\mathbf{W}, \mathbf{X})\right] - \text{KL}\left[q_\phi(\mathbf{W})||p(\mathbf{W})\right]. \tag{3}$$

We refer to this variational approximation as pseudo-observation variational inference for BNNs (POVI-BNN).

Ober and Aitchison (2021) demonstrate the efficacy of POVI-BNNs relative to mean-field Gaussian variational approximations for BNNs, achieving state-of-the-art performance on a number of regression and classification experiments. Their effectiveness is further demonstrated by Bui (2022), who shows that the estimate of the marginal likelihood provided by the POVI-BNN approximation is close to the true value, indicating the approximation is close to the true posterior.

4. Amortising Inference in Bayesian Neural Networks

In this section, we build upon the work of Ober and Aitchison (2021) to develop an effective method of performing amortised inference in BNNs.

Consider the same variational approximation described in Section 3.1, except with diagonal precision matrices $\mathbf{\Lambda}_d^\ell$ and inducing locations $\mathbf{U}$ replaced by training locations $\mathbf{X}$, such that

$$q\left(\mathbf{W}^\ell | \{\mathbf{W}^{\ell'}\}_{\ell'=1}^{\ell-1}, \mathcal{D}\right) \propto \prod_{d=1}^{D_\ell} p\left(\mathbf{w}_d^\ell\right) \prod_{n=1}^{N} \underbrace{\mathcal{N}\left(v_{d,n}^\ell; x_{d,n}^\ell, {\sigma_{d,n}^\ell}^2\right)}_{t_{d,n}^\ell(\mathbf{w}_d^\ell)} \tag{4}$$

where

$$\mathbf{x}_n^1 = \mathbf{W}^1 \mathbf{x}_n, \quad \mathbf{x}_n^\ell = \mathbf{W}^\ell \psi(\mathbf{x}_n^{\ell-1}) \quad \forall \ell = 2, \ldots, L. \tag{5}$$

This form of variational approximation enables the use of per-datapoint amortised inference. Specifically, rather than treating the variational parameters $\{\{\{v_{d,n}^\ell, \sigma_{d,n}^\ell\}_{n=1}^N\}_{d=1}^{D_\ell}\}_{\ell=1}^L$ as free-form parameters to optimise, we obtain them by passing each datapoint $(\mathbf{x}_n, \mathbf{y}_n)$ through an inference network at each layer:

$$\mathbf{v}_n^\ell, \ \log \sigma_n^\ell = g_\phi^\ell(\mathbf{x}_n, \mathbf{y}_n) \quad \forall \ell = 1, \ldots, L. \tag{6}$$

The variational parameters now become those of the inference networks, $\{g_\phi^\ell\}_{\ell=1}^L$. These variational parameters can be shared across datasets, and can be learned using a meta-dataset Ξ with the training objective:

$$\phi^* = \arg\max_\phi \mathcal{L}(\{\phi^\ell\}_{\ell=1}^L; \Xi); \quad \mathcal{L}(\{\phi^\ell\}_{\ell=1}^L; \Xi) = \frac{1}{|\Xi|} \sum_{i=1}^{|\Xi|} \mathcal{L}_{\text{ELBO}}(\{\phi^\ell\}_{\ell=1}^L; \mathcal{D}_i) \tag{7}$$

where $\mathcal{L}_{\text{ELBO}}$ is defined in Equation 3. We refer to our method as amortised pseudo-observation variational inference for BNNs (A-POVI-BNN).

