Exploiting Inferential Structure in Neural Processes (Supplementary Material)

Dharmesh Tailor[1] **Mohammad Emtiyaz Khan**[2] **Eric Nalisnick**[1]

[1]University of Amsterdam, Amsterdam, Netherlands
[2]RIKEN Center for AI Project, Tokyo, Japan

A DERIVATIONS

A.1 VARIATIONAL LOWER BOUND WITH STRUCTURED INFERENCE NETWORK

For the conjugate case in Sec. 3, we show the ELBO in Eq. 1 (for a single task and dropping task indices for clarity) simplifies after substitution of the structured inference network (Eq. 7). We denote the factor by $q(\mathbf{z} \,|\, \mathbf{f}_{\phi_{\text{NN}}}(\mathcal{D}_c^{(i)})) := \exp\left(\left\langle \mathbf{T}(\mathbf{z}), \mathbf{f}_{\phi_{\text{NN}}}(\mathbf{x}_{c,i}, \mathbf{y}_{c,i}) \right\rangle\right)$ (and analogous expression for each target point):

$$\log p(\mathcal{D}_t \,|\, \mathcal{D}_c) \tag{1}$$

$$= \log \int_{\mathbf{z}} p(\mathcal{D}_t \,|\, \mathbf{z})\; p(\mathbf{z} \,|\, \mathcal{D}_c) \tag{2}$$

$$= \log \int_{\mathbf{z}} q_\phi(\mathbf{z} \,|\, \mathcal{D}_c \cup \mathcal{D}_t) \frac{p(\mathcal{D}_t \,|\, \mathbf{z})\; p(\mathbf{z} \,|\, \mathcal{D}_c)}{q_\phi(\mathbf{z} \,|\, \mathcal{D}_c \cup \mathcal{D}_t)} \tag{3}$$

$$\geq \mathbb{E}_{q_\phi(\mathbf{z} \,|\, \mathcal{D}_c \cup \mathcal{D}_t)} \left[\log \frac{p(\mathcal{D}_t \,|\, \mathbf{z})\; p(\mathbf{z} \,|\, \mathcal{D}_c)}{q_\phi(\mathbf{z} \,|\, \mathcal{D}_c \cup \mathcal{D}_t)} \right] \tag{4}$$

$$\approx \mathbb{E}_q \left[\log \frac{p(\mathcal{D}_t \,|\, \mathbf{z})\; q_\phi(\mathbf{z} \,|\, \mathcal{D}_c)}{q_\phi(\mathbf{z} \,|\, \mathcal{D}_c \cup \mathcal{D}_t)} \right] \tag{5}$$

$$= \mathbb{E}_q \left[\log p(\mathcal{D}_t \,|\, \mathbf{z}) \right] + \mathbb{E}_q \left[\log \frac{\cancel{\prod_{i=1}^{N_c} q(\mathbf{z} \,|\, \mathbf{f}_{\phi_{\text{NN}}}(\mathcal{D}_c^{(i)}))}}{\cancel{\prod_{i=1}^{N_c} q(\mathbf{z} \,|\, \mathbf{f}_{\phi_{\text{NN}}}(\mathcal{D}_c^{(i)}))} \prod_{i=1}^{N_t} q(\mathbf{z} \,|\, \mathbf{f}_{\phi_{\text{NN}}}(\mathcal{D}_t^{(i)}))} \frac{\cancel{q(\mathbf{z}; \phi_{\text{PGM}})}\; Z_{c,t}(\phi)}{\cancel{q(\mathbf{z}; \phi_{\text{PGM}})}\; Z_c(\phi)} \right] \tag{6}$$

$$= \mathbb{E}_q \left[\log p(\mathcal{D}_t \,|\, \mathbf{z}) \right] - \sum_{i=1}^{N_t} \mathbb{E}_q \left[\log q(\mathbf{z} \,|\, \mathbf{f}_{\phi_{\text{NN}}}(\mathcal{D}_t^{(i)})) \right] + \log Z_{c,t}(\phi) - \log Z_c(\phi) \tag{7}$$

where the 2nd term resembles the entropy on the individual factors (shown in blue).