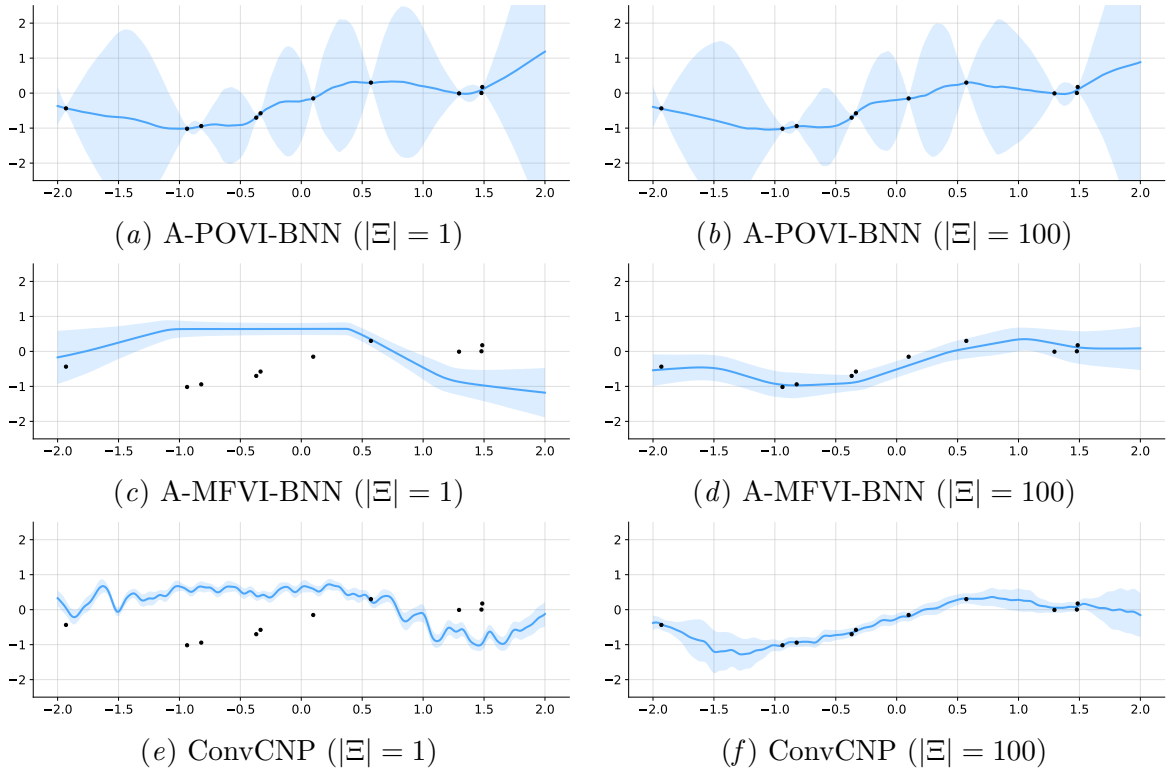

Figure 1: Posterior predictive distributions for A-POVI-BNN (top), A-MFVI-BNN (middle), and ConvCNP (bottom) after training on a meta-dataset of size $|\Xi| = 1$ (left) and $|\Xi| = 100$ (right). Data points are shown as black dots, the predictive mean is shown as a dark blue line and the 95% confidence interval is shown as shaded blue.

An important limitation of this approach is that performing stochastic optimisation through mini-batching datapoints is not possible, as to compute the $q(\mathbf{W}|\mathcal{D})$ we require passing the entire dataset $\mathcal{D}$ through the inference network. Nonetheless, this limitation is only significant for large datasets—provided the entire dataset can be passed through the network at once (i.e. in the small to medium-sized dataset regime which we consider here) this is not an issue. At test time, we can obtain an approximate posterior $q(\mathbf{W}|\mathcal{D}_*)$ with a single pass of the dataset through the inference networks.

5. Results and Discussion

In this section, we evaluate the performance of our model in the meta-learning setting. We consider a synthetic meta-dataset consisting of samples from a GP with a squared-exponential (SE) kernel. Each dataset consists of between 10 and 20 training datapoints. We compare A-POVI-BNNs to amortised mean-field variational inference for BNNs (A-MFVI-BNN), which we detail in Appendix B, and a ConvCNP (Gordon et al., 2019). All

NNs used in the amortised BNN architectures (both the model and inference network) consist of two layers of 50 hidden units and ReLU activation functions. The ConvCNP implementation and architecture is identical to that provided in https://github.com/cambridge-mlg/convcnp/blob/master/convcnp_regression.ipynb.

Figures 1(a) to 1(f) compare the posterior predictive distributions of each method on an unseen dataset drawn from a GP with the same hyperparameters as those in the meta-dataset. We evaluate the predictions of each method trained on a meta-dataset of size $|\Xi| = 1$ and $|\Xi| = 100$. We see that only A-POVI-BNN obtains a sensible predictive posterior on the unseen dataset in both cases. The ConvCNP performs poorly for $|\Xi| = 1$, which is unsurprising given the large number of model parameters increasing it susceptibility to overfitting. The A-MFVI-BNN performs significantly better for $|\Xi| = 100$, yet, in both cases the quality of its predictive posterior is poor relative to both the A-POVI-BNN and ConvCNP.

Despite these results being very preliminary, they are encouraging and suggest that A-POVI-BNNs may provide a more effective alternative to NPs when the size of meta-datasets are small. We intend to explore the effectiveness of A-POVI-BNNs in more diverse settings, such as image completion, in future work.

References

Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your MAML. *arXiv preprint arXiv:1810.09502*, 2018.