A.2 EQUIVALENCE TO BAYESIAN AGGREGATION MEAN UPDATE EQUATION

We show the posterior mean in Eq. 8 can be expressed in the incremental form stated in Volpp et al. [2020] (Eq. 4):

$$\widetilde{\boldsymbol{\mu}} = \widetilde{\boldsymbol{\Sigma}} \left(\sum_{i=1}^{N_c} \mathbf{V}_{c,i}^{-1} \mathbf{m}_{c,i} + \boldsymbol{\Sigma}_0^{-1} \boldsymbol{\mu}_0 \right) \tag{8}$$

$$= \widetilde{\boldsymbol{\Sigma}} \left(\sum_{i=1}^{N_c} \mathbf{V}_{c,i}^{-1} \mathbf{m}_{c,i} + \widetilde{\boldsymbol{\Sigma}}^{-1} \boldsymbol{\mu}_0 - \sum_{i=1}^{N_c} \mathbf{V}_{c,i}^{-1} \boldsymbol{\mu}_0 \right) \tag{9}$$

$$= \boldsymbol{\mu}_0 + \widetilde{\boldsymbol{\Sigma}} \sum_{i=1}^{N_c} \mathbf{V}_{c,i}^{-1} \left(\mathbf{m}_{c,i} - \boldsymbol{\mu}_0 \right) \tag{10}$$

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

A.3 MIXTURE OF GAUSSIAN PRIOR NORMALIZATION CONSTANT

$$C_k = (2\pi)^{-\frac{DN}{2}} \prod_{i=1}^{N_c} \det(\mathbf{V}_{c,i})^{-\frac{1}{2}} \left(\frac{\det(\boldsymbol{\Sigma}_k)}{\det(\widetilde{\boldsymbol{\Sigma}}_k)} \right)^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2} \left(\sum_{i=1}^{N_c} \mathbf{m}_{c,i}^\top \mathbf{V}_{c,i}^{-1} \mathbf{m}_{c,i} + \boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}_k^{-1} \boldsymbol{\mu}_k - \widetilde{\boldsymbol{\mu}}_k^\top \widetilde{\boldsymbol{\Sigma}}_k^{-1} \widetilde{\boldsymbol{\mu}}_k \right) \right\} \quad (11)$$

B EXPERIMENTAL DETAILS

The implementation for robust and mixture Bayesian aggregation is adapted from the implementation of BA[1] and the other baselines are taken from the codebase of Bootstrapped Neural Process[2] (with modifications elaborated in the appendix) [Lee et al., 2020].

For our proposed mixture Bayesian Aggregation, we use the `MixtureOfDiagNormals` implementation from the `Pyro` package. However due to numerical issues with using 32-bit floating-point precision with this implementation, we disable gradient flow through the categorical distribution.

B.1 1-D REGRESSION

We consider the following kernels:

RBF: baseline used in Lee et al. [2020]

$$k(x, x') = s^2 \exp\left(-\frac{(x - x')^2}{2\ell^2} \right)$$

with $s \sim \mathcal{U}[0.1, 1.0)$ and $\ell \sim \mathcal{U}[0.1, 0.6)$;

Matérn–$\frac{5}{2}$: baseline used in Gordon et al. [2020]

$$k(x, x') = \left(1 + \sqrt{5}d + \frac{5}{3}d^2 \right) \exp\left(-\sqrt{5}d \right)$$

with $d = 4|x - x'|$.

Following Lee et al. [2020], the inputs of the context and target sets are sampled according to $x \sim \mathcal{U}(-2, 2)$. The sizes of the context and target sets are sampled according to $N_c \sim \mathcal{U}(3, 47)$ and $N_t \sim \mathcal{U}(3, 50 - N_c)$. In the evaluation phase, 5000 tasks are drawn identically to the data generating process for training.