Thang D Bui. Biases in variational Bayesian neural networks. 2022.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

Andrew Foong, Wessel Bruinsma, Jonathan Gordon, Yann Dubois, James Requeima, and Richard Turner. Meta-learning stationary stochastic process prediction with convolutional neural processes. *Advances in Neural Information Processing Systems*, 33:8284–8295, 2020.

Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and SM Ali Eslami. Conditional neural processes. In *International Conference on Machine Learning*, pages 1704–1713. PMLR, 2018a.

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, SM Eslami, and Yee Whye Teh. Neural processes. *arXiv preprint arXiv:1807.01622*, 2018b.

Jonathan Gordon, John Bronskill, Matthias Bauer, Sebastian Nowozin, and Richard E Turner. Meta-learning probabilistic inference for prediction. *arXiv preprint arXiv:1805.09921*, 2018.

Jonathan Gordon, Wessel P Bruinsma, Andrew YK Foong, James Requeima, Yann Dubois, and Richard E Turner. Convolutional conditional neural processes. *arXiv preprint arXiv:1910.13556*, 2019.

Sebastian W Ober and Laurence Aitchison. Global inducing point variational posteriors for Bayesian neural networks and deep Gaussian processes. In *International Conference on Machine Learning*, pages 8248–8259. PMLR, 2021.

James Requeima, Jonathan Gordon, John Bronskill, Sebastian Nowozin, and Richard E Turner. Fast and flexible multi-task classification using conditional neural adaptive processes. *Advances in Neural Information Processing Systems*, 32, 2019.

Michael Volpp, Fabian Flürenbrock, Lukas Grossberger, Christian Daniel, and Gerhard Neumann. Bayesian context aggregation for neural processes. In *International Conference on Learning Representations*, 2020.

Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. *Advances in neural information processing systems*, 31, 2018.

Appendix A. Relationship to Neural Processes

The NP family (Garnelo et al., 2018a,b) describes the class of meta-learning models which use neural networks to map from datasets to posterior predictive distributions in a single forward pass. Broadly speaking, NPs fall into two categories—conditional NPs (CNPs) and latent NPs (LNPs). In both cases, each observation in the context dataset $\mathcal{D}_c$ is processed separately using an *encoder*, $\mathbf{r}_c = e_{\theta_e}(\mathbf{x}_c, \mathbf{y}_c)\ \forall(\mathbf{x}_c, \mathbf{y}_c) \in \mathcal{D}_c$, where θ_e parameterises a NN. The local encodings $\{\mathbf{r}_c\}_{c=1}^{C}$ are combined into a single representation $\mathbf{r}(\mathcal{D}_c)$ using a permutation invariant aggregation function, such as summation. This encoding is then passed to the model decoder d_{θ_d}, where θ_d also parameterises a NN, such that $d_{\theta_d}(\cdot; \mathbf{r}(\mathcal{D}_c))$ maps from target inputs $\mathbf{X}_T$ to predictive distributions. CNPs assume the posterior predictive distribution $p(\mathbf{y}_T|\mathcal{D}_C, \mathbf{X}_T)$ factorises as

$$p(\mathbf{y}_T|\mathcal{D}_C, \mathbf{X}_T) = \prod_{t=1}^{T} p_\theta(\mathbf{y}_t|d_{\theta_d}(\mathbf{x}_t; \mathbf{r}(\mathcal{D}_c))). \tag{8}$$

LNPs differ to CNPs through their use of a stochastic latent variable $\mathbf{z} \sim p_\theta(\mathbf{z}|\mathbf{r}(\mathcal{D}_c))$, rather than a fixed embedding:

$$p_\theta(\mathbf{y}_T|\mathbf{X}_T, \mathcal{D}_C) = \int p_\theta(\mathbf{z}|\mathbf{r}_C) \prod_{t\in\mathcal{T}} p_\theta(\mathbf{y}_t|d_{\theta_d}(\mathbf{x}_t; \mathbf{z}))d\mathbf{z}. \tag{9}$$

Importantly, doing so enables dependencies to be maintained across target locations inducing by marginalisation over $\mathbf{z}$.