B.2 IMAGE COMPLETION

Image completion is formulated as a regression problem where pixel coordinates are transformed to $[-1, 1]$ and pixel intensities are rescaled to $[-0.5, 0.5]$ following Lee et al. [2020]. A single image constitutes a task. During training, images are restricted to the first 10 classes, with the size of the context and target sets are sampled according to $N_c \sim \mathcal{U}(3, 197)$ and $N_t \sim \mathcal{U}(3, 200 - N_c)$. For the in-distribution setting, we evaluate on a different set of images but restricted also to the first 10 classes. For Fig. 5a we sample the target sets according to $N_t \sim \mathcal{U}(3, 500 - N_c)$ (evaluation only). For evaluation on the out-of-distribution setting, images are taken from the unseen classes 10-46.

B.3 MODEL ARCHITECTURES

Decoder architecture Across all models, we keep the decoder architecture the same, with separate networks outputting the mean and standard deviation of the predictive distribution (following Volpp et al. [2020]). The networks have 128 hidden units and ReLU activation functions. For the 1-D regression experiment, we use a 3-layer MLP and for the image completion experiment, a 4-layer MLP. Following Le et al. [2018], the standard deviation of the predictive distribution is processed using a lower-bounded softplus with a lower bound of 0.1.

[1] `https://github.com/boschresearch/bayesian-context-aggregation`

[2] `https://github.com/juho-lee/bnp`

Encoder architecture For models with latent path, the latent dimensionality is set to 128. ReLU activation function for the hidden layers are used throughout. Unless otherwise stated, the hidden size is 128. Here we state the architectures of the baselines with mean-pooling aggregation in the 1-D regression experiment:

NP: This is adapted from Garnelo et al. [2018] where the deterministic path is removed (as done in Volpp et al. [2020]).

NP+SA: This incorporates (multi-head) self-attention into the encoder architecture of NP.

Following Le et al. [2018], we process the standard deviation of the latent variable using a lower-bounded sigmoid with a lower bound of 10^{-4}.

The architecture of the baselines with Bayesian Aggregation (BA) follow Volpp et al. [2020] with separate MLPs predicting each neural sufficient statistic (or latent observation and observation variance as elaborated in Volpp et al. [2020]). The 2nd neural sufficient statistic (i.e. observation noise) is processed using a lower-bounded sigmoid, identical to how the latent variance is processed in the baselines with mean-pooling. Following Volpp et al. [2020], a Gaussian prior with fixed parameters is used: $\boldsymbol{\mu}_0 = \mathbf{0}, \boldsymbol{\Sigma}_0 = \mathbf{I}$. We use MLPs with 64 hidden units and 4 layers.

For our proposed robust Bayesian Aggregation, we extend the aforementioned BA implementation. The gamma prior parameters are set as follows: $a_0 = b_0 = 10^{-6} \cdot D$ and $c_0 = 10^{-2} \cdot D$. This is similar to Tipping and Lawrence [2005] but appropriately scaled by the latent dimensionality. For the image completion experiment, the depth of all MLPs is increased by 1.

The models evaluated in Sec. 5.1 are trained using 10 latent samples and those evaluated in Sec. 5.2 are trained using 5 latent samples. We use more latent samples for the mixture experiments as suggested in Wang and Van Hoof [2022]. Following Lee et al. [2020], all models are optimized using ADAM with initial learning rate $5 \cdot 10^{-4}$ and cosine annealing scheme for the learning rate schedule. For the 1-D regression experiment, models are trained for 100,000 steps where each step consists of 16 tasks. For the image completion experiment, models are trained for 200 epochs with batches of 100 images.

To evaluate the models, we compute the posterior predictive log-likelihood and RMSE using a Monte-Carlo approximation. Following Kim et al. [2019] we also evaluate the criterion by using the context points as targets as well. This gives an indication of how well the model is fitting the context points (reconstruction error).