A.1. Training Neural Processes

Members of the NP family are trained through minimising the empirical loss of the meta-dataset $\boldsymbol{\Xi}$:

$$\theta^* = \arg\min_\theta \mathcal{L}(\theta; \Xi); \quad \mathcal{L}(\theta; \Xi) := \frac{1}{M} \sum_{m=1}^{M} \mathcal{L}(\theta; \mathcal{D}_m). \tag{10}$$

There are a variety of ways in which the loss function $\mathcal{L}(\theta; \mathcal{D}_m)$ can be constructed. Focusing on LNPs, the two predominant methods are variational inference (VI) and maximum likelihood (ML). VI training employs the loss function

$$\mathcal{L}(\theta; \mathcal{D}) = -\mathbb{E}_{p_\theta(\mathbf{z}|\mathcal{D}_c \cup \mathcal{D}_T)} \left[\log p_\theta(\mathbf{y}_T|\mathbf{X}_T, \mathbf{z})\right] + \mathrm{KL}\left[p_\theta(\mathbf{z}|\mathcal{D}_c \cup \mathcal{D}_T)||p(\mathbf{z}|\mathcal{D}_c)\right] \tag{11}$$

where $\mathcal{D}_c$ and $\mathcal{D}_T$ are two random non-overlapping partitions of the dataset $\mathcal{D}$. This objective is motivated by the interpretation of $p_\theta(\mathbf{z}|\mathcal{D})$ as an approximation to the posterior distribution over $\mathbf{z}$ given $\mathcal{D}$. If $p_\theta(\mathbf{z}|\mathcal{D})$ is a good approximation to the posterior, then this objective becomes equivalent to the standard VI objective (ELBO):

$$\mathcal{L}(\theta; \mathcal{D}) = -\mathbb{E}_{p_\theta(\mathbf{z}|\mathcal{D})} \left[\log p_\theta(\mathbf{y}_T|\mathbf{X}_T, \mathbf{z})\right] + \mathrm{KL}\left[p_\theta(\mathbf{z}|\mathcal{D})||p(\mathbf{z})\right]. \tag{12}$$

Nonetheless, this objective potentially wastes modelling power on obtaining consistent posterior approximations, rather than good posterior predictive distributions. Instead, Foong et al. (2020) argue that the following ML objective should be used instead:

$$\mathcal{L}(\theta; \mathcal{D}) = -\log p_\theta(\mathbf{y}_T|\mathbf{X}_T, \mathcal{D}_c). \tag{13}$$

An important difference between the two objectives is that in the VI objective, the parameters of the embedding function can be interpreted as variational parameters, whereas in this ML objective all parameters are model parameters.

A.2. Amortised Inference in BNNs as Neural Processes

There are distinct similarities between the model introduced in the previous section and members of the latent neural process (LNP) family. In both cases we

1. compute per-datapoint embeddings of the dataset;
2. perform a permutation invariant aggregation of embeddings to get a single embedding for the entire dataset;
3. use this embedding to define a distribution over a latent variable;
4. sample this latent variable to define mappings from input location to predictive distributions.

The only principal difference between LNPs and our model is the use of shared model parameters in the decoder of LNPs. In our method, the *latent variable is the parameters of the decoder*. Alternative models such as CNAPs (Requeima et al., 2019) and ML-PIP (Gordon et al., 2018) are similar in this manner, except that the per-dataset latent variables are only a small subset of the decoder parameters.

Given these similarities, it is natural to consider training our model in an identical manner to LNPs—in particular, using the ML objective. In doing so, the parameters of the inference network no longer act as variational parameters but rather become part of the model, and so increase the propensity of the model to overfit. We leave exploration of this to future work.

Appendix B. Amortised Mean-Field Variational Inference for BNNs

A popular variational approximation for BNNs is the mean-field Gaussian approximation, given by

$$q(\mathbf{W}) = \prod_{\ell=1}^{L} \prod_{i=1}^{|\mathbf{W}^\ell|} \mathcal{N}\left(w_i; \mu_i, \sigma_i^2\right) \tag{14}$$

where $|\mathbf{W}^\ell|$ denotes the number of elements of weight matrix $\mathbf{W}^\ell$. We cannot directly apply per-datapoint amortisation to this form of the mean-field approximation as there are not per-datapoint variational parameters. However, we can obtain per-datapoint variational parameters by instead constructing the variational approximation as

$$q(\mathbf{W}) \propto p(\mathbf{W}) \prod_{n=1}^{N} t_n(\mathbf{W}) \tag{15}$$

where each approximate likelihood $t_n(\mathbf{W})$ is an unnormalised mean-field Gaussian distribution over $\mathbf{W}$. Provided $p(\mathbf{W})$ is also mean-field Gaussian (which is typical), then $q(\mathbf{W})$ will be mean-field Gaussian. The variational parameters of each $t_n(\mathbf{W})$ can then be obtained by passing each data pair $(\mathbf{x}_n, \mathbf{y}_n)$ through an inference network, achieving per-datapoint amortisation.