C EXTENDED TABLE 1 WITH ROBUST BA COMPARISON

	Predictive Log-Likelihood ↑				**RMSE ↓**			
	Seen classes (0-9)		Unseen classes (10-46)		Seen classes (0-9)		Unseen classes (10-46)	
	context	target	context	target	context	target	context	target
NP	$0.701_{\pm 0.065}$	$0.589_{\pm 0.061}$	$0.567_{\pm 0.059}$	$0.405_{\pm 0.039}$	$0.201_{\pm 0.018}$	$0.218_{\pm 0.014}$	$0.244_{\pm 0.014}$	$0.265_{\pm 0.009}$
NP+SA	$0.977_{\pm 0.006}$	$\mathbf{0.840}_{\pm 0.005}$	$0.823_{\pm 0.007}$	$0.609_{\pm 0.008}$	$0.127_{\pm 0.002}$	$\mathbf{0.165}_{\pm 0.001}$	$0.177_{\pm 0.002}$	$0.224_{\pm 0.002}$
NP-BA	$0.866_{\pm 0.097}$	$0.708_{\pm 0.076}$	$0.749_{\pm 0.118}$	$0.537_{\pm 0.093}$	$0.154_{\pm 0.027}$	$0.193_{\pm 0.014}$	$0.193_{\pm 0.033}$	$0.238_{\pm 0.018}$
NP-mBA (K=2)	$0.952_{\pm 0.121}$	$0.778_{\pm 0.083}$	$0.855_{\pm 0.143}$	$0.623_{\pm 0.097}$	$0.128_{\pm 0.033}$	$0.181_{\pm 0.017}$	$0.162_{\pm 0.038}$	$0.221_{\pm 0.020}$
NP-mBA (K=3)	$0.953_{\pm 0.108}$	$0.777_{\pm 0.079}$	$0.857_{\pm 0.132}$	$0.623_{\pm 0.103}$	$0.128_{\pm 0.031}$	$0.180_{\pm 0.015}$	$0.162_{\pm 0.038}$	$0.221_{\pm 0.020}$
NP-mBA (K=5)	$0.975_{\pm 0.083}$	$0.792_{\pm 0.064}$	$0.883_{\pm 0.105}$	$0.642_{\pm 0.084}$	$0.122_{\pm 0.025}$	$0.177_{\pm 0.011}$	$0.155_{\pm 0.031}$	$0.217_{\pm 0.016}$
NP-rBA	$\mathbf{0.981}_{\pm 0.136}$	$0.801_{\pm 0.091}$	$\mathbf{0.897}_{\pm 0.161}$	$\mathbf{0.668}_{\pm 0.101}$	$\mathbf{0.119}_{\pm 0.038}$	$0.178_{\pm 0.019}$	$\mathbf{0.148}_{\pm 0.042}$	$\mathbf{0.214}_{\pm 0.021}$

References

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J. Rezende, S. M. Eslami, and Yee Whye Teh. Neural Processes. *ICML Workshop on Theoretical Foundations and Applications of Deep Generative Models*, 2018.

Jonathan Gordon, Wessel P. Bruinsma, Andrew Y. K. Foong, James Requeima, Yann Dubois, and Richard E. Turner. Convolutional Conditional Neural Processes. In *Proceedings of the International Conference on Learning Representations*, 2020.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive Neural Processes. In *Proceedings of the International Conference on Learning Representations*, 2019.

Tuan Anh Le, Hyunjik Kim, Marta Garnelo, Dan Rosenbaum, Jonathan Schwarz, and Yee Whye Teh. Empirical Evaluation of Neural Process Objectives. In *NeurIPS Workshop on Bayesian Deep Learning*, 2018.

Juho Lee, Yoonho Lee, Jungtaek Kim, Eunho Yang, Sung Ju Hwang, and Yee Whye Teh. Bootstrapping Neural Processes. In *Advances in Neural Information Processing Systems*, 2020.

Michael E. Tipping and Neil D. Lawrence. Variational inference for Student-t models: Robust Bayesian interpolation and generalised component analysis. *Neurocomputing*, 2005.

Michael Volpp, Fabian Flürenbrock, Lukas Grossberger, Christian Daniel, and Gerhard Neumann. Bayesian Context Aggregation for Neural Processes. In *Proceedings of the International Conference on Learning Representations*, 2020.

Qi Wang and Herke Van Hoof. Learning Expressive Meta-Representations with Mixture of Expert Neural Processes. In *Advances in Neural Information Processing Systems*, 2022.